IJoFCS (2012) 2, 8-20

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

DOI: 10.5769/J201202001 or http://dx.doi.org/10.5769/J201202001

Improved Parallel Approach to PCA Based Malicious Activity Detection in Distributed Honeypot Data

João Paulo Carvalho Lustosa da Costa, Edison Pignaton de Freitas, Antonio Manuel Rubio Serrano and Rafael Timóteo de Sousa Júnior
Department of Electrical Engineering
University of Brasilia (UnB)
joaopaulo.dacosta@ene.unb.br, edisonpignaton@unb.br,
toni.rubio.serrano@gmail.com, desousa@unb.br
URL: www.ppgee.unb.br

Abstract - This paper presents the R-D Akaike Information Criterion (AIC) and R-D Minimum Description Length (MDL) for automatically identification of malicious activities in honeypot networks based on state of the art model order selection schemes. Model order selection (MOS) schemes are frequently applied in several signal processing applications, such as RADAR, SONAR, communications, channel modeling, medical imaging, and parameters estimation of dominant multipath components from MIMO channel measurements. The proposal of this paper is a new application for these MOS schemes, which is the identification of the malicious activity in honeypots. The proposed blind automatic techniques are efficient and need neither previous training nor knowledge of attack signatures for detecting malicious activities. In order to achieve such results an innovative approach is considered which models network traffic data as signals and noise allowing the application of signal processing methods. The model order selection schemes are adapted to process network data, showing that the R-D Modified AIC and R-D MDL solve the limitations of other schemes because they can be applied to honeypot networks composed by several computers. The performance of the proposed solution is evaluated using the Probability of Detection (PoD).

I. INTRODUCTION

A honeypot system collects malicious traffic and general information on malicious activities directed towards the network where it is located [2]. It is a very useful tool as data source for intrusion detection systems [3] as well as a decoy for slowing down automated attacks [4]. Network administrators benefit from efficient algorithms for identifying malicious activity using the data stored on the log file of a honeypot. These algorithms are particularly useful to generate statistics, as well as to support intelligent intrusion and prevention systems and to provide important information to network administrators, so that they can take actions to protect the network based on the obtained statistics.

Despite being very useful and important in the network security context, representing a reliable and representative source of attacks iden-

Paper submitted on November 20th, 2012.

tification and threats [5], there is a drawback in using honeypots. The problem is related to the amount of data generated by such systems. Huge volumes of data traffic and network activity logs can be generated, making the efficient and automated analysis of such data a real challenge.

Identification and characterization of malicious activities in honeypot traffic data represent important research topics and have been addressed by a variety of approaches and techniques [7] [8] [9]. Classical methods typically employ data mining [8] [9] and regular file parsing [7] for detecting patterns which indicate the presence of specific attacks in the analyzed traffic and computing general statistical data on the collected traffic. An essential characteristic of these methods is the fact that they depend on previous knowledge of the attacks that are intended to be identified, as well as on the collection of significant quantities of logs to work properly.

Machine learning techniques have also been applied to honeypot data analysis and attack detection [10] yielding interesting results as these techniques are able to identify malicious activities. However, using this type of technique, a preparation period before use is needed, in which it is necessary to run several analysis cycles during a so called *learning* period in order to train the system to recognize a given set of attacks. Only after this period, solutions using this method are able to work effectively. The computational complexity of this learning process should be taken into account as a possible drawback. Furthermore, if the legitimate traffic patterns are altered by any legitimate reasons, machine learning based methods may yield a significant number of false positives, identifying honest connections as malicious activities. These systems are also prone to failure in cases in which specific attacks that were not included in the learning process resembles honest patterns. In such cases, these attacks are not detected, leading to false negatives.

Methods based on principal component analysis (PCA) [11] [12] appeared as a promising alternative to traditional techniques. PCA based methods identify the main groups of highly correlated indicators *i.e.* principal components which represent outstanding malicious activities in network traffic data collected at honeypots. These methods are based on the observation that attack traffic patterns are more correlated than regular network traffic. The advantage of this method is that it relies exclusively on statistical analysis of the collected data. This feature frees the PCA based solutions from analysis of previous information on the attacks to be detected, as well as there is no need for training to recognize attacks and separate them from legitimate traffic. These characteristics make PCA based honeypot data analysis methods suitable for automatic attack detection and traffic analysis. However, current PCA based methods [11] [12] still require human intervention, which besides of being impractical for automatic analysis, leads to errors such as false positives.

In order to solve this limitation, a new solution based on PCA and the state-of-the-art model order selection schemes [13] [14] was proposed in [1] [34] [35]. In [1] the proposed solution is based on the RADOI algorithm [21] in combination with prewhitening [36], having the limitations that this technique does not works when there is only legitimate traffic. In [34] was proposed the use of the Modified Exponential Fitting Test (M-EFT) [23][14][13], which allows detecting legitimate and malicious traffic. Also, as well as the case of RADOI, the M-EFT can detect the quantity of malicious traffic in the connections [34].

However, the solutions presented in [1] and [34] are only useful for the case in which there is only one computer as a honeypot, but it is also possible to use several computers that forms a honeypot network. For this scenario, it was proposed in [35] the use of the global eigenvalues that, as in previous cases, for working in an automatic way it needs to be used in combination with a model-order selection technique. In this paper, we propose the use of the R-D Akaike Information Criterion and R-D Minimum Description length to implement automatic detection of malicious traffic for the case of a honeypot network composed by several computers. The performance of the proposed solution is evaluated in terms of Probability of Detection (PoD).

The remainder of this paper is organized as follows. Section II formally introduces the concept of honeypots, discuss classical analysis methods and presents an analysis of related work on PCA based methods for honeypot data analysis. In Section III the notation used in this paper is defined. In Section IV, we describe the dataset preprocessing method through which we transform the data before Model Order Selection (MOS). In Section V, we introduce classical MOS and also state-of-the-art schemes and propose our analysis method based on R-D AIC and R-D MDL. Section VI presents the proposal for improving the performance in parallel environments, while Section VII presents the experimental results validating the proposal. Section VIII concludes the paper presenting directions for future work.

II. Related Works

This section introduces the concept of honeypot systems and discuss the several methods used for obtaining and analysing data in such systems. Special attention is given to methods based on principal component analysis, which are the focus of our results.

A honeypot is generally defined as an information system resource whose value lies in unauthorized or illicit use of that resource [2], although various definitions exist for specific cases and applications. Honeypot systems are designed to attract the attention of malicious users in order to be actively targeted and probed by potential attackers, differently from intrusion detection systems (IDS) or firewalls, which protect the network against adversaries. Generally, network honeypot systems contain certain vulnerabilities and services which are commonly targeted by automated attack methods and malicious users, capturing data and logs regarding the attacks directed to them. Data collected at honeypot systems, such as traffic captures and operating system logs, is analyzed in order to gain information about attack techniques, general threat tendencies and exploits. It is assumed that traffic and activities directed at such systems are malicious, since they have no production value nor run any legitimate service accessed by regular users. Because of this characteristic (inherent to honeypot systems) the amount of data captured is significantly reduced in comparison to network IDSs which capture and analyze as much network traffic as possible.

Network honeypot systems are generally divided into two categories depending on their level of interaction with potential attackers: Low and High interaction honeypots. Being the simplest of network honeypots, the Low Interaction variant simply emulates specific operating systems TCP/IP protocol stacks and common network services, aiming at deceiving malicious users and automated attack tools [16]. Moreover, this type of honeypot has limited interaction with other hosts in the network, reducing the risks of compromising network security as a whole if an attacker successfully bypasses the isolation mechanisms implemented in the emulated services. High interaction honeypots are increasingly complex, running real operating systems and full implementations of common services with which a malicious user may fully interact inside sandboxes and isolation mechanisms in general. This type of honeypot captures more details concerning the malicious activities performed by an attacker, enabling analysis systems to exactly determine the vulnerabilities which were exploited, the attack techniques utilized and the malicious code executed.

Depending on the type of honeypot system deployed and the specific network set up, honeypots prove effective for a series of applications. Since those systems concentrate and attract malicious traffic, they can be used as decoys for slowing down or completely rendering ineffective automated attacks, as network intrusion detection systems and as a data source for identifying emergent threats and tendencies in the received malicious activity [3]. In the present work, we focus on identifying the principal malicious activities performed against a low interaction network honeypot system. Such a method for malicious activity identification may be applied in different scenarios, *e.g.* network intrusion detection.

A. Data Collection

Among other logs which may provide interesting information about an attacker's action, low interaction honeypots usually collect information regarding the network connections originated and directed to them, outputting *network flow* logs. These log files represent the basic elements which describe a connection, namely: timestamp, protocol, connection status (starting or ending), source IP, source port, destination IP and destination port. The following line illustrates the traffic log format of a popular low interaction honeypot system implementation [17]:

```
2008-06-04-00:00:03.7586 tcp(6) S 56.37.74.42
4406 203.49.33.129 1080 [Windows XP SP1]
```

It is possible to extract diverse information from this type of log while reducing the size of the analysis dataset in comparison to raw packet captures, which contain each packet sent or received by the monitored node. Furthermore, such information may be easily extracted from regular traffic capture files by aggregating packets which belong to the same connection, obtained the afore mentioned network flows

B. Data Analysis Methods

Various methods for honeypot data analysis with different objectives have been developed in order to accompany the increasing size of current honeypot systems, which are being deployed in progressively larger settings, comprising several different nodes and entire honeynets (networks of decoy hosts) distributed among different sites [6]. Most of the proposed analysis techniques are focused on processing traffic captures and malicious artefacts (*e.g.* exploit binaries and files) collected at the honeypot hosts [7]. Packet capture files, from which it is possible to extract network flow information (representing network traffic received and originated at the honeypot), provide both statistical data on threats and the necessary data for identifying intrusion attempts and attacks [18].

Classical methods for analysis of honeypot network traffic capture files rely on traffic pattern identification through file parsing with standard Unix tools and custom made scripts [16]. Basically, these methods consist of direct analysis of plain-text data or transferring the collected data to databases, where relevant statistical information is then extracted with custom queries. Such methods are commonly applied for obtaining aggregate data regarding traffic, but may prove inefficient for large volumes of data. Recently, distributed methods based on cloud infrastructure have been proposed for traffic data aggregation and analysis [19], efficiently delivering the aggregated traffic information needed as input for further analysis by other techniques.

In order to extract relevant information from sheer quantities of logs and collected data, data mining methods are applied to honeypot data analysis, specifically looking for abnormal activity and discovery of tendencies detection among regular traffic (*i.e.* noise). The clustering algorithm DBSCAN is applied in [9] to group packets captured in a honeypot system, distinguishing malicious traffic from normal traffic. Multiple series data mining is used to analyze aggregated network flow data in [8] in order to identify abnormal traffic features and anomalies in large scale environments. However, both methods require previous collection of large volumes of data and do not efficiently extract relevant statistics regarding the attacks targeting the honeypot with adequate accuracy.

A network flow analysis method based on the MapReduce cloud computing framework and capable of handling large volumes of data was proposed in [19] as a scalable alternative to traditional traffic analysis techniques. Large improvements in flow statistics computation time are achieved by this solution, since it distributes both processing loads and storage space. The proposed method is easily scalable, achieving the throughput needed to efficiently handle the sheer volumes of data collected in current networks (or honeypots), which present increasingly high traffic loads. This method may be applied to honeypot data analysis, providing general statistical data on the attack trends and types of threats.

C. Methods based on Principal Component Analysis

Several honeypot data analysis methods have been proposed in current literature, among them are principal component analysis (PCA) based techniques [12], [11]. Such methods aim at characterizing the type and number of malicious activities present in network traffic collected at honeypots through the statistical properties and distribution of the data. They are based on the fact that attack traffic patterns are more correlated than regular traffic, much like principal components in signal measurements. The first step of PCA is the estimation of the number of principal components. For this task, model order selection (MOS) schemes can be applied to identify significant malicious activities (represented by *principal components*) in traffic captures. Automatic MOS techniques are crucial to identify the number of the afore mentioned principal components in large network traffic datasets, this number being the *model order* of the dataset.

Basically, the model order of a dataset is estimated as the number of main uncorrelated components with energy significantly higher than the rest of components. In other words, the model order can be characterized by a power gap between the main components. In the context of network traffic, the principal components are represented by outstanding network activities, such as highly correlated network connections which have, for example, the same destination port. In this case, the principal components represent the outstanding groups of malicious activities or attacks directed at the honeypot system and the model order represents the number of such attacks. The efficacy and efficiency of PCA based methods depend on the adopted MOS schemes, since each scheme has different probabilities of detection for different kinds of data (depending on the kind of noise and statistical distribution of the data itself) [14].

A method for characterizing malicious activities in honeypot traffic data through principal component analysis techniques was introduced in [11]. This method consists in mainly two steps, dataset preprocessing and visual inspection of the eigenvalues profile of the covariance matrix of the preprocessed honeypot traffic samples in order to obtain the number of principal components (which indicate the outstanding groups of malicious activities), *i.e.* the model order. First, raw traffic captures are parsed in order to obtain network flows consisting of the basic IP flow data, namely the five-tuple containing the key fields: source address, destination address, source port, destination port, and protocol type. Packets received or sent during a given time slot (300 seconds in the presented experiments) which have the same key field values are grouped together in order to form these network flows. The preprocessing step includes further aggregation of network flow data, obtaining what the authors define as *activity flows*, which consist of combining the newly generated flows based upon the source IP address of the attacker with a maximum of sixty minutes interarrival time between basic connection flows. In the principal component analysis step, the preprocessed data is denoted by the p-dimensional vector $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ representing the network flow data for each time slot. First, the network flow data obtained after the preprocessing is transformed into zero mean and unitary variance with the following equation:

$$\mathrm{c_i} = \frac{\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{i}}{\sigma_\mathrm{i}^2} \qquad (1)$$

for $i = 1, \ldots, p$, where $\bar{\mathrm{x}}_\mathrm{i}$ is the sample mean and σ^2 is the sample variance for $\mathbf{x}_i$. Then the sample correlation matrix of **C** is obtained with the following expression:

$$R = \frac{1}{N}\left(C\, C^{T}\right) \qquad (2)$$

After obtaining the eigenvalues of the basic network flow dataset correlation matrix **R**, the number of principal components is obtained via visual inspection of the screen plot of eigenvalues in descending order. The estimation of the model order by visual inspection is performed by following subjective criteria such as considering only the eigenvalues greater than one and visually identifying a large gap between two consecutive eigenvalues.

The same authors proposed another method based on the same PCA technique and the equations described above for detecting new attacks in low-interaction honeypot traffic [12]. In the proposed model new observations are projected onto the residuals space of the least significant components and their distances from the k-dimensional hyperspace defined by the PCA model are measured using the square prediction error (SPE) statistic. A higher value of SPE indicates that the new observation represents a new direction that has not been captured by the PCA model of attacks seen in the historical honeypot traffic. As in the previous model, the model order of the preprocessed dataset is estimated through different criteria, including visual inspection of the eigenvalues screen plot.

Even though those methods are computationally efficient, they are extremely error prone, since the model order selection schemes (through which the principal components are determined) are based on subjective parameters which require visual inspection and human intervention. Apart from introducing uncertainties and errors, the requirement for human intervention also makes it impossible to implement such methods as an independent automatic analysis system. Thus these PCA based analysis methods are impractical for large networks, where the volume of collected data is continuously growing. Moreover, the uncertainty introduced by subjective human assistance is unacceptable, since it may generate a significant number of false positive detections.

III. Notation

Throughout the paper scalars are denoted by italic letters ($a, b, A, B, \alpha, \beta$), vectors by lower-case bold-face letters (**a**, **b**) and matrices by bold-face capitals (**A, B**). Lower-order parts are consistently named: the (i, k)-element of the matrix **A** is denoted as $a_{i,k}$. We denote by the diagonal vector of a matrix A. The element-wise productorial of vectors is denoted by $\Theta\ \Pi$. Concatenation between two elements a and b is denote by $a|b$.

We use the superscripts ^{T} and $^{-1}$ for transposition and matrix inversion, respectively.

IV. Applying Model Order Selection to Honeypot Data Analysis

Our method for MOS based honeypot data analysis bascially consists in applying state of the art MOS schemes to identify principal components of pre-processed aggregated network flow datasets. Each principal component represents a malicious activity and the number of such principal components (obtained through MOS) represents the number of malicious activities. In case this number is equal to zero, no malicious activity is present and in case it is greater than zero, there is malicious activity. Our objective in this paper is to automatically estimate the number of principal components (*i.e.* model order) of network flow datasets collected by honeypots. In this section, we introduce the method in details and the steps of data pre-processing necessary before model order selection is performed on the final dataset.

It has been observed that the traffic generated by outstanding malicious activities targeting honeypot systems has significantly higher volumes than regular traffic and it is also highly correlated, being distinguishable from random traffic and background noise [11]. Due to these characteristics it is viable to apply model order selection schemes to identify the number of principal components which represent malicious activities in network traffic captured by honeypot systems. Assuming that all traffic directed to network honeypot systems is malicious (*i.e.* generated by attempts of intrusion or malicious activities), outstanding highly correlated traffic patterns indicate individual malicious activities. Hence, each principal component detected in a dataset containing information on the network traffic represents an individual malicious activity. Analysing such principal components is an efficient way to estimate the number of different hostile activities targeting the honeypot system and characterizing them.

In order to estimate the number of principal components (*i.e.* malicious activities) the application of model order selection schemes arises naturally as an efficient method. After an appropriate preprocessing of the raw network traffic capture data, it is possible to estimate the model order of the dataset thus obtaining the number of malicious activities. The preprocessing is necessary in order to aggregate similar connections and network flows generated by a given malicious activity. It is observed that, after applying the preprocessing described in the previous section, groups of network flows pertaining to the same activity (*e.g.* groups which represent connections to and from the destination and source ports, respectively) have high correlated traffic profiles, yielding only one principal component. Thus, hostile activities which generate multiple connections are correctly detected as a single activity and not several different events.

Our method consists in applying RADOI with noise pre-whitening, a state-of-the-art *automatic* model order selection scheme based on the eigenvalues profile of the noise covariance matrix, to network flow datasets after preprocessing the data with the aggregation method described in the next sub-section. RADOI with noise pre-whitening was determined to be the most efficient method for performing model order selection of this type of datasets through experiments with real honeypot data in which several classical and state-of-the-art MOS schemes were evaluated (refer to Section VII for the results).

Since it is generally assumed that all traffic received by network honeypot systems is malicious, the model order obtained reflects the number of significant malicious activities present in the collected traffic, which are characterized by highly correlated and outstanding traffic. In our approach, the model order d obtained after applying the MOS scheme is considered as the number of malicious activities detected and the d highest dataset covariance matrix eigenvalues obtained represent the detected malicious activities. Further analysis of these eigenvalues enables other algorithms or analysts to determine exactly which ports were targeted by the detected attacks [12].

A. Data Pre-Processing Model

Before performing model order selection on the collected dataset it is necessary to transform it in order to obtain aggregate network flow data which represents the total connections per port and transport layer protocol. The proposed preprocessing method considers an input of network flow data extracted directly from log files generated by specific honeypot implementations (*e.g.* honeyd [17]) or from previously parsed and aggregated raw packet capture data (such parsing may be easily performed via existing methods [11]). It is possible to efficiently implement this preprocessing method based on a cloud infrastructure, providing scalability for large volumes of data [19]. Network flow data is defined as lines which represent the basic IP connection tuple for each connection originated or received by the honeypot system, containing the following fields: time stamp, transport layer protocol, connection status (starting or ending), source IP address, source port, destination IP address and destination port.

First, the original dataset is divided into n time slots according to the time stamp information of each network flow (n is chosen according to the selected time slot size). Subsequently the total connections directed to each m destination ports targeted during each time slot are summed up. We consider that the total connections to a certain destination port m during a certain time slot n is represented as follows:

$$x_m(n) = x_{0m}(n) + n_m(n) \qquad (3)$$

whereis $x_m(n) \in \mathbb{R}$ is the measured data in the port, $x_{0_m}(n) \in \mathbb{R}$ is the component related to the outstanding malicious activities $n_m(n) \in \mathbb{R}$ and is the noise component, mainly consisting of random connections and broadcasts sent to port m . Note that in case that no significant malicious activity is present, the traffic is mostly composed of port scans, broadcasts and other random non-malicious network activities, for instance. Therefore, the noise presentation fits well in (3).

In the matrix form, we can rewrite (3) as

$$\mathbf{X} = \mathbf{X_0} + \mathbf{N} \qquad (4)$$

where $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ is the total number of connections directed to M ports during N time slots. Particularly, if a certain port m has not been targeted by outstanding malicious activities, m-th line of $\boldsymbol{X}_0$ is fulled with zeros. On the other hand, if a certain i-th host is responsible for a malicious activity resulting in connections to P_i ports, these ports have a malicious traffic $\mathbf{S}_{\boldsymbol{i}} \in \mathbb{R}^{P_i \times N}$ highly correlated. Therefore, mathematically, $\mathbf{X_0}$ is given by

$$X_0 = \sum_{i=1}^{d} J_i \ S_i \qquad (5)$$

where $J_i \in \ ^{M \times P_i}$ is a zero padding matrix, such that the product $\boldsymbol{J}_{\mathrm{i}}$ by $\boldsymbol{S}_{\mathrm{i}}$ inserts zero lines in the ports without significant malicious activities. The total number of hosts with malicious traffic is represented by d. In an extreme case, when each line of $\boldsymbol{S}_{\mathrm{i}}$ has very high correlation, the rank of $\boldsymbol{S}_{\mathrm{i}}$ is 1. Therefore, the rank of $\boldsymbol{X}_0$ is d which is also known in the literature as model order or the total number of principal components, representing the total number of outstanding malicious activities detected in the honeypot dataset.

In order to represent the correlated traffic of the malicious traffic, we assume the following model

$$\boldsymbol{S}_i = \boldsymbol{Q}_i \boldsymbol{S}'_i, \qquad (6)$$

where $\boldsymbol{S}'_i \in \mathbb{R}^{P_i \times N}$ represents totally uncorrelated traffic and $\boldsymbol{Q}_i \in \mathbb{R}^{P_i \times P_i}$ is the correlation matrix between the ports. Note that if the correlation is not extremely high, the model order d represents the sum of the number of uncorrelated malicious activities of all hosts which interacted with the honeypot environment. Therefore, the model order d is at least equal to the total number of malicious hosts.

The correlation matrix of $\boldsymbol{X}$ defined in (4) is computed as

$$\boldsymbol{R}_{xx} = \mathrm{E}\{\boldsymbol{X}\boldsymbol{X}^T\} = \boldsymbol{R}_{0xx} + \boldsymbol{R}_{nn}, \qquad (7)$$

where $\mathrm{E}\{\cdot\}$ is the expected value operator and $R_{nn} = \sigma_{nn}^2 \boldsymbol{I} \ \in \mathbb{R}^{M \times M}$ is valid for zero mean white noise, where σ_{nn}^2 is the variance of the noise samples in (3). Note that we assume that the network flows generated by outstanding malicious activities are uncorrelated with the rest of traffic.

V. Model Order Selection Schemes

Several model order selection schemes exist, each of them with different characteristics which may affect their efficacy when applied to network traffic data. In this section, we present an overview of model order selection schemes and propose the necessary modifications in order to apply those schemes to malicious activity identification in honeypot data.

Usually, model order selection techniques are evaluated by comparing the *Probability of Correct Detection* or *PoD* (*i.e.* the probability of correctly detecting the number of principal components of a given dataset) of each technique for the type of data that is being analysed, since the different statistical distributions, noise and characteristics of specific datasets may alter the functioning and accuracy of different MOS schemes [14]. In other words, it is necessary to evaluate different MOS schemes with different characteristics in order to determine which MOS scheme is better suited for detecting malicious activities in honeypot network flow data. In this sense, we propose methods based on different schemes and evaluate them in the experiments presented in the next section.

In Subsection V-A, we show a brief review of the 1- Akaike's Information Criterion (AIC) [20], [13] and 1- Minimum Description Length (MDL) [20], [13], which are classical MOS methods, serving as a standard for comparing and evaluating novel MOS techniques and applications. Since RADOI [21] is one of the most robust model order selection schemes mainly for scenarios with colored noise, we propose the RADOI together with a noise prewhitening scheme in Subsection V-B.

Considering data preprocessed with the procedures described in the previous section, our method proceeds to performing model order selection of the dataset obtained. Similarly to [11], we also apply the zero mean in the measured sample. Therefore,

$$x_{\mathrm{ZM}_m} = x_m - \bar{x}_m \,, \tag{8}$$

where the vector $x_i \in \mathbb{R}^{1\times N}$ has all temporal samples of network flows directed to the port i, $\bar{x}_i$ is the mean value, and x_{ZM_i} contains the zero mean temporal samples. Such procedure is applied for each group of network flows directed to a single port in order to obtain $\boldsymbol{X}_{\mathrm{ZM}}$. By applying (8), the assumption that the samples have zero mean is fulfilled.

The techniques shown here are based on the eigenvalues profile of the noise covariance matrix $\boldsymbol{R}_{xx}$. Since the covariance matrix is not available, we can estimate it by using samples of the traffic. Therefore, we can approximate the covariance matrix to the following expression

$$\hat{R}_{xx} = \frac{1}{N}\boldsymbol{X}_{\mathrm{ZM}}\boldsymbol{X}_{\mathrm{ZM}}^{\mathrm{T}} \tag{9}$$

where $\hat{R}_{xx}$ is an estimate of $\boldsymbol{R}_{xx}$. In contrast to [11], we do not apply the unitary variance reviewed in (1), since the variance, which is the power of the components, is an useful information for the adopted model order selection schemes.

The eigenvalue decomposition of $\hat{\mathrm{R}}_{xx}$ is given by

$$\hat{R}_{xx} = \boldsymbol{E}\Lambda\boldsymbol{E}^{\mathrm{T}}, \tag{10}$$

where Λ is a diagonal matrix with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_\alpha$ with $\alpha = \min(M,N)$ and the matrix $\boldsymbol{E}$ has the eigenvectors. However, for our model order selection schemes, only the eigenvalues are necessary.

A. 1-D AIC and 1-D MDL

In AIC, MDL and Efficient Detection Criterion (EDC) [22], the information criterion is a function of the geometric mean, $g(k)$, and arithmetic mean, $a(k)$, of the k smallest eigenvalues of (10) respectively, and k is a candidate value for the model order d.

In [23], we have shown modifications of AIC and MDL for the case that $M > N$, which we have denoted by -D AIC and -D MDL. These techniques can be written in the following general form

$$\hat{d} = \operatorname*{argmin}_{k}\ J(k),$$

where,

$$J(k) = -N(\alpha - k)\log\left(\frac{g(k)}{a(k)}\right) + p(k,N,\alpha) \tag{11}$$

where $\hat{d}$ represents an estimate of the model order d. The penalty functions for 1-D AIC and 1-D MDL are given by $p(k,N,\alpha) = k(2\alpha - k)$ and $p(k,N,\alpha) = \frac{1}{2}k(2\alpha - k)\log(N)$ and respectively. According to [13] $\alpha = \min[M,N]$, while according to [23], we should use $\alpha = M$, and $0 \le k \le \min[M,N]$.

VI. Improving Performance in Parallel Environments

The previous pre-processing and MOS analysis methods are fit for small and medium sized environments with few honeypot systems collecting data and consequently generating moderate quantities of network flow data for subsequent analysis. However, in current enterprise network environments, it is often necessary to set up many honeypot systems distributed across different network portions in order to capture all relevant activities. In such an environment, the quantities of data generated may increase exponentially and overwhelm centralized data analysis solutions. In order to construct a scalable honeypot data analysis system, a promising approach consists in applying parallel processing techniques that distribute data analysis across several computer nodes that concurrently perform the necessary computation, thus increasing system velocity and capacity.

A trivial method to parallelize our techniques consists in aggregating the data collected by different honeypot systems at a central location and then distributing slices of data to individual computer nodes, that then run our analysis algorithms (pre-processing and MOS schemes) on their assigned data. The results are then aggregated at a central node. An analogous alternative is simply using parallel algorithms to compute the pre-processing and MOS scheme operations on the centralized data, distributing the computation (as opposed to data) to the computer nodes in a cluster. However, both of these direct approaches have a common shortcoming. In both cases, it is necessary to first transfer vast amounts of data to a central location in the network in order to start the analysis and then redistributed this data to the cluster nodes, which adds a huge communication overhead to the overall solution while degrading performance. Formally, we consider that the total quantity of data collected by k honeypots in the network is given by:

$$X_{\mathrm{T}} = [X_1|X_2| \dots |X_K] \tag{12}$$

where X_k is the data matrix of the k-th node. In this approach, the k-th node transmits its M by N data matrix X_k to the central node. Therefore, a data overhead of $(K-1)MN$ is foreseen. Note that usually $M > N$. The central node then computes the eigenvalues of the sample covariance matrix X_{T}.

Fortunately, it is possible to build on characteristics of our data model and the underlying MOS schemes to perform distributed analysis of the collected data without having to transfer it between different nodes. We propose instead an architecture where each node locally computes the eigenvalue decomposition of the sample covariance matrix corresponding to its locally collected data. The nodes then transmit only the diagonal vector of the resulting eigenvalue matrix to a central node, which aggregates the individual eigenvalue and estimate the model order of the full datatset employing global eigenvalue techniques [26], [23], [27]. A similar approach for locally processing network data in collection nodes is also presented in [15], where the authors adapt the MapReduce framework to enable nodes to perform local computation on their local data and then aggregate the result, instead of transfering data to a central local that then redistributes it to the worker nodes. Apart from improving network performance, this technique also results in a larger gap between eigenvalues, increasing overall probability of detection, making it more efficient in detecting attacks and less prone to false negatives.

This method is formalized as follows. We consider a scenario where K nodes are continuously collecting traffic and generating $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ as described in Section IV-A. After a certain number N_G of collection time slots, the total data collected by the nodes consists in K data matrices $X_k \in \mathbb{R}^{M \times N_G}$ for $k = 1, \dots, K$. In the end of the collection period of N_G time slots, each k-th node then computes the sample covariance matrix $\hat{R}_{xx,k}$ for its locally collected data X_k. Notice that, at this point, the trivial next step would be for each k-th node to simply send its sample covariance matrix $\hat{R}_{xx,k}$ to a central node that would perform the remaining steps in estimating the model order.

$$R_{xx,\mathrm{T}} = \frac{1}{K} \sum_{k=1}^{K} R_{xx,k}, \tag{13}$$

where $R_{xx,k}$ is the sample covariance matrix of X_k. In this case, since the sample covariance matrix is transmitted the data overhead is $(K-1)M^2$. Note that mathematically we obtain the same eigenvalues via (12) or via (13). Therefore, (13) should be preferentially used due to the reduced overhead. On the other hand, we avoid the excessive data transfers by requiring that each k-th node computes the eigenvalue decomposition of $\hat{R}_{xx,k}$, obtaining the eigenvalue matrix Λ_k. Finally, each node transfers only the diagonal eigenvalue vector $\mathrm{diag}(\Lambda_k)$, instead of the complete sample covariance matrix $\hat{R}_{xx,k}$. The central node then aggregates each individual eigenvalue vector $\mathrm{diag}(\Lambda_k)$ into a global eigenvalues vector $\Lambda^{(G)}$, which is used to estimate model order through RADOI. $\Lambda^{(G)}$ is obtained as follows:

$$\Lambda^{(G)} = \odot \prod_{k=1}^{K} \mathrm{diag}(\Lambda_k) \qquad (14)$$

Notice that in this approach, each node is only required to transfer vectors of M real numbers representing the eigenvalues. If the full data matrix or the local sample covariance matrix were transmitted, it would be necessary to transfer $M \cdot N_G$ or $M \cdot M$ real number values, respectively. This represents a factor N_G or M a factor decrease in the total size of transmitted data, in comparison to transmitting the full data matrix or the local sample covariance matrix, respectively. Notice that even if N increases, meaning that the resolution is increased with more samples being taken for each time period, the size of the transmitted data is the same. In practice, it means that the local resolution of each sensor does not affect the total quantity of data that needs to be transmitted for the central node for analysis.

The corresponding R-dimensional versions are obtained by replacing the eigenvalues of $\hat{R}_{xx}$ by the global eigenvalues λ_i defined in equation (14). Additionally, for computing the number of free parameters for the AIC and MDL methods and their R-D extensions, we assume to set the parameter *number of sensors* to the number of *global eigenvalues* and the parameter *number of temporal samples* [25].

VII. Simulations

In order to validate the R-D EFT approach for estimating the model order of the analysis dataset parallel as described in Section VI, simulation experiments were performed. These experiments show that the threshold between eigenvalues increases as expected, while the total data transfer dramatically decreases. In these experiments we compare the global eigenvalues profile obtained by the parallel method described in the previous section with the eigenvalue profiles obtained by three trivial approaches for distributed honeypot data model order estimation. We consider a scenario with K = 10 nodes, model order d and traffic to M = 10 ports collected over N_G = 40 10 minute time slots. The signal and noise samples are i.i.d. zero mean Gaussian and the SNR is defined as

$$SNR = 20\log_{10}\left(\frac{\sigma_s}{\sigma_n}\right), \qquad (15)$$

where σ_s is the signal variance and σ_n is the noise variance.

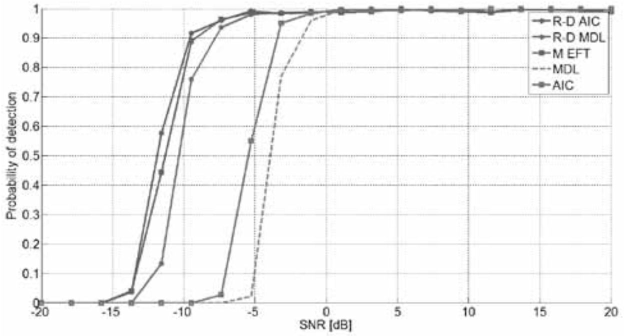

Figure 1. Probability of correct Detection (PoD) versus SNR for d = 3

Figure 1 depicts the Probability of correct Detection (PoD) versus the SNR for *d* = 3. By applying the Global Eigenvalues, a considerable improvement is achieved when we compare the performance of AIC and MDL to the performance of the R-D AIC and R-D MDL.

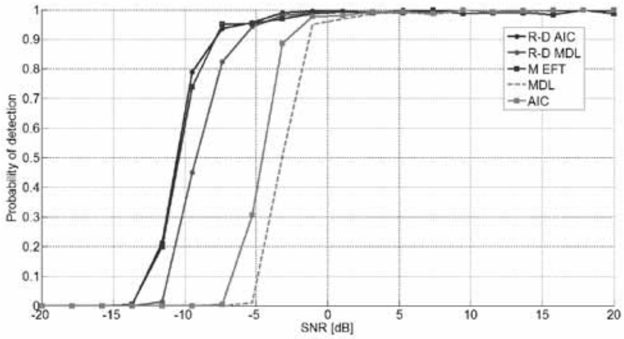

Figure 2. Probability of correct Detection (PoD) versus SNR for d = 4

In Figure 2, we consider *d* = 4, which means that there is a greater malicious traffic.

In Figs. 1 and 2, the M-EFT achieveso s a very high PoD close to the R-D AIC and R-D MDL. Therefore, M-EFT can be also optionaly used instead of R-D AIC and R-D MDL.

Note that in case that the R-D AIC and R-D MDL may not applicable for real applications of malicious traffic [34]. Therefore, the R-D EFT should be applied [26] in such scenarios.

VIII. ConclusionS

In this paper, we have shown that by applying the global eigenvalues instead of the eigenvalues, an improvement in the probability of correct detection (PoD) of model order selection schemes is obtained.

We have shown here that the R-D AIC and R-D MDL, which are parallelized versions of the AIC and MDL, works even for low SNR regimes.

We also suggest to use the R-D EFT for all scenarios. In order to use the R-D EFT, low levels of Probability of False Alarm should be considered.

Acknowledgements

The research presented in this paper was conducted under a grant from DELL Computers of Brazil under the agreement 001/2010. The authors wish to thank the Brazilian Ministry of Plan, Budget and Management for its support to this research.

References

[1] B. M. David, J. P. C. L. da Costa, A. C. A. Nascimento, D. Amaral, M.D. Holtz, and R. T. de Sousa Jr., “Blind automatic malicious activity detection in honeypot data,” in The International Conference on Forensic Computer Science (ICoFCS), 2011.

[2] L. Spitzner, “Honeypots: Catching the insider threat,” in Proceedings of the 19th Annual Computer Security Applications Conference, ser. ACSAC ’03. Washington, DC, USA: IEEE Computer Society, 2003,pp. 170–.

[3] Z. Li-juan, “Honeypot-based defense system research and design,” Com-puter Science and Information Technology, International Conference on, vol. 0, pp. 466–468, 2009.

[4] I. Mokube and M. Adams, “Honeypots: concepts, approaches, and challenges,” in Proceedings of the 45th annual southeast regionalconference, ser. ACM-SE 45. New York, NY, USA: ACM, 2007, pp. 321–326. [Online]. Available: http://doi.acm.org/10.1145/1233341.1233399

[5] F. Zhang, S. Zhou, Z. Qin, and J. Liu, “Honeypot: a supplemented active defense system for network security,” in Parallel and Distributed Com-puting, Applications and Technologies, 2003. PDCAT’2003. Proceedings of the Fourth International Conference on, 2003, pp. 231 – 235.

[6] E. Alata, M. Dacier, Y. Deswarte, M. Kaniche, K. Kortchinsky, V. Nicomette, V. H. Pham, and F. Pouget, “Collection and analysis of attack data based on honeypots deployed on the internet,” in Quality of Protection, ser. Advances in Information Security, D. Gollmann, F. Massacci, and A. Yautsiukhin, Eds.Springer US, 2006, vol. 23, pp. 79–91.

[7] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, “Honeypot forensics,” in Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, June 2004, pp. 22 – 29.

[8] W. He, G. Hu, X. Yao, G. Kan, H. Wang, and H. Xiang, “Applying multiple time series data mining to large-scale network traffic analysis, ”in Cybernetics and Intelligent Systems, 2008 IEEE Conference on, September 2008, pp. 394 –399.

[9] A. Ghourabi, T. Abbes, and A. Bouhoula, “Data analyzer based on data mining for honeypot router,” Computer Systems and Applications, ACS/IEEE International Conference on, vol. 0, pp. 1–6, 2010.

[10] Z.-H. Tian, B.-X. Fang, and X.-C. Yun, “An architecture for intrusion detection using honey pot,” in Machine Learning and Cybernetics, 2003 International Conference on, vol. 4, 2003, pp. 2096 – 2100 Vol.4.

[11] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, “Characterization of attackers’ activities in honeypottrafficusing principal component analysis,” in Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing. Washington, DC, USA: IEEE Computer Society, 2008, pp. 147–154.

[12] ——, “A technique for detecting new attacks in low-interaction honeypot traffic,” in Proceedings of the 2009 Fourth International Conference on Internet Monitoring and Protection. Washington, DC, USA: IEEE Computer Society, 2009, pp. 7–13.

[13] J. P. C. L. da Costa, A. Thakre, F. Roemer, and M. Haardt, “Comparison of model order selection techniques for high-resolution parameter estimation algorithms,” in Proc. 54th International Scientific Colloquium (IWK’09), Ilmenau, Germany, Oct. 2009.

[14] J. P. C. L. da Costa, Parameter Estimation Techniques for Multi-dimensional Array Signal Processing, 1st ed. Shaker, 2010.

[15] D. Logothetis, C. Trezzo, K. C. Webb, and K. Yocum, “In-situ mapreduce for log processing,” in Proceedings of the 2011 USENIX conference on USENIX annual technical conference, ser. USENIXATC’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 9–9. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002181.2002190

[16] V. Maheswari and P. E. Sankaranarayanan, “Honeypots: Deployment and data forensic analysis,” in Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) - Volume 04, ser. ICCIMA ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 129–131.

[17] N. Provos, “A virtual honeypot framework,” in Proceedings of the 13th conference on USENIX Security Symposium - Volume 13, ser. SSYM’04. Berkeley, CA, USA: USENIX Association, 2004, pp. 1–1. [Online]. Available: http://portal.acm.org/citation.cfm?id=1251375.1251376

[18] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, “Honeypot forensics part i: Analyzing the network,” IEEE Security and Privacy, vol. 2, pp.72–78, July 2004.

[19] Y. Lee, W. Kang, and H. Son, “An internet traffic analysis method with mapreduce,” in Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, April 2010, pp. 357 –361.

[20] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-33, pp. 387–392, 1985.

[21] E. Radoi and A. Quinquis, “A new method for estimating the number of harmonic components in noise with application in high resolution radar,” EURASIP Journal on Applied Signal Processing, pp. 1177–1188, 2004.

[22] L. C. Zhao, P. R. Krishnaiah, and Z. D. Bai, “On detection of the number of signals in presence of white noise,” J. Multivar. Anal., vol. 20, pp. 1–25, October 1986. [Online]. Available:http://portal.acm.org/citation.cfm?id=9692.9693

[23] J. P. C. L. da Costa, M. Haardt, F. Roemer, and G. Del Galdo, “Enhanced model order estimation using higher-order arrays,” in Proc. 40th Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, USA, Nov. 2007.

[24] R. R. Nadakuditi and A. Edelman, “Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples, ”IEEE Transactions of Signal Processing, vol. 56, pp. 2625–2638, Jul. 2008.

[25] H.-T. Wu, J.-F. Yang, and F.-K. Chen, “Source number estimators using transformed Gerschgorin radii,” IEEE Transactions on Signal Processing, vol. 43, no. 6, pp. 1325–1333, 1995.

[26] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa Jr., “Multi-dimensional model order selection,” EURASIP Journal on Advances in Signal Processing, vol. 26, 2011.

[27] J. P. C. L. da Costa, M. Haardt, and F. Roemer, “Robust methods based on the HOSVD for estimating the model order in parafac models, in Proc. 5-th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2008), 2008, pp. 510–514.

[28] Y. Liu, C.-S. Bouganis, P. Cheung, P. Leong, and S. Motley, “Hardware eficient architectures for eigenvalue computation,” in Design, Automation and Test in Europe, 2006. DATE ’06. Proceedings, vol. 1, 2006,pp. 1 –6.

[29] Y. Hu, “Parallel eigenvalue decomposition for toeplitz and related matrices,” in Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, May 1989, pp. 1107 –1110 vol.2.

[30] J. Grouffaud, P. Larzabal, and H. Clergeot, “Some properties of ordered eigenvalues of a wishart matrix: application in detection test and model order selection,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’96), vol. 5, May 1996, pp. 2463 – 2466.

[31] A. Quinlan, J. Barbot, P. Larzabal, and M. Haardt, “Model order selec-tion for short data: An exponential fitting test (EFT),” EURASIP Journal on Applied Signal Processing, 2007, special Issue on Advances in Subspace-based Techniques for Signal Processing and Communications.

[32] M. O. Ulfarsson and V. Solo, “Rank selection in noisy PCA with SURE and random matrix theory,” in Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), Las Vegas, USA, Apr. 2008.

[33] S. Kritchman and B. Nadler, “Determining the number of components in a factor model from limited noisy data,” Chemometrics and Intelligent Laboratory Systems, vol. 94, pp. 19–32, Nov. 2008

[34] J. P. C. L. da Costa, E. P. de Freitas, B. M. David, D. Amaral, and R. T. de Sousa Jr., “Improved Blind Automatic Malicious Activity Detection in Honeypot Data,” The International Conference on Forensic Computer Science (ICoFCS) 2012, Brasília, Brazil, best paper award.

[35] B. M. David, J. P. C. L. da Costa, A. C. A. Nascimento, M. D. Holtz, D. Amaral, and R. T. de Sousa Jr., “A Parallel Approach to PCA Based Malicious Activity Detection in Distributed Honeypot Data,” International Journal of Forensic Computer Science (IJoFCS), 2011.

[36] M. Haardt, R. S. Thomae, and A. Richter, “Multidimensional high-resolution parameter estimation with applications to channel sounding,” in High-Resolution and Robust Signal Processing, Y. Hua, A. Gershman, and Q. Chen, Eds. 2004, pp. 255–338, Marcel Dekker, New York, NY, Chapter 5.

[37] D. Amaral and L. Peotta, and A. Edelman, “Honeypot de Baixa Interação Como Ferramenta para Detecção de Tráfego com Propagação de Botnets”, in The International Conference on Forensic Computer Science (ICoFCS), 2007. Available: http://dx.doi.org/10.5769/C2007011

[38] M. C. Sacchetin, A. R. A. Gregio, L. O. Duarte and A. Montes , “Botnet Detection and Analysis Using Honeynet”, in The International Conference on Forensic Computer Science (ICoFCS), 2007. Available: http://dx.doi.org/10.5769/C2007003

[39] R. Rebiha, and A. V. Moura, “Automated Malware Invariant Generation”, in The International Conference on Forensic Computer Science (ICoFCS), 2009. Available: http://dx.doi.org/10.5769/C2009001

[40] D. Amaral, G. Araújo, and A. Romariz, “Detecting Attacks to Computer Networks Using a Multi-layer Perceptron Artificial Neural Network”, in The International Journal of Forensic Computer Science (IJoFCS), 2008, V. 3, N. 1. Available: http://dx.doi.org/10.5769/J200801007