Activity Monitoring for large honeynets and network telescopes

Jérôme François, Radu State, Olivier Festor

Activity Monitoring for large honeynets and network telescopes

Jérôme Francois, Radu State and Olivier Festor
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, while the second deployment target is the monitoring of a large network telescope. 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.**

***Index Terms*—honeypot, backscatter, telescope, monitoring, intersection graphs, centrality, locality statistics**

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 honeynet 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. In our work we had to process similar attack traffic from a different security monitoring platform (a network telescope) and compare it to the results obtained from the honeynet. This paper extends our previous works [1] and [2].

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 describes two realistic distributed environments (a honeynet and a network telescope) and section 4 shows how this method can be used for them. 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 5 presents related works and finally section 6 concludes the paper.

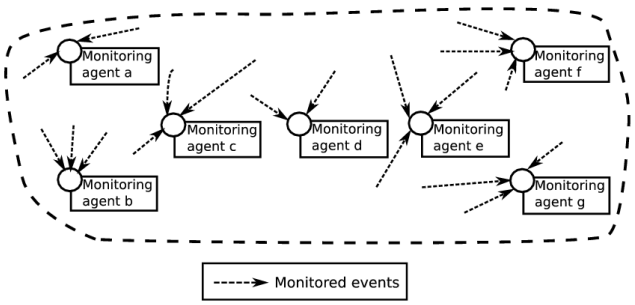


Fig. 1. Distributed monitoring model

II. Intersection graphs

The method based on intersection graphs has been introduced in [3] for profiling communications patterns between the users of a high profiled enterprise. Actually, the data used were the exchanged emails and the goal was to detect if someone was aware of the Enron scandal before it was revealed. Thanks to this method, the authors observe that there were significative changes of the graph topology and highlight the responsable nodes which are in reality people. Therefore, using this technique seems to be a good way to detect behavior changes of the attacks in the Internet and IP addresses which are concerned by these changes.

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, the adjacency matrix is symmetric :

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

As we want to connect nodes which share or don't share some characteristics, it is totally useless for a node connected to be connected to itself and we will consider this statement as an assumption in all this article.

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

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

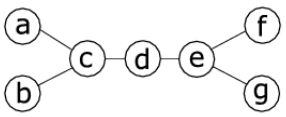


Fig. 2. An undirected graph

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. This is the main idea used in [3].

In figure 2, considering an exploring distance k = 2, nodes c and e have a centrality of 4. For the node d, the associated value is 6. Based 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 [4]. Assuming an adjacency matrix A, x an eigen vector and λ the corresponding eigen value, we have :

$$A \times x = \lambda \times x$$

The more central node is the highest value in the eigenvector of the highest eigenvalue. Considering the figure 2 and the previously introduced adjacency matrix, this vector is (-0.5, 0, -0.316, 0.500, 0.000, -0.447, -0.447). The maximal value is the fourth which corresponds to the node d once again.

Thus, different methods can be used and we propose to use the first one in this paper because it is done easily by walking in the graph and because we can compute the centrality incrementally for different distances i.e. by increasing the depth of the walking contrary to the second methods where the eigenvectors and eigenvalues are to be recomputed for each submatrix.

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 analyze 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. So, detecting changes in the graph can be highlighted by looking for the maximal centrality as proposed in [3]. This method has the advantage that one value is an indicator of the graph topology contrary to have one value per node. If more details are needed, 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.

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

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

$$of \ neighbors \ of \ v \ at \ a \ maximal \ distance \ k$$

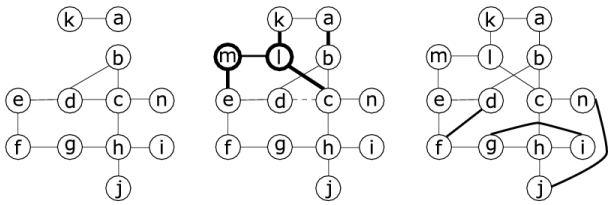
$$M_k = \max_{v \in nodes} \psi_k(v) \quad (1)$$

Actually, the number of neighbors at a maximal distance k is computed for each node. Then M_k is the maximum value that were be calculated.

Consider the example of the evolution of a graph which is described below and presented briefly in the figure 3:

- 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)

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. Varying k allows to select information and especially to limit the subgraph of extended neighbors in order to avoid to have a constant maximal locality statistic which corresponds to a subgraph covering all the graph.

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 (t = 6). The maxima 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 maximal locality statistic (maximum centrality). In the previous example, node c is always central.

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 at time t:

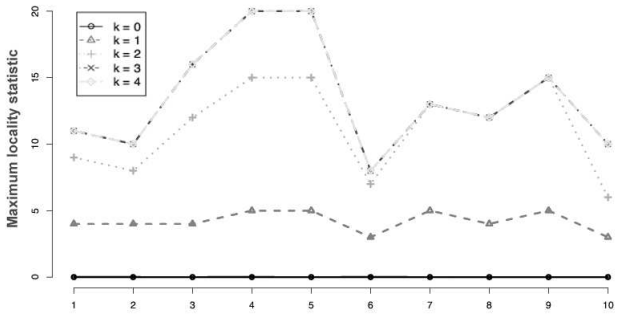


Fig. 4. Locality statistics according to time

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

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

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

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

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. Therefore we compute for each node the size of the subgraph which contains the neighbors at a maximal distance k. Then we calculate the common average value during the sliding window: $\hat{\mu}_{k,t,\tau}(v)$. Then, the variance is computed: $\hat{\sigma}_{k,t,\tau}(v)$. Therefore, each node have an associated standardized value for the centrality which is $\tilde{\psi}_{k,t}(v)$. The standardized locality statistics is the maximum value between all $\tilde{\psi}_{k,t}(v)$. 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.

Besides, there is a peak at the beginning of the curves due to the initialization of the sliding window. During this stage, when a new node appears it becomes often the more central node or at least one of the highest central node. It is not a real problem as that the apparition of new node is an important fact. Finally, by comparing the peaks of the figure 2 and 5, there are not at the same positions because the figure 5 illustrates the dynamicity of the graph. Therefore, even if the maximum locality statistic increases during 3 time units which means that the peak is the last value, the standardized locality statistics can be a previous one if the increasing is more important at the beginning than at the end.

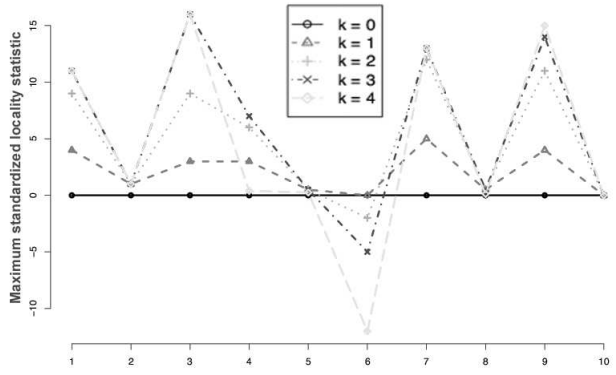


Fig. 5. Locality statistics according to time ($\tau = 5$)

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. The major idea is to consider an arc between two nodes, if and only if the associated agents have observed a different 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 little overlap in the observed parameters, they should be linked and it will be highlighted by the locality statictics.

III. Data description

A. Network telescope

The principle of network telescope is simple. A monitoring device saves all incoming traffic to a specific range of IP addresses. In fact, these addresses are unused and cover a range which is generally a subnetwork of consecutive addresses. The main characteristic of a telescope is its size which is generally huge. It is possible to create more interactive network telescopes which emulate diversified services like shown in [5], but in our case the telescope is totally passive and just records the incoming packets. Because the monitored addresses are normal and are secret, an attacker is unable to know these ones and attacks can be targeted against these.

We used in our work data from the telescope developed in the CAIDA project [6]. The monitored addresses form an A class network and the number of addresses is 2^{24}. This huge telescope gathers data from a fraction of $\frac{1}{256}$ of the Internet. Only backscatter packets are captured by this telescope. Backscatter packets are generated indirectly by a denial of service attacks and for a comprehensive overview, the reader is referred to [7]. Basically, a backscatter packet contains an the ack field set as it is a response. The basic scenario is as follows: an attacker does a SYN flooding of a victim in order to force the victim to reply to each packet. The attacker can spoof the source IP addresses in order to hide her identity and avoid additional bandwidth consumption on her side. The victim of the denial of service attack replies to the spoofed addresses and these replies are called backscatter packets. The figure 6 shows a simple scenario where an attacker spoofs three IP addresses but only one is assigned to a real and legitimate network interface. The others are a part of the addresses of a telescope which collects these backscatter packets. Therefore the response can be captured by the telescope. The assumption of that the telescope monitors only backscatter packets is limited because some of this packets can be generated by an ACK port scanning. Moreover, the telescope stores also the ICMP response which can be due to a ICMP echo request for instance.

During our analysis, only the period from 26 to 36 August 2004 is studied on a hour by hour basis. About 460 millions of packets have been gathered during this period corresponding to 24.1 GB of data. For more information about the data, please refer to the table I.

		Network Telescope
#Observed IP source addresses		116 777 216
Number of incoming packets	2004/08/26	52 784 835
	2004/08/27	88 411 307
	2004/08/28	142 096 855
	2004/08/29	77 094 947
	2004/08/30	51 850 438
	2004/08/31	45 742 568
Number of unique source IP addresses	2004/08/26	171 257
	2004/08/27	244 643
	2004/08/28	241 883
	2004/08/29	242 491
	2004/08/30	231060
	2004/08/31	246 982
Size of data	2004/08/26	3,8 MB
	2004/08/27	6,3 GB
	2004/08/28	1,5 GB
	2004/08/29	5,5 GB
	2004/08/30	3,7 GB
	2004/08/31	3,3 GB

TABLE I
Global information about the telescope data

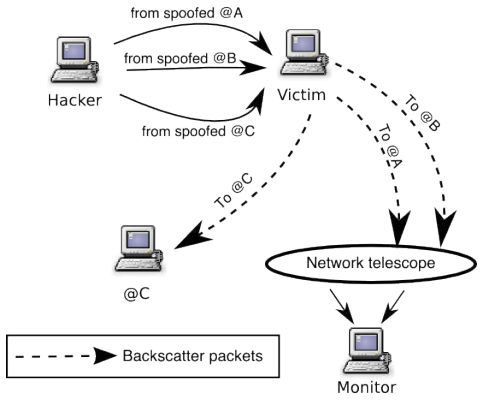


Fig. 6. Backscatter principle

B. Honeynet

A honeypot is described in [8] as an environment where vulnerabilities are deliberately introduced. Malicious in-

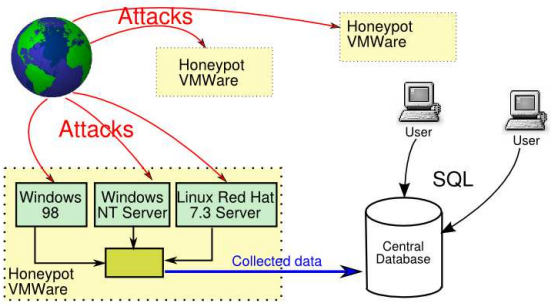


Fig. 7. Leurre.com honeynet architecture

truders 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. More recently, honeypots and honeynets have been used to observe the behavior and spreading of automated malware like worms and autorooters. The basic idea behind a honeypot is that a vulnerable system is simulated to the outside and more or less simulated services are exposed in order to achieve an interaction with the attacker (or automated malware). The degree of interactions can vary from simple and low interaction honeypots (like the ones described in [9]) and up to complete worm capturing architectures (the mwcollect project is a very good example of such an architecture), or even human driven high interaction honeypots. The first description of such a honeypot, although not named as such, can be found in the [10], where a human network administrator manually emulates a rogue vulnerable system in order to study an intruder.

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 honeypots 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). The number of monitored IP addresses is 3×43 which is very lower than for the telescope. However, the IP addresses are well distributed in IP domains contrary to the telescope whose the data can be biased by attacks targeted specific IP domains. Data is collected locally and centralized in a database. There are low interactions honeypot and the collected data are stored in a central Database accessed by SQL request as you can see on the honeynet description in the figure 7.

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 II 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 II
GLOBAL INFORMATION ABOUT THE HONEYNET DATA. THE MONTHS ARE REPRESENTED IN NUMBER (05, 06, 07...)

IV. INTERSECTION GRAPHS APPLICATION

In this section, the intersection graphs method is applied to the previously described monitoring platform : honeynet and network telescope. Several aspects will be studied: source IP addresses, source ports, attack tools used, misconfigurations and targeted services.

A. Source IP addresses

1) Honeynet: The goal of our first analysis is to analyze 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 α 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 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.

Determining the threshold is not easy. In fact, it depends on the objective. For example, some characteristics (like source IP addresses) are more variable and so normally the thresholds will be very low because we should not see the same value many times. Other characteristics have often the same value as the targeted port of an attack (like web servers). Therefore, the conclusions have to consider these thresholds in order to say if the different nodes see really

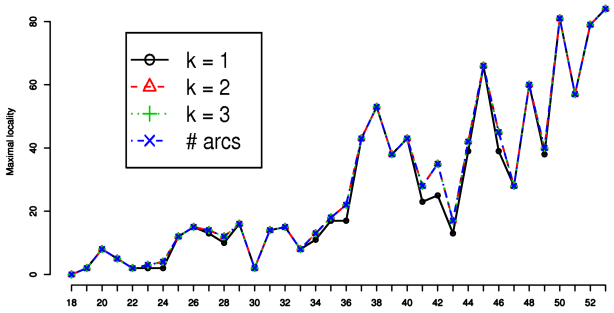


Fig. 8. Maximal locality, (shared addresses *alpha* $\leq 0.25\%$), x-axis are the week numbers

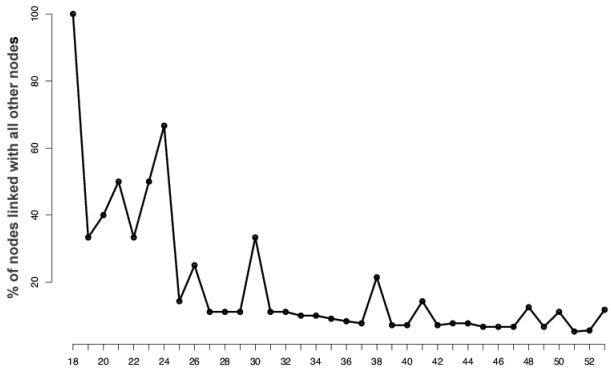


Fig. 9. Number of central nodes with the maximum locality for the honeynet

different things or not. Moreover, tuning them to obtain result similarities between the Honeynet and the telescope is a good way to evaluate how these monitoring platform kinds are different by comparing the thresholds.

After some tests, for small thresholds α, the plots tend to overlap and a good setting of this value is 0.25%, where only few points are not overlapped. The figure 8 shows the maximum locality and the total number of arcs in the graph and all the curves are very similar and close to the number of arcs. It means that for $k = 1$, a node is linked to each other one except for few cases which means that at least one honeypot is very different in terms of observed IP addresses. Therefore, the figure 9 shows the number of nodes with the maximal centrality and so the ones which are linked with each others. There are some peaks but the curve decreases and tends to the value of 10%. Obviously, the corresponding honeypot platforms can be known and this information is useful for improving the analysis of honeypot data by limiting the amount of its.

The figure 10 represents the standardized locality with $\tau = 5$ weeks. Using the method of the intersection graphs, 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.

2) Network telescope: The goal of this study is similar to the previous honeynet analysis. We wanted to detect if a part of a telescope detects source IP addresses which are not detected by other parts. The range of IP addresses monitored is sliced into several /16 subnetworks. Because of the size of the telescope is a /8, we consider $2^8 = 256$ subnetworks. This division is logically equivalent to a distributed monitoring model described in figure 1. When this model is instantiated, we obtain the architecture illustrated in figure 11. In fact, each subnetwork of the telescope is considered as an entity for which there is one monitoring agent.

The nodes are the subnetworks and two nodes are linked if the intersection of their source IP addresses is less than

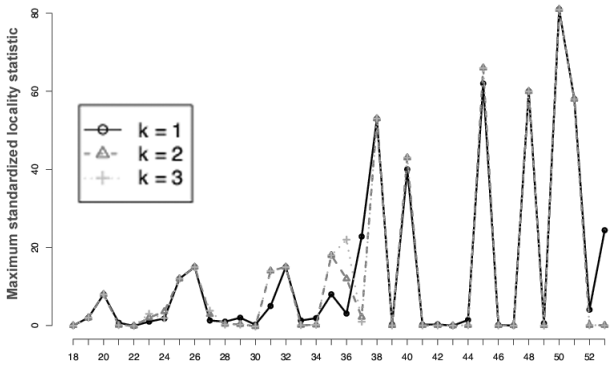


Fig. 10. Honeynet source IP addresses analysis - Standardized locality with $\tau = 5$ (shared addresses $\leq 0.25\%$)

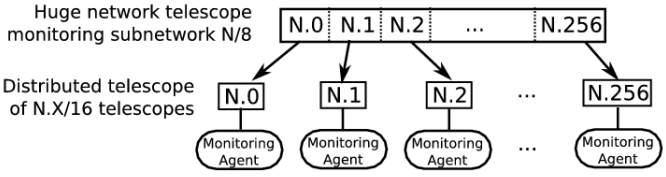


Fig. 11. A distributed telescope

a threshold α of their union. If the subnetworks were really different in term of observed source IP addresses, a lot of links would appear and the locality statistic would increase.

We have tested threshold values of 5% and the maximum locality statistic is always 0 except for the first hour which is probably due to a lack of data at the beginning of the capture (because the August 26 is the first day of August for which we have data). A threshold value of 5% is low but we also intended to compare honeynet (threshold was 0.25%) and telescopes and we concluded that there is a high redundancy of information in the telescope case.

B. Source ports

A second goal was to detect plateforms 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.

1) Honeynet: 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 β 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 12 (for a threshold of 10%) and respectively in figure 13 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 12 and 13.

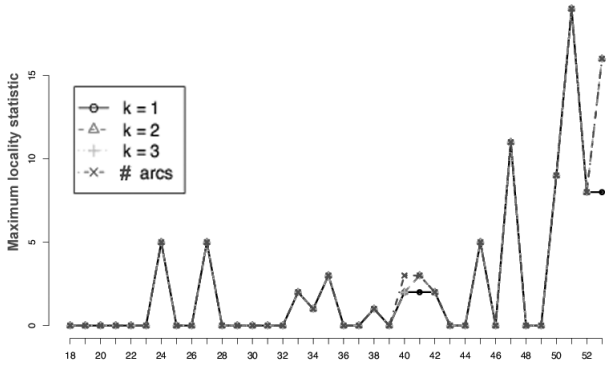


Fig. 12. Honeynet source ports analysis - locality statistic (shared ports $\leq$ 10%)

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 : $\sum_{i=43-1}^{1} i = 946$.

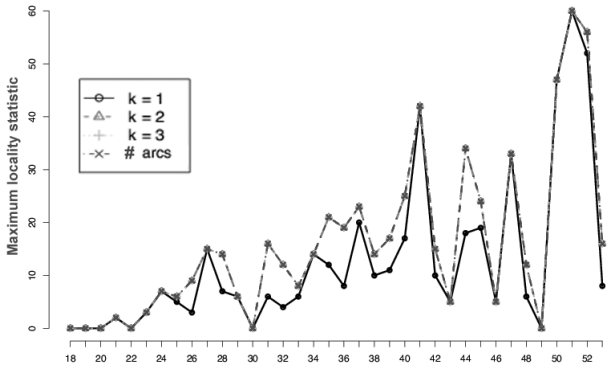


Fig. 13. Honeynet source ports analysis - locality statistic (shared ports $\leq$ 25%)

2) Network telescope: The packets that have been captured by the telescope are only backscatter packets and so the source ports of these packets are in fact attacked ports. It's interesting to study them in the same manner that we have done it for the honeynets. The difference here is that the nodes are the subnetworks of size /16 of the range of monitored IP addresses. Our goal was to detect if sometimes, only particular ports were attacked.

Using a threshold of 5% we obtained the plots shown in figure 14. The number of arcs and the locality statistic is close to 0. The source ports shown by the different subnetworks are the same. The conclusion is the same as for the honeynet case : attackers attack frequently the same ports and the telescope can detect this phenomena.

A peak appears clearly on the figure 14 and in fact there are 3 subnetworks detecting unusual source ports. This is opposed to the honeynet case for which a peak is not always significant due to a low amount of data. Because a telescope monitors a fraction of $\frac{1}{256}$ of the Internet, a high peak like its shows a real specific phenomena at this time and this peak is a proof of attacks on original ports.

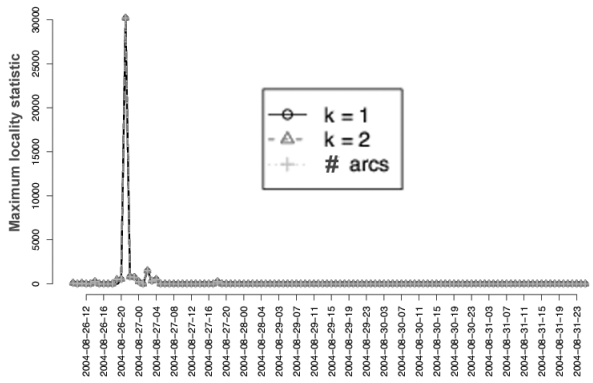


Fig. 14. Telescope source ports analysis - locality statistic (shared ports $\leq$ 5%)

C. 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 experiment, only the honeynet is considered.

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 15. 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.

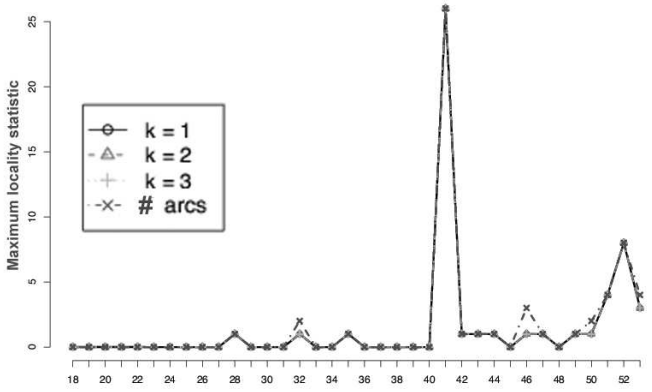
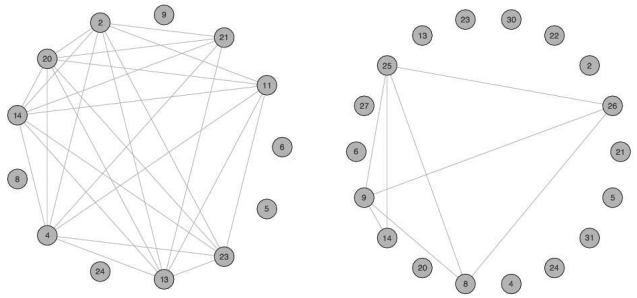


Fig. 15. Honeynet acknowledgment numbers analysis - locality statistic (shared acknowledgment numbers $\geq 90\%$)

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 16 presents the graphs of weeks 41 and 52 corresponding to the peaks. In the figure 16(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 16(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. 16. Intersection graphs for acknowledgment numbers shown by the honeynet

D. Detecting misconfigurations

During our previous analysis, many source IP addresses were invalid like many local addresses. It can be due to some attackers but smart ones prefer to use valid addresses in order to be undetected. Therefore, most of them can be considered as misconfiguration problems on user computers or at the Internet service provider

1) Sources: There are many types of addresses that are dedicated to specific use and that shouldn't be use on Internet. The table III gives a summary of such addresses as well as their target deployment usage. However, we were amazed by the large quantity of observed IP addresses that should in theory never appear on the Internet. Several factors jointly produce them: misconfigured enterprise routers/firewalls, missing ISP level ingress/egress filtering and maybe defective devices.

Range	Description
10.0.0.0 → 10.255.255.255	Class A private addresses
172.16.0.0 → 172.31.255.255	Class B private addresses
192.168.0.0 → 192.168.255.255	Class C private addresses
224.0.0.0 → 239.255.255.255	Class D multicast addresses
240.0.0.0 → 255.255.255.255	Class E addresses reserved for experimental use
127.0.0.0 → 127.255.255.255	Loopback addresses
0.0.0.0 → 0.255.255.255	addresses of network 0 (class A)
169.254.0.0 → 169.254.255.255	addresses of DHCP client which can't obtain an address from the server
192.0.2.0 → 192.0.2.255	Loopback addresses

TABLE III
Abnormal source addresses on Internet

The left barchart of figure 17 shows the proportion (per 100 000) of the different type of abnormal addresses

considering unique IPs in comparison with the total number of unique IPs for the observed days in the case of the telescope. This graph allows to observe both the main types of abnormal addresses and their corresponding global proportion.

There is a category which is about constant (colored in black). It is the proportion of network 0 addresses (class A). Normally 0.0.0.0 can be used only as source broadcast address on local segments but not on the global Internet. However the global proportion increases significantly from June to August with peaks in June, at the end of August and the beginning of September. Very strangely is also the apparition of multicast addresses as source addresses. Multicast addresses can be only used as a destination address and will never appear as source addresses. Moreover this increase in abnormal addresses is also due to private IP addresses used in outgoing reply packets. These packets are received by the telescope (and for these packets the source appears to be a private IP address).

An attacker is able to forge such packets thanks different software like [11] but as previously introduced, discovering the attacker is easier in this case. Moreover, these packets are backscatter packets which means that main of them are responses from victims which don't forge the packets, such that we can safely assume that the majority is not malicious. The most probably source of these packets are misconfigured routers/firewalls/NATs. This increase can be also caused by an ISP deploying some new policy based routing rules, which were misconfigured. The concerned computers are connected to Internet but don't receive the responses of their own requests. Another justification of the apparition of private addresses (the class C for instance, which are generally used by home users) are a definite evidence of misconfigured network devices. However, the main issue is that the ISP does not block these addresses. The observed results can be generalized beyond the simple observed traffic as follows:

2^{24} : *IP addresses monitored by the telescope*
2^{32} : *all possible IP addresses*
Assuming that about 75% *of addresses are used on Internet*
y : *number of IP addresses concerned by an analysis*
x : *estimation of the number of IP addresses corresponding to the same analysis for the whole Internet*

$$x = \frac{2^{32} * 0.75 * y}{2^{24}}$$

This type of generalization can be applied to all the observed data in this paper

We performed a similar analysis with the data from the honeynet (at the right on the same figure 17) but in this case, a bar represents a month period. The results show a different pattern than the backscatter analysis. First the graph shows two peaks but not at the same time. The first in May and the second in July. The usage of private class IP addresses is also significant and the explanation might be the same i.e. the misconfiguration of local network and providers that don't do ingress filtering. However the main type of abnormal IPs is the range of addresses automatically assigned by a computer when the DHCP server don't respond to its request for obtaining an address. The cause is probably due to local networks with a non valid configuration of the DHCP service.
For comparing the two traces, we had to compare data from backscatter traffic observed from the telescope with data (directly incoming and backscatter) from the honeynets. We could not rely entirely on only the backscatter traffic from the honeynets due to the lack of massive datasets.

2) Open Windows specific ports: The Windows operating systems uses a series of defaults ports for its proprietary network protocols: ports 137, 138 and 139. The Netbios service is designed for sharing resources on a local network and this port is not only useless on the Internet but represents one major entry point for malware and malicious intruders. Moreover the port 445 is also a dangerous port because it is used for file sharing and many worms (Sasser and mutants exploit). To prevent these attacks, these ports should be filtered by a firewall.

Considering the telescope, the figure 18 shows the number of unique IPs with an open port per 100 000 unique IPs. Receiving a backscatter response of a given port means that the port was open during the connexion of the attacker performing the denial of service attack. The ports 137,138 and 445 seem to be protected even if there is a little peak for the port 445 in November. However it's clear that the port 139 is less filtered as we can see on the several peaks of the graphs. It seems that in 2004, professional networks and home computers were generally protected by firewalls contrary to some years before, but this is seen through traces of Denial of service attacks. Since, most victims are typically either enterprises or blackhats waging Internet wars, these low numbers are justified.

The honeypot data contains only one IP address having the port 139 open, such that the use of honeypot is not a good way to detect this kind of misconfiguration. Only a telescope with a large range of IP can efficiently detect it. However you can notice that the only visible port is also the one which is the most frequently observed as opened by the telescope.

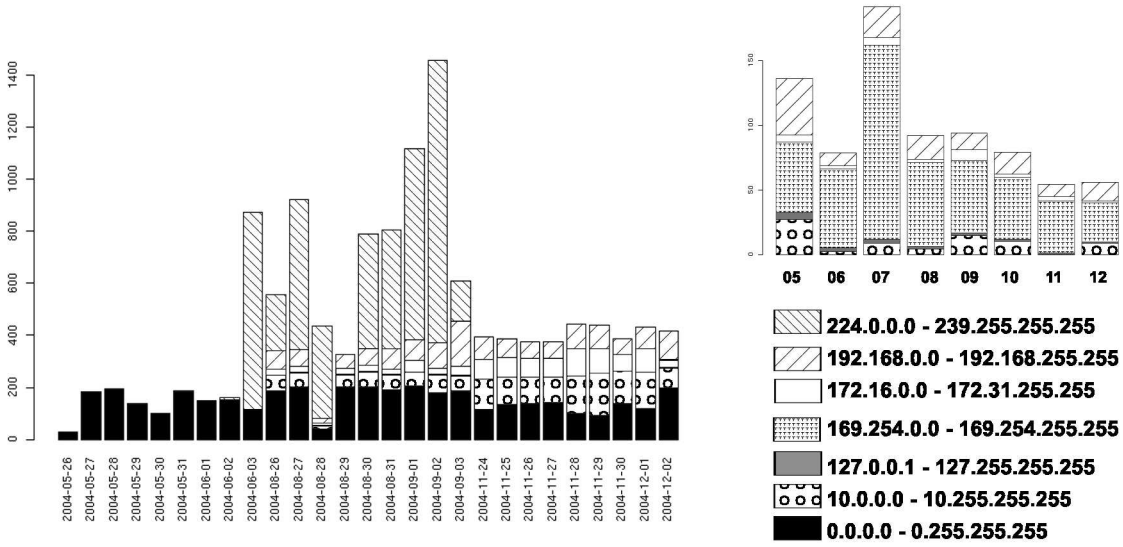


Fig. 17. Number of unique IP addresses of the different categories of abnormal IPs per 100 000 unique IP addresses. (Left : backscatter data, right : honeypot data by month)

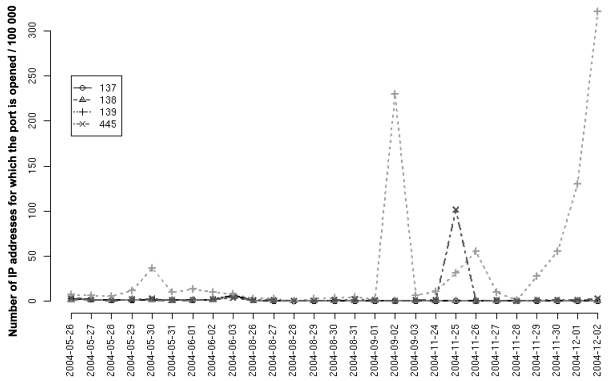


Fig. 18. Number of unique IPs with an open port per 100 000 unique IP addresses and according to each specific windows port. The chart represents the backscatter data of the telescope.

3) Analysis of ICMP 'Destination unreachable' message: When a host connects to another host which is not available, an ICMP message is sent to the source with the type 3 equal to 'Destination unreachable message'. An additional code [12] is also used to provide additional information. We analyzed the following 8 codes in our work:

- 0 : net unreachable
- 1: host unreachable
- 2 : protocol unreachable
- 3 : port unreachable
- 4 : fragmentation needed and don't fragment was set
- 9 : communication with destination network is administratively prohibited
- 10 : communication with destination host is administratively prohibited
- 13 : communication administratively prohibited

Polite firewalls will typically answer with codes 9, 10 or 13 to show that a device or service is filtered. Although such information can be very helpful when troubleshooting a network like detecting firewall misconfigurations, it can leak information about existing devices/open ports to an attacker and could determine him to try more advanced reconnaissance techniques. Less polite firewalls, configured by more security conscious network managers might directly reply with TCP packet whose the RST bit is set.

The figure 19 shows the evolution of the ICMP type 3 message codes. The left graph is about the telescope and highlights clearly a main change between October and November. First of all, the code 13 decreases much which can be due to a significant change in the behavior of network administrator which prefers to limit the revealed information. Moreover, the code 3 becomes the most popular code. This code means that the port is unreachable and so that the host exists. Therefore this change shows that the attacks are much well targeted from November and most of them are port scanning. The bars about honeypots is the right one on the figure 19. Once again, there is a change but it is smoother than for the telescope with the same observation as before, i.e. a decrease of code 13 and an increase of code 3. Finally, the main difference is that the honeynet detects the change earlier than the telescope.

E. Most attacked services

A natural question is related to which services are the most attacked services. We did this analysis on backscatter data for the different monitoring methods. Therefore, the packets reflect denial of service attacks. There are four main services which are attacked:

- The most attacked port and consistently ranked number 1 over all this period is port 80: it seems that

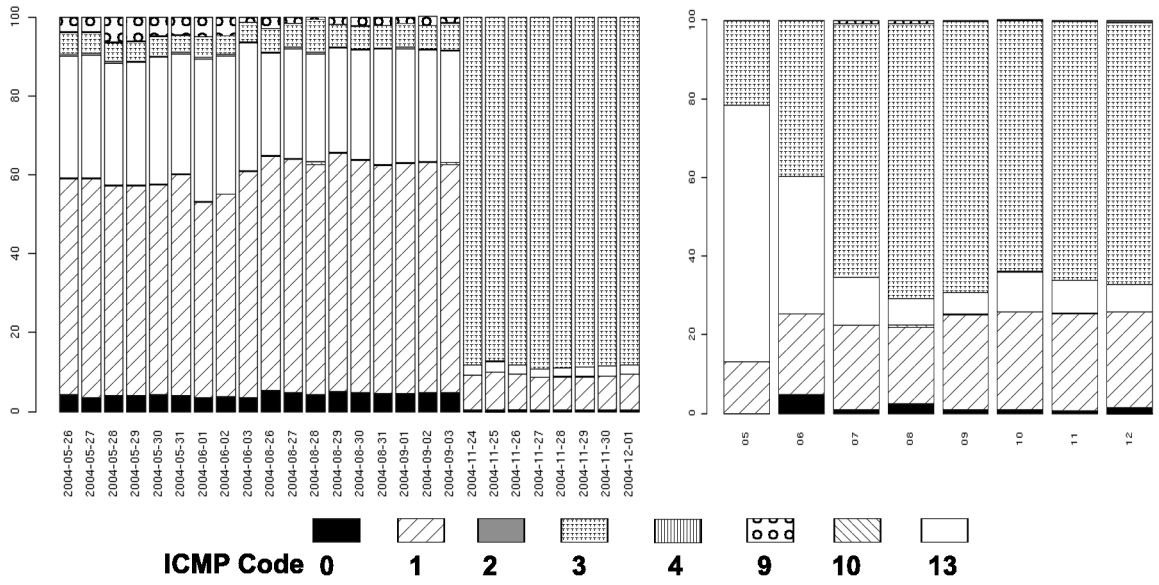


Fig. 19. Proportion of the different ICMP codes for the icmp type 3 (Destination unreachable). Backscatter telescope data are represented at the left and honeypot data is at right

web servers are the major target of denial of service attacks,

- port 6667 shows up frequently in the attacks. This port is typically used for IRC talks (or IRC anonymizing proxies like psyBNC). We suppose that these attacks are targeted at specific servers and can be associated to Internet war games waged to take the control of a a IRC channel,
- Name Servers (port 53) are also attacked (although to a lesser extend than IRC),
- Attacks against BGP routers (port 179) are also highly interesting and can be observed, since these attacks aim at either de-connecting a network domain, or can serve as preliminaries for a routing prefix hijack.

The table IV compare the most attacked services between the telescope (3 days) and the honeypot for May. Then we can see that the overlap of the ports is small : only the port 80. However if we consider table V in September, the overlap is totally different because 7 ports appear in the Honeypot and in the Telescope data. To conclude, even if sometimes, the two methods allow to get the same results, it appears that the results can be different and therefore the methods can be considered as complementary.

Moreover, in these tables (IV and V) an interesting fact is to have the port 7000 which is known as a backdoor. In the table VI the ports which are in the most attacked services with known vulnerabilities are listed. The vulnerabilities are common backdoors or ports used for the spreading of a worm. So, the attackers try also to do targeted denial of service attacks to open ports which are not reserved for a normal service.

Thus, we can conclude that ports are opened even if no service are traditionally associated and for which a vulnerability is known.

Port	Vulnerability
1011	Augudor
1025	Spybot
1433	Spybot
6000	Lovgate
7000	SubSeven
7001	Freak88
7300	NetMonitor
8000	Gaobot

TABLE VI
Some services which are in the most attacked services and which present known vulnerabilities

V. Related works

The honeypots and honeynets are presented in [8] 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. [13] is also an introduction to the different kinds of honeypot and highlights the different advantages of them and less frequently addressed question like legality or privacy problems.

In [14], the same authors propose a more elaborated method to study the data of the honeynet. In fact, the authors cluster the different captured network packets using the Levenshtein distance in order to group packets which are due to the same attack.

In [9], 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.

Honeypot May		Telescope 2004-05-26		Telescope 2004-05-27		Telescope 2004-05-28	
80	35 (63.64)	80	734 (7.61)	80	973 (10.03)	80	980 (16.27)
6667	5 (9.09)	21	15 (0.16)	21	17 (0.18)	139	14 (0.23)
3389	3 (5.45)	6667	15 (0.16)	4662	15 (0.15)	21	13 (0.22)
7000	3 (5.45)	139	13 (0.13)	139	13 (0.13)	22	11 (0.18)
1107	1 (1.82)	1002	12 (0.12)	25	11 (0.11)	113	10 (0.17)
1205	1 (1.82)	22	10 (0.10)	8080	11 (0.11)	25	10 (0.17)
1214	1 (1.82)	8080	10 (0.10)	110	10 (0.10)	8080	9 (0.15)
1235	1 (1.82)	110	9 (0.09)	113	10 (0.10)	443	8 (0.13)
1254	1 (1.82)	113	8 (0.08)	135	10 (0.10)	110	6 (0.10)
1271	1 (1.82)	111	6 (0.06)	22	8 (0.08)	178	6 (0.10)

TABLE IV
THE MOST ATTACKED SERVICES DURING MAY WHICH HAVE SENT SYN/ACK. THE FIRST NUMBER IS THE PORT AND THE SECOND THE NUMBER OF UNIQUE IP ADDRESSES WHICH ARE CONCERNED. THE NUMBER BETWEEN PARENTHESIS IS THE PERCENTAGE ACCORDING TO ALL UNIQUE COUPLE IP ADDRESS - OPEN PORT

Honeypot September		Telescope 2004-09-01		Telescope 2004-09-02		Telescope 2004-09-03	
80	116 (50.88)	80	956 (14.89)	80	1100 (19.66)	80	508 (17.69)
7000	49 (21.49)	7000	37 (0.58)	139	413 (7.38)	7000	24 (0.84)
7100	11 (4.82)	7200	13 (0.20)	7000	30 (0.54)	7100	21 (0.73)
22	9 (3.95)	7100	12 (0.19)	7100	22 (0.39)	7200	18 (0.63)
7200	7 (3.07)	21	10 (0.16)	7200	18 (0.32)	3389	12 (0.42)
7090	6 (2.63)	25	9 (0.14)	21	14 (0.25)	21	11 (0.38)
3389	4 (1.75)	22	8 (0.12)	3389	11 (0.20)	8080	8 (0.28)
21	3 (1.32)	443	8 (0.12)	22	10 (0.18)	139	5 (0.17)
113	2 (0.88)	8080	8 (0.12)	8080	8 (0.14)	6000	5 (0.17)
6667	2 (0.88)	3389	7 (0.11)	25	7 (0.13)	1524	2 (0.07)

TABLE V
THE MOST ATTACKED SERVICES DURING SEPTEMBER WHICH HAVE SENT SYN/ACK. THE FIRST NUMBER IS THE PORT AND THE SECOND THE NUMBER OF UNIQUE IP ADDRESSES WHICH ARE CONCERNED. THE NUMBER BETWEEN PARENTHESIS IS THE PERCENTAGE ACCORDING TO ALL UNIQUE COUPLE IP ADDRESS - OPEN PORT

Network telescopes have been the focus of several research works. In [15], the authors assume a simplified model and propose a simple formula to compute the probability of observing a denial of service attack with a telescope. An updated result in [16] shows with another telescope that the previous model is to simple and that spoofed addresses are not uniformly randomly generated. An interesting work is presented in [15] and leads to the evaluation of the the aggressivity of denial of service attacks. Finally, the authors in [5] propose to use high interactive telescopes with emulated services in order to learn more application specific attacks. Network telescopes are also name darknets and the authors in [17] introduces the greynets which are small telescopes with some unused addresses scattered within a set of used IP addresses. They evaluate their efficiency depending on the number of probes, ie. the number of unused addresses.

The reference book in system administration [7] 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 [3], 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 [18] 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 abnormal activity is observed.

VI. Conclusion

In the work presented in this paper we were challenged by several research questions. Firstly, we needed a generic method to analyze both telescope and honeynet data. The main goal was to compare these two ways of gathering malicious network traffic. While a telescope monitors a large range of consecutive IP addresses, the honeynet monitors a limited set of IP addresses dispersed over the Internet. The amount of data is much higher for the telescope if compared to the honeyet. A second contribution of our work was to assess the utility of each method to collect network information. For instance, we have observed that a honeynet is sufficient for learning the distribution of source addresses, contrary to telescope for which a high redundancy might become an obstacle in the analysis.

On the other hand, both methods did provide similar results about the services/ports that are attacked, but the telescope is superior when detecting less frequently attacked services. This is quite obvious, due to the much higher data volume. Concerning the used attack tools, the honeynet permitted to show that these are more and more diversified and sophisticated. Regarding the misconfigurations, the network telescope and the honeynet are about equivalent for most of the studied cases.

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. The main difficulty encountered during our work is related to processing such large datasets: data counts to more than 200 GB and this task pushed our computational resources to their limits. Future work will address more advanced data mining and statistical analysis techniques.

Several papers individually analyzed either telescope data or honeynet data, but none had tried yet to compare these two data source simultaneously. Our work is to the best of our knowledge the first attempt to compare the two methods over the same time period.

Acknowledgment We thank Fabien Pouget and Marc Dacier from the Leurre.com project for their collaboration in the honeypot project. Moreover, we would like to thank Emile Aben and Colleen Shannon at CAIDA for granting us access to the telescope backscatter data.

References

[1] J. Francois, R. State, and O. Festor, "Large scale activity monitoring for distributed honeynets," in *ICIMP '07: Proceedings of the Second International Conference on Internet Monitoring and Protection*, 2007, p. 6.

[2] R. State, J. Francois, and O. Festor, "Tracking global wide configuration errors," in *IEEE / IST Workshop on Monitoring, Attack Detection and Mitigation*, Tubingen/Germany, 09 2006, H.: Information Systems.

[3] 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.

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

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

[6] C. Shannon, D. Moore, and E. Aben, "The caida backscatter-2004-2005 dataset - may 2004 - november 2005, http://www.caida.org/data/passive/backscatter_2004_2005 _dataset.xml."

[7] 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.

[8] 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.

[9] 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.

[10] A. R. W. Cheswick, S Bellowin, *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison Wesley, 1994.

[11] G. Roualland and J.-M. Saffroy, "http://ippersonality.sourceforge.net."

[12] IANA, "http://www.iana.org/assignments/icmp-parameters," 2005.

[13] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *ACM-SE 45: Proceedings of the 45th annual southeast regional conference*. New York, NY, USA: ACM, 2007, pp. 321–326.

[14] 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.

[15] 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.

[16] 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.

[17] W. Harrop and G. Armitage, "Defining and evaluating greynets (sparse darknets)," in *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 344–350.

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