Addressing the Attack Attribution Problem using Knowledge Discovery and Multi-criteria Fuzzy Decision-Making

Olivier Thonnard
Royal Military Academy
Polytechnic Faculty
Brussels, Belgium
olivier.thonnard@rma.ac.be

Wim Mees
Royal Military Academy
Polytechnic Faculty
Brussels, Belgium
wim.mees@rma.ac.be

Marc Dacier
Symantec Research
Sophia Antipolis
France
marc_dacier@symantec.com

ABSTRACT

In network traffic monitoring, and more particularly in the realm of threat intelligence, the problem of "attack attribution" refers to the process of effectively attributing new attack events to (un)-known phenomena, based on some evidence or traces left on one or several monitoring platforms. Real-world attack phenomena are often largely distributed on the Internet, or can sometimes evolve quite rapidly. This makes them inherently complex and thus difficult to analyze. In general, an analyst must consider many different attack features (or criteria) in order to decide about the plausible root cause of a given attack, or to attribute it to some given phenomenon. In this paper, we introduce a global analysis method to address this problem in a systematic way. Our approach is based on a novel combination of a knowledge discovery technique with a fuzzy inference system, which somehow mimics the reasoning of an expert by implementing a multi-criteria decision-making process built on top of the previously extracted knowledge. By applying this method on attack traces, we are able to identify large-scale attack phenomena with a high degree of confidence. In most cases, the observed phenomena can be attributed to so-called *zombie armies* - or botnets, i.e. groups of compromised machines controlled remotely by a same entity. By means of experiments with real-world attack traces, we show how this method can effectively help us to perform a behavioral analysis of those zombie armies from a long-term, strategic viewpoint.

Keywords

Intelligence monitoring and analysis, attack attribution.

1. INTRODUCTION

In the field of threat intelligence, "attack attribution" refers to the process of effectively attributing new attack events to known or unknown phenomena by analyzing the traces they have left on sensors or monitoring platforms deployed on the Internet. The objectives of such a process are twofold: *i)* to get a better understanding of the root causes of the observed attacks; and *ii)* to characterize emerging threats from a global viewpoint by producing a precise analysis of the modus operandi of the attackers on a longer time scale.

In this paper, we introduce a global threat analysis method to address this problem in a systematic way. We present a knowledge mining framework that enables us to identify and characterize large-scale attack phenomena on the Internet, based on network traces collected with very simple and easily deployable sensors. Our approach relies on a novel combination of knowledge discovery (by means of maximum cliques) and a multi-criteria decision-making algorithm that is based on a fuzzy inference system (FIS). Interestingly, a FIS does not need any training prior making inferences. Instead, it takes advantage of the previously extracted knowledge to make sound inferences, so as to attribute incoming attack events to a given phenomenon.

A key aspect of the proposed method is the exploitation of external characteristics of malicious sources, such as their spatial distributions in terms of countries and IP subnets, or the distribution of targeted sensors. We take advantage of these statistical characteristics to group events that seem a priori unrelated, whereas most current techniques used for anomalous traffic correlation rely only on the intrinsic properties of network flows (e.g., protocol characteristics, IDS alerts or signatures, firewall logs, etc) [1, 31].

Our research builds also on prior work in malicious traffic analysis, also referred to as Internet *background radiation* [17, 4]. We acknowledge also the seminal work of Yegneswaran et al. on "Internet situational awareness" [30], in which they explore ways to integrate honeypot data into daily network security monitoring. Their approach aims at providing tactical information, for daily operations, whereas our approach is more focused on strategic information revealing the long-term behaviors of large-scale phenomena. Furthermore, many of these large-scale phenomena are apparently related to the ubiquitous problem of *zombie armies* - or botnets, i.e. groups of compromised machines that are remotely controlled and coordinated by a same entity. Still today, zombie armies and botnets constitute, admittedly, one of the main threats on the Internet, and they are used for different kinds of illegal activities (e.g., bulk spam sending, online fraud, denial of service attack, etc) [3, 18]. While most previous studies related to botnets have focused on un-

CSI-KDD '09, June 28, Paris, France
Copyright 2009 ACM 978-1-60558-669-4 ...$10.00.

derstanding their inner working [23, 6, 2], or on techniques for detecting bots at the network-level [8, 9], we are instead more interested in studying the global behaviors of those armies from a strategic viewpoint, i.e.: how long do they stay alive on the Internet, what is their average size, and more importantly, how do they evolve over time with respect to different criteria such as their origins, or the type of activities (or scanning) they perform.

In Section 2, we present the first component of our method, namely the extraction of cliques of attackers. This step aims at discovering knowledge by identifying meaningful correlations in a set of attack events. In Section 3, we present a multi-criteria decision-making algorithm that is based on a fuzzy inference system. The purpose of this second component consists in combining intelligently the previously extracted knowledge, so as to build sequences of attack events that can be very likely attributed to the same global phenomena. Then, in Section 4, we present our experimental results and the kind of findings we can obtain by applying this analysis method to a set of attack events. Finally, we conclude in Section 5 and we suggest some future directions.

2. KNOWLEDGE DISCOVERY IN ATTACK TRACES

2.1 Introduction

We need first to introduce the notion of "attack event". Our dataset is made of network attack traces collected from a distributed set of sensors (e.g., server honeypots), which are deployed in the context of the *Leurre.com Project* [14, 22]. Since honeypots are systems deployed for the sole purpose of being probed or compromised, any network connection that they establish with a remote IP can be considered as malicious, or at least suspicious. We use a classical clustering algorithm to perform a first low-level classification of the traffic. Hence, each IP source observed on a sensor is attributed to a so-called *attack cluster* [21] according to its network characteristics, such as the number of IP addresses targeted on the sensor, the number of packets and bytes sent to each IP, the attack duration, the average inter-arrival time between packets, the associated port sequence being probed, and the packet payload (when available). Therefore, all IP sources belonging to a given attack cluster have left very similar network traces on a given sensor and consequently, they can be considered as having the same *attack profile*. This leads us then to the concept of attack event, which is defined as follows:

> An *attack event* refers to a subset of IP sources having the same attack profile on a given sensor, and whose coordinated activity has been observed within a specific time window.

Fig. 1 illustrates this notion by representing the time series (i.e., the number of sources per day) of three coordinated attack events observed on two different sensors in the same time interval, and targeting three different ports. The identification of those events can be easily automated by using the method presented in [20]. By doing so, we are able to extract interesting events from this spurious, nonproductive traffic collected by our sensors (previously termed "Internet background radiation" in [17]), and we can focus on the most important events that might originate from coordinated phenomena. In the rest of this Section, we show how to take advantage of different characteristics of such attack events to discover knowledge by means of an unsupervised clique-based clustering technique.

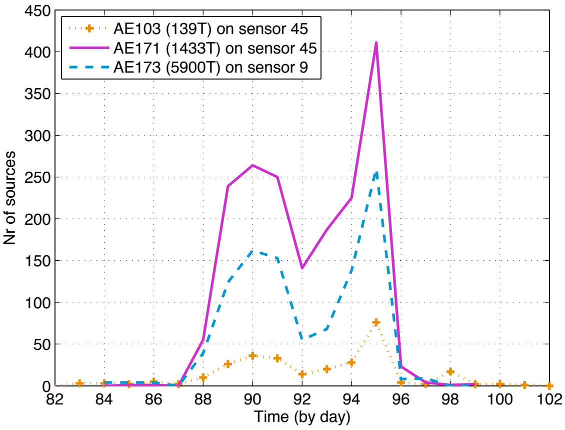

Figure 1: Illustration of 3 attack events observed on 2 different sensors, and targeting 3 different ports.

2.2 Defining Attack Characteristics

In most knowledge discovery applications, the first step consists in selecting certain key characteristics from the dataset, i.e., salient features that may (hopefully) provide meaningful *patterns* [11]. We give here an overview of different attack characteristics we have selected to perform the extraction of knowledge from our set of attack events. In this specific case, we consider these characteristics as useful to analyze the root causes of global phenomena observed on our sensors. However, we do not pretend that they are the only ones that could be used in threat monitoring, and other characteristics might certainly prove even more relevant in the future. For this reason, the framework is built such that other attack features could be easily included when necessary.

So, the two first characteristics retained are related to the *origins* of the attackers, i.e. their spatial distributions. First, the geographical location can be used to identify attack activities having a specific distribution of originating countries. Such information can be important to identify, for instance, botnets that are located in a limited number of countries. It is also a way to confirm the existence, or not, of so-called safe harbors for cybercriminals or hackers. Somehow related to the geographical location, the IP network blocks provide also an interesting viewpoint on the attack phenomena. Indeed, IP subnets can give a good indication of the spatial "uncleanliness" of certain networks, i.e., the tendency for compromised hosts (e.g., zombie machines) to stay clustered within unclean networks [5]. So, for each attack event, we can create a feature vector representing either the distribution of originating countries, or of IP addresses grouped by Class A-subnet (i.e., by /8 prefix).

The next attack characteristic deals with the *targets* of the attackers, namely the distribution of sensors that have been targeted by the sources. Botmasters may indeed send commands at a given time to all zombies to instruct them to start scanning (or attacking) one or several IP subnets, which of course will create coordinated attack events on specific sensors. Therefore, it seems important to look at relationships that may exist between attack events and the sensors they have been observed on. Since attack events are defined per sensor, we decided to group all strongly correlated attack events that occurred within the same time window of existence (as explained in [20]), and we then use each group of attack events to create the feature vector representing the proportion of sensors that have been targeted.

Besides the origins and the targets, the type of activity performed by the attackers seems also relevant to us. In fact, bot software is often crafted with a certain number of available exploits targeting a reduced set of TCP or UDP ports. In other words, we might think of each botnet having its own *attack capability*, which means that a botmaster will normally issue scan or attack commands only for vulnerabilities that he might exploit to expand his botnet. So, it seems to make sense to take advantage of this feature to look for similarities between the sequences of ports that have been targeted by the sources of the attack events. Let us remind that, in our low-level classification of the network traffic [21], each source is associated to the complete *sequence of ports* that it has targeted on a given sensor for the whole duration of the attack session (e.g., less than 24 hours), which allows us to compute and compare the distributions of port sequences for the observed attack events.

Finally, we have also decided to compute, for each pair of events, the ratio of common IP addresses. We are aware of the fact that, as time passes, some zombie machines of a given botnet might be cured while others may get infected and join the botnet. Additionally, certain ISPs apply a quite dynamic policy of IP address allocation to residential users, which means that bot-infected machines can have different IP addresses when we observe them at different moments. Nevertheless, and according to our domain experience, it is reasonable to expect that if two distinct attack events have a high percentage of IP addresses in common, then the probability that those two events are somehow related to the same global phenomenon is increased (assuming that the time difference between the two events is not too large).

2.3 Extracting Cliques of Attackers

2.3.1 Principles

In our global threat analysis method, we have developed a knowledge discovery component that involves an unsupervised graph-theoretic correlation process. The idea consists in discovering all groups of highly similar attack events (through their corresponding feature vectors) in a reliable and consistent manner, and for each attack characteristic that can bring an interesting viewpoint on the root causes.

In a clustering task, we typically consider the following steps [11]: *i)* feature selection and/or extraction; *ii)* definition of a similarity measure between pairs of patterns; *iii)* grouping similar patterns; *iv)* data abstraction (if needed), to provide a compact representation of each cluster; and *v)* the assessment of the clusters quality and coherence.

In the previous Section, we have already described the attack features that are of interest in this paper; so now we need to measure the similarity between two such input vectors (or distributions, in our case). Clearly, the choice of a similarity metric is very important, as it has an impact on the properties of the final clusters, such as their size, quality, and consistency. To reliably compare the kind of empirical distributions mentioned here above, we have chosen to rely on strong statistical distances. As we do not know the real underlying distribution from which the observed samples were drawn, we use non-parametric statistical tests, such as Pearson's χ^2, to determine whether two one-dimensional probability distributions differ in a significant way (with a significance level of 0.05). The resulting *p-value* is then validated against the Jensen-Shannon divergence (JSD) [15], which derives itself from the Kullback-Leibler divergence [12]. Let p_1 and p_2 be for instance two probability distributions over a discrete space X, then the K-L divergence of p_2 from p_1 is defined as:

$$D_{KL}(p_1||p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}$$

which is also called the information divergence (or *relative entropy*). D_{KL} is commonly used in information theory to measure the difference between two probability distributions p_1 and p_2, but it is not considered as a *true* metric since it is not symmetric, and does not satisfy the triangle inequality. For this reason, we can also define the Jensen-Shannon divergence as:

$$JS(p_1, p_2) = \frac{D_{KL}(p_1||\bar{p}) + D_{KL}(p_2||\bar{p})}{2}$$

where $\bar{p} = (p_1 + p_2)/2$. In other words, the Jensen-Shannon divergence is the *average* of the KL-divergences to the *average distribution.* The JSD has the following notable properties: it is always bounded and non-negative; $JS(p_1, p_2) = JS(p_2, p_1)$ (symmetric), and $JS(p_1, p_2) = 0$ when $p_1 = p_2$ (idempotent). To be a true metric, the JSD must also satisfy the triangular inequality, which is not true for all cases of (p_1, p_2). Nevertheless, it can be demonstrated that the *square root* of the Jensen-Shannon divergence is a true metric [7], which is what we need for our application.

Finally, we take advantage of those similarity measures to group all attack events whose distributions look very similar. We simply use an unsupervised graph-based approach to formulate the problem: the vertices of the graph represent the patterns (or feature vectors) of all attack events, and the edges express the similarity relationships between those vertices, as calculated with the distance metrics described here above. Then, the clustering is performed by extracting so-called *maximal cliques* from the graph, where a maximal clique is defined as an induced sub-graph in which the vertices are fully connected and it is not contained within any other clique. To perform this unsupervised clustering, we use the *dominant sets* approach of Pavan et al. [19], which proved to be an effective method for finding maximal *weighted* cliques. This means that the weight of every edge (i.e., the relative similarity) is also taken into consideration by the algorithm, as it seeks to discover maximal cliques whose total weight is maximized. This generalization of the MCP is also known as the maximum weight clique problem (MWCP). We refer the interested reader to [27, 26] for a more detailed description of this clique-based clustering technique applied to our honeynet traces.

2.3.2 Some Experimental Clique Results

Our data set comes from a 640-day attack trace obtained with the *Leurre.com* honeynet in the time period from September 2006 to June 2008. This trace was collected by 36 platforms located in 20 different countries and belonging to 18 different class A-subnets. We have selected only the most prevalent types of activities observed on the sensors, i.e. about 130 distinct attack profiles for which an activity involving a sufficient number of IP sources had been observed at least once on a given day during the whole period. This data set comprises totally 1,195,254 distinct sources, which have sent about 3,423,577 packets to the sensors. By using the technique described in [20], we have extracted 351 attack events that were somehow coordinated on at least two different sensors. This reduced set of attack events still accounts for 282,363 unique sources (23.6 % of the data set), or 741,349 packets (21.5%).

For the set of attack characteristics considered above, we applied our clique-based clustering on those attack events. Table 1 on page 5 presents a high-level overview of the cliques obtained for each attack dimension separately. As we can see, a relatively high volume of sources could be classified into cliques for each dimension. The last colon with the most prevalent patterns gives an indication of which countries or class A-subnets (e.g., originating or targeted IP subnets) are most commonly observed in the cliques that lie in the upper quartile with respect to the number of sources. Interestingly, it seems that many coordinated attack events are coming from a given IP subspace. Regarding the targeted platforms, several cliques involve a single class A-subnet. About the type of activities, we can observe some commonly targeted ports (e.g., Windows ports used for SMB or RPC, or SQL and VNC ports), but also a large number of uncommon high TCP ports that are normally unused on standard (and clean) machines (such as 6769T, 50286T, 9661T, . . .). A non-negligeable volume of sources is also due to UDP spammers targeting Windows Messenger popup service (ports 1026 to 1028/UDP).

2.4 Consolidation of the Knowledge

In order to assess the consistency of the resulting cliques of attack events, it can be useful to see them charted on a two-dimensional map so as to *i)* verify the proximities among clique members (*intra-clique* consistency), and *ii)* understand potential relationships between *different* cliques that are somehow related (i.e. *inter-clique* relationships). Moreover, the statistical distances used to compute those cliques make them intrinsically coherent, which means also that certain cliques of events may be somehow related to each other, although they were separated by the clique algorithm.

Since most of the feature vectors we are dealing with have a high number of variables (e.g., a geographical vector has more than 200 country variables), obviously the structure of such high-dimensional data set cannot be displayed directly on a 2D map. Multidimensional scaling (MDS) is a set of methods that can help to address this problem. MDS is based on dimensionality reduction techniques, which aim at converting a high-dimensional dataset into a two or three-dimensional representation that can be displayed, for example, in a scatter plot. The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map. As a consequence, MDS allows an analyst to visualize how far observations are from each other for different kinds of similarity measures, which in turn can deliver insights into the underlying structure of the high-dimensional dataset.

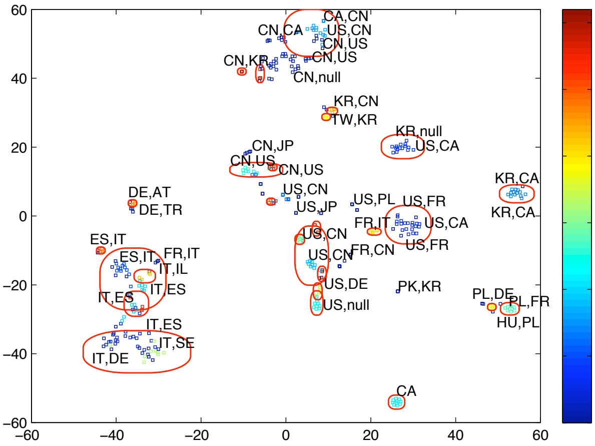

Figure 2: Visualization of geographical cliques of attackers. The coloring refers to the different cliques and the red circles indicate their sizes on the low-D map. The superposed text labels indicate *only the two top attacking countries* for some of the data points.

Because of the intrinsic non-linearity of real-world data sets, we applied a recent MDS technique called *t-SNE* to visualize each dimension of the data set, and to assess the consistency of the cliques results. t-SNE [28] is a variation of *Stochastic Neighbour Embedding*; it produces significantly better visualizations than other MDS techniques by reducing the tendency to crowd points together in the centre of the map. Moreover, this technique has proven to perform better in retaining both the local and global structure of real, high-dimensional datasets in a single map, in comparison to other non-linear dimensionality reduction techniques such as Sammon mapping, Isomaps or Laplacian Eigenmaps [10]. Stochastic Neighbor Embedding aims at minimizing a cost function that is based on the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method. t-SNE improves further this technique by using an initial Student-t distribution, rather than a Gaussian, to compute the similarity between two points in the low-dimensional space (which tends to alleviate the problem of "crowding" points in the center of the map, see [28] for a detailed explanation).

Figure 2 shows the resulting two-dimensional plot obtained by mapping the geographical vectors on a 2D map using t-SNE. Each datapoint on this map represents the geographical distribution of a given attack event. The coloring refers to the clique membership of each event, as obtained previously by applying the clique-based clustering, and the dotted circles indicate the clique sizes. We could easily verify that two adjacent events on the map have highly similar geographical distributions (even from a statistical viewpoint), while two distant events have clearly nothing in common in terms of originating countries. Quite surprisingly, the resulting mapping is far from being chaotic; it presents a relatively sparse structure with clear datapoint groupings, which means also that most of those attack events present very tight relationships regarding their origins. Due to the

Attack Dimension	Nr of Cliques	Max.size (nr events)	Min.size (nr events)	Volume of sources (%)	Most prevalent patterns found in the cliques$^{(1)}$
Geolocation	31	40	3	84.4	⟨CN,CA,US,FR,TW⟩, ⟨IT,ES,FR,SE,DE,IL⟩, ⟨KR,US,BR,PL,CN,CA⟩ ⟨US,JP,GB,DE,CA,FR,CN,KR⟩, ⟨US,FR,JP,CN,DE,ES,TW⟩, ⟨CA,CN⟩ ⟨PL,DE,ES,HU,FR⟩
IP Subnets (Class A)	25	51	3	91.2	⟨87,82,151,83,84,81,85,213⟩, ⟨222,221,60,218,58,24,124,121,219,82,220⟩ ⟨201,83,200,24,211,218,89,124,61,82,84⟩, ⟨24,60⟩ ⟨83,84,85,80,88⟩, ⟨193,195,201,202,203,216,200,61,24,84,59⟩
Targeted platforms	17	86	2	70.1	⟨202⟩, ⟨88, 192⟩, ⟨195⟩, ⟨193⟩, ⟨194⟩ ⟨129, 134, 139, 150⟩, ⟨24, 213⟩
Port sequences	22	66	4	93.2	⟨I⟩, ⟨1433T⟩, ⟨I-445T⟩, ⟨5900T⟩, ⟨1026U⟩, ⟨135T⟩, ⟨50286T⟩ ⟨I-445T-139T-445T-139T-445T⟩, ⟨6769T⟩, ⟨1028U-1027U-1026U⟩

Table 1: Some experimental clique results obtained from a honeynet dataset collected from Sep 06 until June 08. (1) **the given patterns represent the average distributions for the most prevalent cliques, i.e. the ones lying in the upper quartile in terms of number of sources. For the IP subnets (resp. targeted platforms), the numbers refer to the distributions of originating (resp. targeted) class A-subnets.**

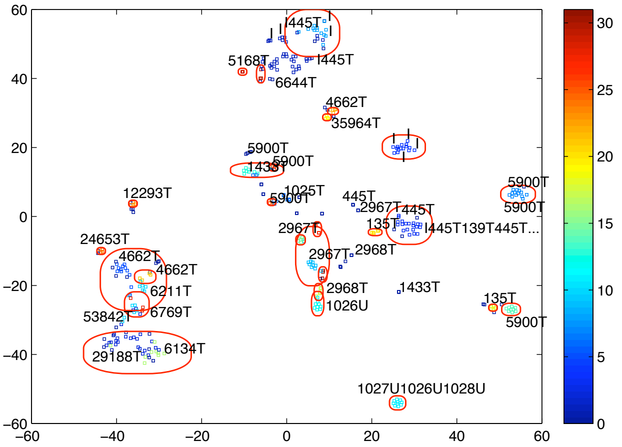

Figure 3: Same visualization of the geographical cliques of attackers as Fig 2, but here the superposed text labels indicate the port sequences targeted by the attackers.

strict statistical distances used to calculate cliques, this kind of correlation can hardly be obtained by chance only.

Similar "semantic mapping" can naturally be obtained for the other dimensions (e.g., subnets, platforms, etc), so as to help assessing the quality of other cliques of attackers. To conclude this Section, Figure 3 shows the same geographical mapping on which the port sequences of several attack events have been superposed on top of the datapoints. This can help to visualize unobvious relationships among different types of activities and their origins, and it leads also to the natural intuition that an intelligent algorithm could potentially leverage the results of this knowledge discovery process, by combining efficiently different sets of cliques.

3. MULTI-CRITERIA DECISION-MAKING

3.1 Requirements and Motivation

The decision-support component of our method shall take advantage of the knowledge obtained via the extraction of cliques, and of the global semantic mappings obtained through dimensionality reduction. The final objective consists in re-constructing *sequences of attack events* that can be attributed with a high confidence to the same root phenomenon in function of multiple criteria. In other words, we want to build an inference engine that takes as input the extracted knowledge to classify incoming attack events into either "known phenomena", or otherwise to identify a new phenomenon when needed (e.g., when we observe the first attack event of a new zombie army). There exists certainly many different classification algorithms that are able to map multiple input features to multiple output classes, even for complex, non-linear mappings, such as Support Vector Machines, Artificial Neural Networks, etc. However, we are confronted to specific constraints that do not allow us to use this type of supervised machine learning techniques. First, we have *a priori* zero-knowledge of the expected output, which means that we can not provide training samples showing the characteristics of the output we are looking for. Secondly, we want to include some domain knowledge to specify which type of combinations we expect to be promising in the root cause identification. Third, the inference system must be flexible enough to allow additional criteria to be used in the future, so as to further improve the inference capabilities. Finally, we favor the "white-box" approach having a transparent reasoning process, which allows an expert to understand the reasons (i.e., the combinations of criteria) for which the system has grouped a given set of events into the same root phenomenon.

Although large-scale phenomena on the Internet are complex and dynamic, our intuition is that two consecutive attack events should be linked to the same root phenomenon if and only if they share at least two different attack characteristics. That is, we want to build a decision-making process that will attribute two attack events to the same phenomenon when the events features are "close enough" for any combination of at least two attack dimensions out of the complete set of criteria: $\{origins, targets, activity, common_{IP}\}$. So, we hypothesize that real-world phenomena may perfectly evolve over time, which means that two consecutive attack events of the same zombie army must not necessarily have all their attributes in common. For example, the bots composition of a zombie army may evolve over time because of the cleaning of infected machines and the recruitment of new bots. From our observation viewpoint, this will translate into a certain shift in the IP subnet distribution of the zombie machines for subsequent attack events of this army

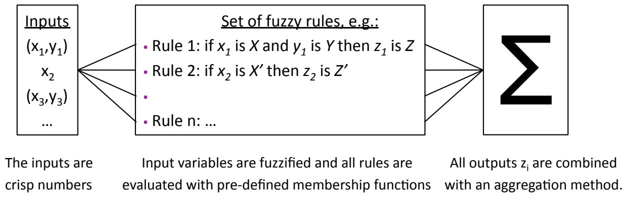
Inputs
(x1,y1)
x2
(x3,y3)
...
Set of fuzzy rules, e.g.:
Rule 1: if x1 is X and y1 is Y then z1 is Z
Rule 2: if x2 is X' then z2 is Z'
Rule n: ...
Σ
The inputs are crisp numbers
Input variables are fuzzified and all rules are evaluated with pre-defined membership functions
All outputs zi are combined with an aggregation method.

Figure 4: Main components of a Fuzzy System.

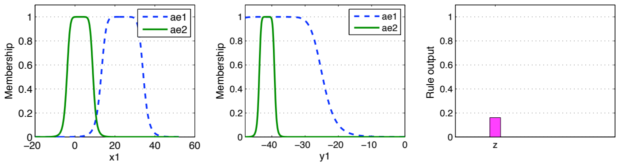

Figure 5: Fuzzy rule evaluation.

(and thus, most probably different cliques w.r.t. the origins). Or, a zombie army may be instructed to scan several consecutive IP subnets in a rather short interval of time, which will lead to the observation of different events having highly similar distributions of originating countries and subnets, but those events will target completely different sensors, and may eventually use different exploits (hence, targeting different port sequences).

On the other hand, we consider that only one correlated attack dimension is not sufficient to link two attack events to the same root cause, since the result might then be due to chance only (e.g., a large proportion of attacks originate from some large or popular countries, certain Windows ports are commonly targeted, etc). However, by combining intelligently several attack viewpoints, we can reduce considerably the probability that two attack events would be attributed to the same root cause whereas they are in fact unrelated.

3.2 Fuzzy Inference Systems

We still need to formally define what is the "relatedness degree" between two attack events, certainly when they do not belong to a same clique but are somehow "close" to each other. Intuitively, attack events characteristics in the real world have unsharp boundaries, and the membership to a given phenomenon can be a matter of degree. For this reason, we have developed a decision-making process that is based on a fuzzy inference system (FIS). The mathematical concepts behind fuzzy reasoning are quite simple and intuitive; in fact, it aims at reproducing the reasoning of a human expert with very simple mathematical functions. Fuzzy inference is thus a convenient way to map an input space to an output space with a flexible and extensible system, and using the codification of common sense and expert knowledge. The mapping then provides a basis from which decisions can be made.

The main components of an inference system are sketched in Fig. 4. To map the input space to the output space, the primary mechanism is a list of if-then statements called rules, which are evaluated in parallel, so the order of the rules is unimportant. Instead of using crisp variables, all inputs are *fuzzified* using membership functions in order to determine the degree to which the input variables belong to each of the appropriate fuzzy sets. If the antecedent of a given rule has more than one part (i.e., multiple 'if' statements), a fuzzy logical operator is applied to obtain one number that represents the result of the antecedent for that rule. For example, the fuzzy OR operator simply selects the maximum of the two values. The results of all rules are then combined and distilled into a single, crisp value that can be used to make a decision. This aggregation process can be done in two different ways. Mamdani's inference [16] expects the output membership functions to be also fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification by computing for instance the centroïd of the output function. Whereas in a Sugeno-type inference system [25], the output membership functions are either linear or constant. The general form of a rule in a Sugeno fuzzy model is: if $Input_1$ is x and $Input_2$ is y then Output is $z = a.x + b.y + c$. For a zero-order Sugeno model, the output level z is a constant (a=b=0). The output level z_i of each rule is weighted by the firing strength w_i of the rule. The most common way to calculate the final output of the system is the weighted average of all rule outputs:

$$Final\ output = \frac{\sum_i w_i.z_i}{\sum_i w_i}$$

When it is possible to model a fuzzy system using Sugeno-type inference, the defuzzification and aggregation process is thus greatly simplified and much more efficient than with Mamdani's inferences, which is why we used a Sugeno-type system to model each attack phenomenon.

Concretely, we use the knowledge obtained from the extraction of cliques to build the fuzzy rules that describe the behavior of each phenomenon. The characteristics of new incoming attack events are then used as input to the fuzzy systems that model the phenomena identified so far. In each of those fuzzy systems, the features of the *most recent* attack event shall define the current parameters of the membership function used to evaluate the following simple rules: if x_i is *close* AND if y_i is *close* then z_i is *related*, $\forall i \in \{geo, subnets, targets, portsequence\}$. Fig 5 gives a graphical representation of how such a rule is evaluated for the subnets of origins of two given attack events. Since this characteristic is represented by a 2D mapping, we can see the result of evaluating the relative position of the events according to both dimensions (x, y). Each membership function is maximal within the cliques, then it decreases smoothly to take into account the fuzziness of real-world phenomena. In this case, the antecedents of the rule hold respectively 0.16 and 1.0, which results in an output of 0.16 (since a logical AND in fuzzy logic corresponds to the MIN operator).

So, the membership functions referred to as "is close" in the fuzzy rules are defined by the characteristics of the cliques to which the attack events belong. The calculation of the rule output $z_i \in [0, 1]$ is just the intersection between the two curves, which quantifies the inter-relationship between the cliques (and hence, between the attack events). Similarly, we can evaluate the fuzzy rules for the other dimensions considered in the inference system. For the last dimension, i.e. the common IP's, we use a static membership function whose input is the common IP ratio calculated between the two events. Fig 6 represents this static membership function, where we can see the output Z_{IP} increasing

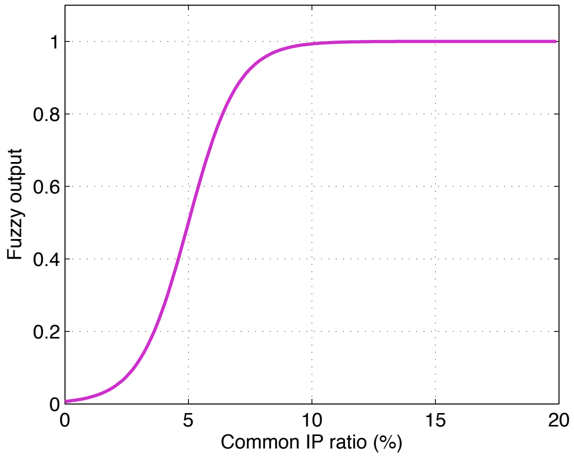

Figure 6: Common IP Membership function.

smoothly as the ratio of common IP addresses increases from 0 to 10%, where Z_{IP} is then maximal. This curve is actually drawn from our knowledge, or domain experience, in monitoring malicious traffic.

Note that, initially, the inference engine has no knowledge, so the first incoming attack event will create the first phenomenon. Then, each time a new event could not be attributed to an existing phenomenon, the inference engine will create a new fuzzy system to model this new emerging phenomenon. The inference engine is thus self-adaptive by design.

3.3 Multi-criteria Decision-making

Having formally defined how to evaluate the output of each rule, for each phenomenon, a last problem remains regarding the weighted average that is used as aggregation function in a classical Sugeno inference system. In fact, it does not allow us to express that certain *combinations of criteria* (or rule outputs) must be somehow prioritized, as previously described in the requirements. We need thus to introduce another type of multi-criteria aggregation function that allows to model more complex requirements such as "most of", or "at least two" criteria to be satisfied in the overall decision function. Yager has introduced in [29] a special type of operator called Ordered Weighted Aggregation (OWA), which allows to include some relationships between multiple criteria in the aggregation process. An OWA operator provides an aggregation function for criteria whose result lies between the classical "and" and "or" operators, which are in fact the two extreme cases. Assume $Z_1, Z_2, \ldots, Z_n$ are n criteria of concern in our multi-criteria problem. For each criteria, $Z_i(x) \in [0, 1]$ indicates the degree to which x satisfies that criteria, which corresponds in our case to the rules output of a given fuzzy system. Then, we define a mapping function $F : I^n \rightarrow I$ where $I = [0, 1]$ as an OWA operator of dimension n, if associated with F is a weighting vector $W = (W_1, W_2, \ldots, W_n)$ such that

1. $W_i \in [0, 1]$
2. $\sum_i W_i = 1$

and where

$$F(z_1, z_2, \ldots, z_n) = W_1.z'_1 + W_2.z'_2 + \ldots + W_n.z'_n$$

with z'_i being the ith largest element in the collection $z_1, ..., z_n$. That is, Z' is an ordered vector composed of the elements of Z put in descending order, which means that the weights W_i are associated with a particular ordered position rather than a particular element. Yager [29] has carefully studied the mathematical foundations of OWA operators, and he demonstrated that such operators have the desired properties such as monotonicity, generalized commutativity, associativity and idempotence. To define the weights W_i to be used, Yager suggests two possible approaches: either to use some learning mechanism with sample data and a regression model, or to give some semantics or meaning to the W_i's by asking a decision-maker to provide directly those values. We selected the latter approach by defining the weighting vector as $W = (0.1, 0.35, 0.35, 0.1, 0.1)$, which translates our intuition about the dynamic behaviors of large-scale phenomena. It can be interpreted as: "at least three criteria must be satisfied, but the first criteria is of less importance compared to the 2nd and 3rd ones". These values were carefully chosen in order to avoid the grouping of unrelated events when, for example, two events are coming from popular countries and targeting common (Windows) ports in the same interval of time, but those events are in reality not related to the same phenomenon. In this worst-case scenario, we can imagine that the ordered vector of criteria (obtained from the evaluation of the fuzzy rules) could be something similar to $Z = (0.3, 0.1, 0, 1, 0)$. That is, we have a high correlation for the targeted port sequences ($z_4 = 1$), and we have then some weak correlation (due to chance) for the geographical origins ($z_1 = 0.3$) and also for the subnets of origins ($z_2 = 0.1$). By applying our weighting vector W to $Z' = (1, 0.3, 0.1, 0, 0)$, we get as final decision value $F = 1 * 0.1 + 0.3 * 0.35 + 0.1 * 0.35 = 0.24$. By considering other scenarios, we can verify that the values of the weighting vector W work as expected, i.e. it minimizes the final output value in these cases. Moreover, these considerations enable us also to fix our decision threshold to an empirical value of about 0.25. That is, when the final output value F lies under this threshold, we will reject the attribution of the attack event under scrutiny to the current phenomenon whose fuzzy system is being evaluated. Finally, when several fuzzy systems provide an output value lying above the threshold, we will obviously chose the highest one to attribute the event; however, this case was rarely observed in our experiments. There exists certainly other alternatives for choosing the W_i's, but according to our experimental results, this choice proved to be very effective in identifying sequences of attack events having the same root cause.

4. BEHAVIORAL ANALYSIS OF GLOBAL PHENOMENA

4.1 Main Characteristics

In this Section, we provide some experimental results obtained by applying our multi-criteria inference method to the same set of attack events we already introduced in Section 2.3 (clique analysis). As already mentioned, these experimental results only aim at validating the applicability and usefulness of the method proposed. They do not pretend to offer a complete view of all possible phenomena observable on the Internet. At the contrary, they show that, even with a limited number of data sources, it is possible to observe and reason about a couple of interesting phenomena. Furthermore, these anecdotal, yet representative, examples show that our method helps in characterizing their root cause, i.e., in addressing the attack attribution issue.

So, over the whole collection period (640 days), we found about 32 global phenomena. In total, 348 attack events (99% of our data set) could be attributed to a given large-scale phenomenon. An in-depth analysis has revealed that most of those phenomena (apart from the noisy network worm W32.Rahack.H [24], also known as W32/Allaple) are quite likely related to *zombie armies*, i.e., groups of compromised machines belonging to the same botnet(s). We conjecture this for the following main reasons: *i)* the apparent coordination of the sources, both in time (i.e., coordinated events on several sensors) and in the distribution of tasks (e.g., scanners versus attackers); *ii)* the short durations of the attack events, typically a few days only, whereas "classical" worms tend to spread over longer, continuous periods of time; *iii)* the absence of known classical network worm spreading on many of the observed port sequences; and *iv)* the source growing rate, which has a sort of exponential shape for worms and is somehow different for botnets [13].

To illustrate the results, Table 2 on page 10 presents an overview of some global phenomena found in our dataset. Thanks to our method, we are able to characterize precisely the behaviors of the identified phenomena or zombie armies. Hence, we found that the largest army had in total 57 attack events comprising 69,884 sources, and could survive for about 112 days. The longest lifetime of a zombie army observed so far was still 586 days. Fig. 7 shows the cumulative distributions (CDF) of the lifetime and size of the identified armies. Those figures reveal some interesting aspects of their global behaviors: according to our observations, at least 20% of the zombie armies had in total more than ten thousand observable[1] sources during their lifetime, and the same proportion of armies could survive on the Internet for at least 250 days. On average, zombie armies have a total size of about 8,500 observed sources, a mean number of 658 sources per event, and their mean survival time is 98 days.

Regarding the origins, we observe some very persistent groups of IP subnets and countries of origin across many different armies. On Fig. 8, we can see the CDF of the sources involved in the zombie armies of Table 2, where the x-axis represents the first byte of the IPv4 address space. It appears clearly that malicious sources involved in those phenomena are highly unevenly distributed and form a relatively small number of tight clusters, which account for a significant number of sources and are thus responsible for a large deal of the observed malicious activities. This is consistent with other prior work on monitoring global malicious activities, in particular with previous studies related to measurements of Internet background radiation [4, 17, 31]. However, we are now able to show that there are still some notable differences in the spatial distributions of those zombie armies with respect to the average distribution over all sources (represented with the blue dashed line). In other words, certain armies of compromised machines can have very different spatial distributions, even though there is a large overlap between "zombie-friendly" IP subnets. Moreover, because of the dynamics of this kind of phenomena, we can even observe very different spatial distributions within a *same army* at different moments of its lifetime. This is a strong advantage of our analysis method that is more precise and enables us to distinguish *individual* phenomena, instead of global trends, and to follow their dynamic behavior over time.

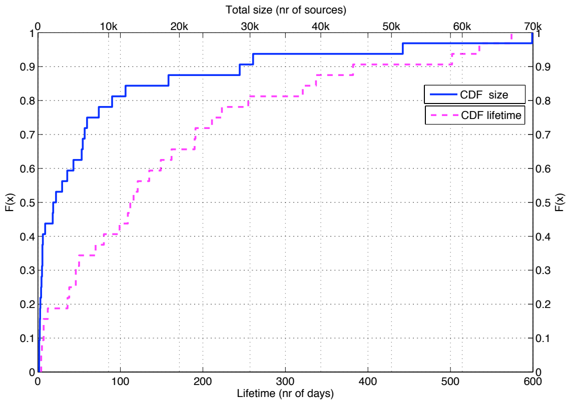

Figure 7: Empirical CDF of the size and lifetime of zombie armies.

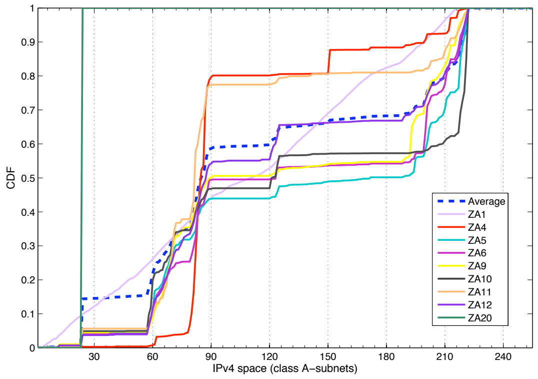

Figure 8: Empirical CDF of sources in IPv4 address space for the 9 zombie armies illustrated in Table 2.

Another interesting observation on Fig. 8 is related to the subnet CDF of ZA1 (uniformly distributed in the IPv4 space, which means randomly chosen source addresses) and ZA20 (a constant distribution coming exclusively from the subnet 24.0.0.0/8). A very likely explanation is that those zombie armies have used spoofed addresses to send UDP spam messages to the Windows Messenger service. So, this indicates that IP spoofing is still possible under the current state of filtering policies implemented by certain ISP's on the Internet.

Finally, in terms of *attack capability*, we observe that about 50% of the armies could target at least two completely dif-

[1] It is important to note that the sizes of the zombie armies given here only reflect the number of sources we could *observe* on our sensors; the actual sizes of those armies are most probably much larger, even though some churn effects (DHCP, NAT) could also affect these numbers.

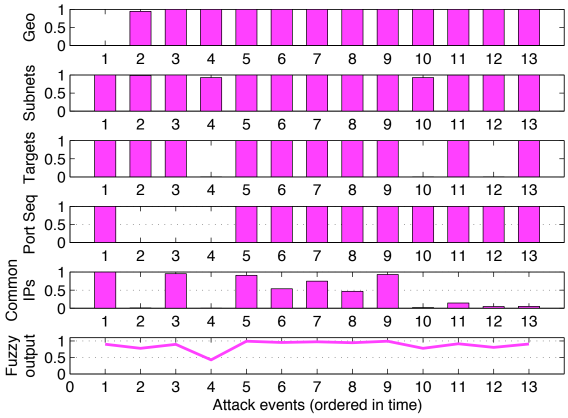

Figure 9: Output of the fuzzy inference system (z_i and $F(z_i)$) modeling the zombie army nr 12.

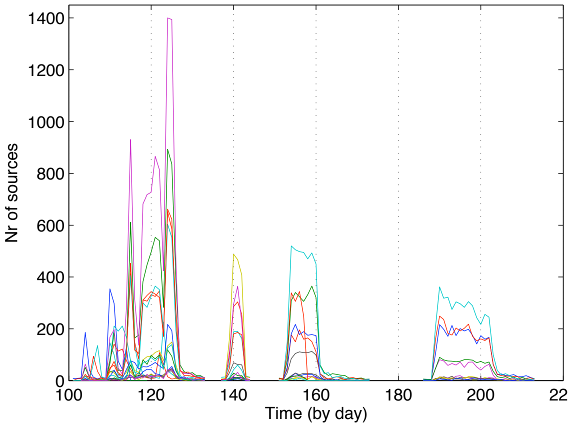

Figure 10: Time series of coordinated attack events for zombie army ZA10 (i.e., nr of sources observed by day).

ferent ports (thus, probably two different exploits, at least), and one army had even an attack capability greater than 10 (ZA4 in Table 2). At this stage, it is unclear why a zombie army would target such a large number of unusual, high TCP ports (12293T, 15264T, etc). A recurrent misconfiguration or P2P phenomenon is thus not excluded; but even in that case, it is very interesting to note that our method was able to attribute all those different events to the same root phenomenon, thanks to the combination of several statistical metrics.

4.2 Some Detailed Examples

In this Section, we further detail two zombie armies to illustrate some typical behaviors we could observe among the identified phenomena, e.g.:

i) a move (or drift) in the origins of certain armies (both geographical and IP blocks) during their lifetime;

ii) a large scan sweep by the same army targeting several consecutive class A-subnets;

iii) within a same army, multiple changes in the port sequences (or exploits) used by zombies to scan or to attack;

iv) a coordination between different armies.

Zombie army 12 (ZA12) is an interesting case in which we can observe the behaviors *ii)* and *iii)*. Fig. 9 represents the output of the fuzzy system modeling this phenomenon. Each bar graph represents the fuzzy output z_i for a given attack dimension, whereas the last plot shows the final aggregated output from which the decision to group those events together was made (i.e., $F(z_i)$). We can clearly see that the targets and the activities of this army have evolved between certain attack events (e.g., when the value of z_i is low). That is, this army has been scanning (at least) four consecutive class A-subnets during its lifetime (still 183 days), while probing at the same time three different ports on these subnetworks.

Then, the largest zombie army observed by the sensors (ZA10) has showed the behaviors *i)* and *iv)*. On Fig. 10, we can see that this army had four waves of activity during which it was randomly scanning 5 different subnets (note the almost perfect coordination among those attack events). When inspecting the subnet distributions of those different attack waves, we could clearly observe a drift in the origins of those sources, quite likely as certain machines were infected by (resp. cleaned from) the bot software. Finally, we found another smaller army (ZA11) that is clearly related to ZA10 (e.g., same temporal behavior, similar activity, same targets); but in this case, a different group of zombie machines, resulting in very different subnet CDF's on Fig. 8), was used to attack only specific IP addresses on our sensors, probably by taking advantage of the results given by the army of scanners (ZA10).

5. CONCLUSIONS

We have introduced a general analysis method to address the complex problem related to "attack attribution". Our approach is based on a novel combination of knowledge discovery and a multi-criteria fuzzy decision-making process. By applying this method, we have showed how apparently unrelated attack events could be attributed to the same global attack phenomenon, or to the same army of zombie machines operating in a coordinated manner. To the best of our knowledge, this is the first formal, systematic and rigorous method that enables us to identify and characterize precisely the behaviors of those large-scale attack phenomena. As future work, we envisage to extend our method to other data sets, such as high-interaction (eventually client) honeypot data, or malware data sets, and to include even more relevant attack features so as to improve further the inference capabilities of the system, and thus also our insights into malicious behaviors observed on the Internet.

Acknowledgments

This work has been partially supported by the European Commission through project FP7-ICT-216026-WOMBAT 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.

Id	Nr of events	Total size (nr sources)	Lifetime (nr days)	Targeted sensors (Class A- subnets)	Attack capability	Main origins (countries / subnets)
1	10	18,468	535	24.*,193.*,195.*,213.*	1026U	US,JP,GB,DE,CA,FR,CN,KR,NL,IT 69,128,195,60,81,214,211,132,87,63
4	82	26,962	321	202.*	12293T,15264T,18462T,25083T,25618T,28238T,29188T, 32878T,33018T,38009T,4152T,46030T,4662T,50286T,. . .	IT,ES,DE,FR,IL,SE,PL 87,82,83,84,151,85,81,88,80
5	13	9,644	131	195.*	135T,139T,1433T,2968T,5900T	CN,US,PL,IN,KR,JP,FR,MX,CA 218,61,222,83,195,221,202,24,219
6	15	51,598	>1 year	> 7 subnets	ICMP (W32.Rahack.H / Allaple)	KR,US,BR,PL,CN,CA,FR,MX,TW 201,83,200,24,211,218,89,124
9	23	11,198	218	192.*,193.*,194.*	2967T,2968T,5900T	US,CN,TW,FR,DE,CA,BR,IT,RU 193,200,24,71,70,213,216,66
10	57	69,884	112	128.*,129.*,134.*,139.*,150.*	I-I445T	CN,CA,US,FR,TW,IT,JP,DE 222,221,60,218,58,24,70,124
11	14	2,636	110	129.*,134.*,139.*,150.*	I-445T-139T-445T-139T-445T	US,FR,CA,TW,IT 82,71,24,70,68,88,87
12	14	27,442	183	192.*,193.*,194.*,195.*	1025T,1433T,2967T	US,JP,CN,FR,TR,DE,KR,GB 218,125,88,222,24,60,220,85,82
20	10	30,435	337	24.*, 129.*, 195.*	1026U,1026U1028U1027U,1027U	CA,CN 24,60

Table 2: Overview of some large-scale phenomena found in a honeynet dataset collected from Sep 06 until Jun 08.

6. REFERENCES

[1] Paul Barford and David Plonka. Characteristics of network traffic flow anomalies. In *In Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2001.

[2] Paul Barford and Vinod Yegneswaran. *An Inside Look at Botnets*. Advances in Information Security. Springer, 2006.

[3] David Barroso. Botnets - the silent threat. In *European Network and Information Security Agency (ENISA)*, November 2007.

[4] Zesheng Chen, Chuanyi Ji, and Paul Barford. Spatial-temporal characteristics of internet malicious sources. In *Proceedings of INFOCOM*, 2008.

[5] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane. Using uncleanliness to predict future botnet addresses. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 93–104, New York, NY, USA, 2007. ACM.

[6] Evan Cooke, Farnam Jahanian, and Danny McPherson. The Zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005 Workshop)*, Cambridge, MA, July 2005.

[7] B. Fuglede and F. Topsoe. Jensen-shannon divergence and hilbert space embedding. pages 31–, June-2 July 2004.

[8] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium*, 2008.

[9] Guofei Gu, Junjie Zhang, and Wenke Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, February 2008.

[10] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 833–840, 2003.

[11] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall advanced reference series, 1988.

[12] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics 22: 79-86.*, 1951.

[13] Wenke Lee, Cliff Wang, and David Dagon, editors. *Botnet Detection: Countering the Largest Security Threat*, volume 36 of *Advances in Information Security*. Springer, 2008.

[14] C. Leita, V.H. Pham, O. Thonnard, E. Ramirez-Silva, F. Pouget, E. Kirda, and Dacier M. The Leurre.com Project: Collecting Internet Threats Information Using a Worldwide Distributed Honeynet. In *Proceedings of the WOMBAT Workshop on Information Security Threats Data Collection and Sharing, WISTDCS 2008*. IEEE Computer Society press, April 2008.

[15] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, Jan 1991.

[16] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Hum.-Comput. Stud.*, 51(2):135–147, 1999.

[17] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40, New York, NY, USA, 2004. ACM.

[18] Markus Kötter Georg Wicherski Paul Bächer, Thorsten Holz. Know your enemy: Tracking botnets. In *http://www.honeynet.org/papers/bots/*.

[19] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[20] V. Pham, M. Dacier, G. Urvoy Keller, and T. En Najjary. The quest for multi-headed worms. In *DIMVA 2008, 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, July, 2008, Paris, France*, Jul 2008.

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

[22] The Leurre.com Project. http://www.leurrecom.org.

[23] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 41–52, New York, NY, USA, 2006. ACM.

[24] Symantec Security Response. W32.rahack.h, [april 2009].

[25] Michio Sugeno. *Industrial Applications of Fuzzy Control.* Elsevier Science Inc., New York, NY, USA, 1985.

[26] Olivier Thonnard and Marc Dacier. A framework for attack patterns' discovery in honeynet data. *DFRWS 2008, 8th Digital Forensics Research Conference, August 11- 13, 2008, Baltimore, USA*, 2008.

[27] Olivier Thonnard and Marc Dacier. Actionable knowledge discovery for threats intelligence support using a multi-dimensional data mining methodology. In *ICDM'08, 8th IEEE International Conference on Data Mining series, December 15-19, 2008, Pisa, Italy*, Dec 2008.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing

data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.

[29] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190, 1988.

[30] V Yegneswaran, P Barford, and V Paxson. Using honeynets for internet situational awareness. In *Fourth ACM Sigcomm Workshop on Hot Topics in Networking (Hotnets IV)*, 2005.

[31] Vinod Yegneswaran, Paul Barford, and Johannes Ullrich. Internet intrusions: global characteristics and prevalence. In *SIGMETRICS*, pages 138–147, 2003.