DOI: http://dx.doi.org/10.5769/C2011016

Blind Automatic Malicious Activity Detection in Honeypot Data

Bernardo Machado David, João Paulo C. L. da Costa, Anderson C. A. Nascimento, Marcelo Dias Holtz, Dino Amaral, Rafael Timóteo de Sousa Júnior
Department of Electrical Engineering
University of Brasilia
bernardo.david@redes.unb.br, jpdacosta@unb.br, andclay@ene.unb.br,
holtz@redes.unb.br, dinoamaral@redes.unb.br, desousa@unb.br

***Abstract* — Model order selection (MOS) schemes are frequently applied in several signal processing applications. In this paper, we propose a new application for such state-of-the-art model order selection schemes, which is an automatic method for blind identification of malicious activities in honeypot systems. Our proposed blind automatic techniques are efficient and need no previous training nor knowledge of attack signatures for detecting malicious activities. In order to achieve such results, we model network traffic data as signals and noise, which allows us to apply modified signal processing methods. We adapt model order selection schemes to process network data, showing that RADOI achieves the best performance and reliability in detecting attacks. The efficiency and accuracy of our theoretical results are tested on real data collected at a honeypot system located at the network border of a large banking institution in Latin America.**

Keywords-Intrusion — Detection; Honeypot; Model Order Selection; Principal Component Analysis

1. Introduction

The Problem. A honeypot system collects malicious traffic and general information on malicious activities directed towards the network where it is located [23]. It serves both as data source for intrusion detection systems as well as a decoy for slowing down automated attacks [13], [16]. Efficient algorithms for identifying malicious activities in honeypot data are particularly useful in network management statistics generation, intelligent intrusion prevention systems and network administration in general as administrators can take actions to protect the network based on the results obtained [28]. Even though honeypots provide a reliable and representative source for identifying attacks and threats [1], they potentially produce huge volumes of complex traffic and activity logs making their efficient and automated analysis quite a challenge.

Previous Works. Several methods have been proposed for identifying and characterizing malicious activities in honeypot traffic data based on a variety of approaches and techniques [21], [9], [7]. Classical methods typically employ data mining [9], [7] and regular file parsing [21] for detecting patterns which indicate the presence of specific attacks in the analysed traffic and computing general statistical data on the collected traffic. These methods depend on previous knowledge of the attacks which are going to be identified and on the collection of significant quantities of logs in order to work properly. Recently, machine learning techniques have also been applied to honeypot data analysis and attack detection [24] yielding interesting results as those techniques are able to identify malicious activities without relying on previously provided malicious traffic patterns and attack signatures. However, it is necessary to run several analysis cycles during a *learning* period in order to train the system to recognize certain attacks before such methods are able to work effectively, rendering them computationally expensive. Furthermore, if the legitimate traffic patterns are altered by any natural causes, 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 not detecting attacks which were not included in the learning process or whose traffic resembles honest patterns.

Principal component analysis (PCA) based methods [2], [3] came on to the scene 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. Such methods are based on the clever observation that attack traffic patterns are more correlated than regular network traffic. Since they solely rely on statistical analysis of the collected data, these methods need not be provided with previous information on the attacks to be detected neither need to be trained to recognize attacks and separate them from legitimate traffic. This characteristic makes PCA based honeypot data analysis methods suitable for automatic attack detection and traffic analysis. However, current PCA based methods [2], [3] still require human intervention, rendering them impractical for automatic analysis and prone to errors such as false positives.

Our Contributions. We propose a method for automatically identifying attacks in low interaction honeypot network traffic data based on state-of-the-art model order

selection schemes [6], [4]. In order to obtain this result we present the following contributions:

- We propose to model network traffic as signals and noise data, interpreting highly correlated components as significant network activities (in this case, malicious activities).
- It is possible to identify malicious activities in honeypot network flow datasets *without* any previous information or attack signatures by applying model order selection schemes.
- Blind malicious detection schemes in the literature [3], [2] require human inspection to detect malicious activities. In this paper, we obtain a blind automatic detection method without the need of any human intervention by using model order selection schemes.
- We adapt RADOI to successfully identify the main attacks contained in the simulation data set, efficiently distinguishing outstanding malicious activities from noise such as backscatter and broadcast packets.

More generally, our method is an intrusion detection system which does not require previous knowledge of attack signatures and might find interesting applications in contexts other than honeypot systems. Finally, being based on eigenvalues decomposition techniques, our method is efficiently implementable in hardware [14] and can also be parallelized [10].

Roadmap. The remainder of this paper is organized as follows. In Section 2, we define the notation used in this paper. In Section 3, we formally introduce the concept of honeypots, discuss classical analysis methods and present an analysis of related work on PCA based methods for honeypot data analysis. In Section 4, we describe the dataset preprocessing method through which we transform the data before Model Order Selection (MOS). In Section 5, we introduce classical MOS and also state-of-the-art schemes and propose our analysis method based on RADOI. In Section 6, we evaluate several MOS schemes in experiments with real data, presenting experimental results which attest the validity of our approach. In Section 7, we finally conclude with a summary of our results and direction for future research.

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

3. Related Works

In this section, we introduce the concept of honeypot systems and discuss the several methods used for obtain 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 [23], 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 at 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 [15]. 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 [13]. 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 at 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 [18]:

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 [1]. 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 [21]. 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 [22].

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 [18]. 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 [12], 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 [7] 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 [9] 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 [12] 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 [3], [2]. 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 MOS schemes adopted, 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) [4].

A method for characterizing malicious activities in honeypot traffic data through principal component analysis techniques was introduced in [2]. 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 inter-arrival time between basic connection flows. In the principal component analysis step, the preprocessed data is denoted by the p-dimensional vector $\mathbf{x} = (x_1,...,x_p)^{\mathrm{T}}$ 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:

$$\mathbf{c_i} = \frac{\mathbf{x}_i - \bar{\mathbf{x}}_i}{\sigma_i^2}, . \quad (1)$$

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

$$\mathbf{R} = \frac{1}{N}(\mathbf{C} \cdot \mathbf{C}^{\mathrm{T}}). \quad (2)$$

After obtaining the eigenvalues of the basic network flow dataset correlation matrix $\mathbf{R}$, 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 [3]. 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 prone to error, 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.

4. 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 our 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 is also highly correlated, being distinguishable from random traffic and background noise [2]. 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 where several classical and state-of-the-art MOS schemes were evaluated (refer to Section VI 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 [3].

D. 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 [18]) or from previously parsed and aggregated raw packet capture data (such parsing may be easily performed via existing methods [2]). It is possible to efficiently implement this preprocessing method based on a cloud infrastructure, providing nice scalability for large volumes of data [12]. 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:

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

where $x_m(n) \in \mathbb{R}$ is the measured data in the port, $x_{0m}(n) \in \mathbb{R}$ is the component related to the outstanding malicious activities and $n_m(n) \in \mathbb{R}$ 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

$$\mathbf{X} = \mathbf{X_0} + \mathbf{N} \tag{4}$$

Where $\boldsymbol{X} \in \mathbb{R}^{MXN}$ 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, the 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 $\boldsymbol{S}\,i \in \mathbb{R}^{PiXN}$ highly correlated. Therefore, mathematically, $\mathbf{X_0}$ is given by

$$X_0 = \sum_{i=1}^{d} J_i S_i \tag{5}$$

where $\boldsymbol{J}_i \in {}^{MxPi}$ is a zero padding matrix, such that the product $\boldsymbol{J}_i$ by $\boldsymbol{S}_i$ 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 $\boldsymbol{S}_i$ has very high correlation, the rank of $\boldsymbol{S}_i$ 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

$$\boldsymbol{S}_i = \boldsymbol{Q}_i \ \boldsymbol{S'}_i, \tag{6}$$

where $\boldsymbol{S'}_i \in \mathbb{R}^{PiXN}$ represents totally uncorrelated traffic and $\boldsymbol{Q}_i \in \mathbb{R}^{PiXPi}$ 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

$$\begin{aligned} \boldsymbol{R}_{xx} &= \mathbf{E}\{\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\} \\ &= \boldsymbol{R}_{0xx} + \boldsymbol{R}_{nn'} \end{aligned} \tag{7}$$

where E{·} is the expected value operator and $\boldsymbol{R}_{nn} = \sigma_{nn}^2 \boldsymbol{I} \in \mathbb{R}^{MxM}$ 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.

5. 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 [4]. 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-*D* Akaike's Information Criterion (AIC) [26], [6] and 1-*D* Minimum Description Length (MDL) [26], [6], which are classical MOS methods, serving as a standard for comparing and evaluating novel MOS techniques and applications. Since RADOI [20] 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 [2], we also apply the zero mean in the measured sample. Therefore,

$$x_{z_{M_m}} = x_m - \overline{x}_m, \tag{8}$$

where the vector $x_i \in \mathbb{R}^{1xN}$ has all temporal samples of network flows directed to the port i, $\overline{x}_i$ is the mean value, and $x_{z_{M_i}}$ 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 $\boldsymbol{X}_n z_M$. 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

$$\hat{R}_{xx} = \frac{1}{n} \mathbf{z}_M \mathbf{z}_M^T, \tag{9}$$

where $\hat{R}_{xx}$ is an estimate of $\boldsymbol{R}_{xx}$. In contrast to [4], 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 $\hat{R}_{xx}$ is given by

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

where Λ is a diagonal matrix with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_\alpha$ 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.

E. 1-D AIC and 1-D MDL

In AIC, MDL and Efficient Detection Criterion (EDC) [29], 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 [5], we have shown modifications of AIC and MDL for the case that $M > N$, which we have denoted by 1-D AIC and 1-D MDL. These techniques can be written in the following general form

$$\hat{d} = \arg\min J(k) \qquad where$$
$$J(k) = -N(\alpha - k)\log\left(\frac{g(k)}{a(k)}\right) + p(k, N, \alpha), \tag{11}$$

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 $p(k, N, \alpha) = \frac{1}{2} k(2\alpha - k) \log(N)$ respectively. According to [13] $\alpha = min[M, N]$, while according to [21], we should use $\alpha = M$, and $0 \leq k \leq min[M, N]$.

F. RADOI with Noise Prewhitening

The RADOI model order selection scheme is an empirical approach [20]. Here we propose to incorporate the noise prewhitening to the RADOI scheme in order to improve its performance. In order to apply the noise prewhitening, first samples containing only noise traffic are collected. Such noise samples can be obtained from M_n ports where no significant malicious activities are observed. In practice, we can select the M_n ports with lowest traffic rates (*i.e.* ports which received an insignificant number connections during the time span observed, for example, less than 1 connection per minute). By using the noise samples, we compute an estimate of the noise correlation matrix

$$\hat{R}_{nn} = \frac{1}{n} \mathbf{z}_M \mathbf{z}_M^T, \tag{12}$$

where $\boldsymbol{N}_{z_M}$ contains the zero mean noise samples computed similarly as in (8). With $\hat{R}_{nn}$, the noise prewhitening matrix can be computed by applying the Cholesky decomposition

$$\hat{R}_{nn} = LL^T, \tag{13}$$

where $\boldsymbol{L} \in \mathbb{R}^{M_n \times M_n}$ is full rank.

The noise prewhitening of $\boldsymbol{X}$ is given by

$$\boldsymbol{X}_{\text{pwt}} = L^{-1}\boldsymbol{X} \tag{14}$$

We compute the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_\alpha$ of the covariance matrix of $\boldsymbol{X}_{\text{pwt}}$ and we apply them on the RADOI cost function, which is given by

$$\hat{d} = \underset{k}{\text{argmin}} \text{R}ADOI(k) \quad \text{where} \tag{15}$$

$$\text{R}ADOI(k) = \lambda_{k+1} \cdot \left(\sum_{i=2}^{M} \lambda_i \right)^{-1} - \xi_k \cdot \left(\sum_{i=1}^{M-1} \xi_i \right)^{-1} \tag{16}$$

where $\xi_k = 1 - \dfrac{\alpha \cdot (\lambda_k - \mu_k)}{\mu_k}$, $\mu_k = \dfrac{1}{M-k} \cdot \sum_{i=k+1}^{M} \lambda_i$,

and α is given by

$$\alpha = \left[\arg\max_k \frac{(\lambda_k - \mu_k)}{\mu_k} \right]^{-1} \tag{17}$$

In [20], it is shown that RADOI outperforms the Gerschgoerin disk estimator (GDE) criterion [27] in the presence of colored noise, while its performance in the presence of white noise is similar to the GDE criterion.

6. Simulations

In this section, we describe a series of experiments that were performed in order to validate our proposed scheme for detection of malicious activities in honeypot network traffic. Throughout this section we consider a dataset collected at a large real world honeypot installation. First, in Subsection VI-B, we manually determine the number of attacks in the experimental dataset and then analyse the data preprocessing model. In Subsection VI-C, we compare the performance of several model order selection schemes presented in Section V, determining that RADOI with zero mean and noise prewhitening is the most efficient and accurate method for analysing such data.

G. Experimental Environment

In the experiments presented in this section we consider a dataset containing network flow information collected by a large real world *honeyd* virtual network honeypot installation. The reader is referred to [4], [6] in order to check the performance of the MOS schemes for simulated data. Extensive simulation campaigns are performed in [4], [6].

Honeyd is a popular framework which implements virtual low interaction honeypots simulating virtual computer systems at the network level [18]. The simulated information system resources appear to run on unallocated network addresses, thus avoiding being accessed by legitimate users. In order to deceive network fingerprinting tools and honeypot evasion methods, honeyd simulates the networking protocol stack of different operating systems. It is also capable of providing arbitrary network services and routing topologies for an arbitrary number of virtual systems.

Among other monitoring and management related data, honeyd automatically generates network activity logs in the form of network flow data as described in Section III-A. A dataset comprised of such network flow logs is analysed in the following experiments. For experimental purposes, the data preprocessing model and the different model order selection schemes were numerically implemented, providing accurate results. However, the issues of efficiency [14], [10] and scalability [12] for large volumes of data are not addressed, which is left as subject for future works.

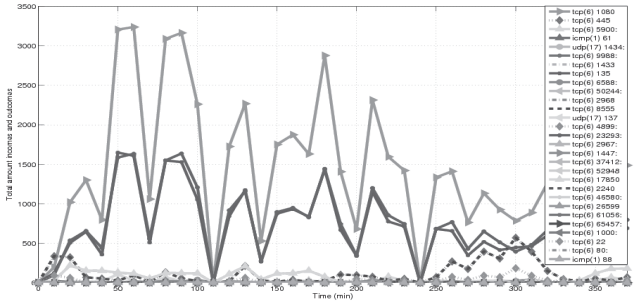

Figure 1: Traffic over M different ports vs N time slots. Each time slot spans 10 minutes. The total amount of M ports and the total amount of N time slots are 29 and 37, respectively.

H. Data model fitting based on collected data

It is necessary to manually analyse the experimental dataset in order to obtain an accurate estimate of the number of attacks that it contains. Notice that this manual analysis is not part of the proposed method, which is completely automatic. The results obtained in this analysis are merely utilized as a reference value to be compared with the results obtained by the different MOS schemes in the process of validating our automatic results.

Besides the number of connections per port, this manual analysis takes into consideration common knowledge on which services are mostly targeted in such attacks. First, we are interested in obtaining summarized information on the total number of connections per port. Thus, we evaluate our proposed data preprocessing model, obtaining a preprocessed summarized dataset from the original network flow data.

A time slot of 10 minutes is considered, with data collection starting at at 2007-08-02-13:51:59 and spanning approximately 370 minutes (or 37 slots). During the data collection period considered, network activities targeting 29 different TCP and UDP ports were observed, thus yielding a preprocessed data matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ with $M = 29$ different ports and $N = 37$ time slots, representing the total number of connections directed to or originated from the M ports during each of the N time slots. In Fig. 1, the preprocessed data matrix $\boldsymbol{X}$ is depicted, providing graphical information on the traffic profiles. Although it is not possible to distinguish all curves, notice that some ports have

outstandingly higher traffic while the traffic profile pertaining to the rest of the ports are close to zero, behaving akin to noise. Thus, we show that some traffic profile curves are significantly higher than others due to the attacks directed at them. Once again note that this is not part of the blind automatic method proposed, serving only as a reference for our experiments.

According to Fig. 1, the traffic profiles of some ports clearly indicate malicious activities and attacks. By manually analysing the collected network flow data and visually inspecting the traffic plot, it is possible to determine that a threshold of more than an average of 100 connections per 10 minutes time slots to a certain port during the observed time span indicates malicious activities. Traffic profiles of less than an average of 100 connections per 10 minutes to a given port (or 0.17 connections per second) are considerably less than the number of connections to the highly attacked ports, being considered noise and not indicating significant malicious activities. Therefore, we conclude that outstanding malicious activities are observed on ports m = 1, 2, 7, 8, 12, 15, 20, that in Fig. 1 respectively correspond to the following ports: TCP 1080, TCP 445, TCP 1433, TCP 135, TCP 8555, TCP 23293, and TCP 17850.

Further analysis of the traffic profile of each port indicates that the pair of ports TCP 135 and TCP 23293 are destination and source ports for the same connections respectively. Therefore, their traffic profiles are almost identical, i.e., highly correlated. The ports TCP 445 and TCP 8555 are also destination and source ports for a certain group of connections, as well as the ports TCP 1433 and TCP 17850. The destination ports of the pairs described before along with TCP port 1080 are typically opened by commonly probed and attacked services, which explains the intense activity observed and confirms the hypothesis that the traffic directed to those ports actually represents malicious activities.

Although a high level of network activity is observed in 7 different ports, 3 pairs have very highly correlated patterns and for this reason can be considered as only 3 main components (representing 3 different significant malicious activities which, in this case, are easily identifiable as attacks to services commonly present in popular operating systems and network equipment). Hence, given the traffic profile in Fig. 1 we conclude that the model order for the dataset being analysed in the following experiments is equal to 4, since it is the number of malicious activities or attacks identified after manually analysing network data.

In Fig. 2, the traffic profile of all ports which received or originated less than an average of 100 connections per time slot is depicted. Notice that, once again, it is not possible to distinguish the traffic profiles but this figure clearly shows that traffic not generated by attacks behaves like random noise. Thus, the traffic in those ports is considered noise (generated by broadcast messages, faulty applications and other random causes) and we consider, therefore, that it does not characterize malicious activities. This analysis is not part of the method proposed, serving only as reference for analysing our experiments.

Based on the data model presented in Section 4, the data shown in Fig. 2 is that of the noise components represented by matrix $\boldsymbol{N} \in {}^{M_n \times N}$. Note that since $P_1 = 7$ and m= 1, 2, 7, 8, 12, 15, 20 , the zero padding matrix $\boldsymbol{J} \in {}^{M \times P_1}$ described in (5) which indicates the ports with outstanding malicious activities has $j_{(i,k)} = 1$ only for the following values of $(i, k) = \{(1,1), (2,2) (7,3), (8,4), (12,5), (15,6) (20,7)\}$, otherwise, $j_{(i,k)} = 0$.

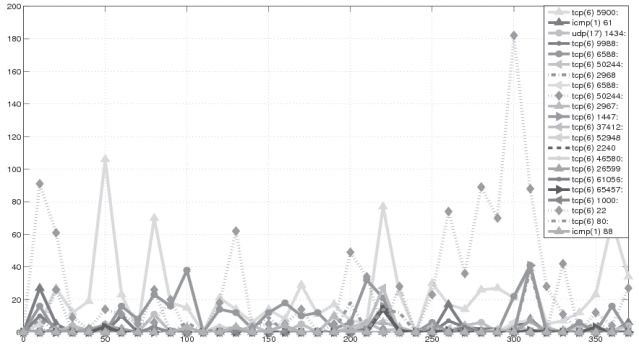

Figure 2: Noise traffic over M_n ports vs N time slots (M_n = *22* and N = 37). This traffic profile represents noise which does not indicate significant malicious activities.

We now compute the eigenvalues of the covariance matrix of obtained from the preprocessed dataset depicted in Fig. 1 and the eigenvalues of the covariance matrix of obtained from the noise only components of the preprocessed dataset depicted in Fig. 2. The eigenvalue profiles of the covariance matrices obtained from the full preprocessed dataset and the noise only components of are depicted in Fig. 3 and in Fig. 4, respectively. Comparing both eigenvalues profiles in log scale, the eigenvalues in Fig. 4 which do not represent malicious activities fit much better to the linear curve than the eigenvalues which indicate outstanding malicious activities.[1] In addition, by visual inspection, it is possible to estimate the model order in the malicious traffic in Fig. 3, which is clearly equal to 4 (as indicated by the break up in the linear eigenvalues profile, which behaves as a super-exponential profile).

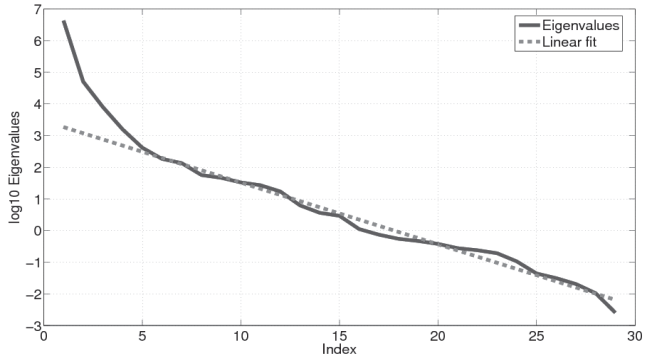

Figure 3: Malicious activity traffic plus noise eigenvalues profile compared to the linear fit. Plot of the logarithm base 10 of the eigenvalues λ_i vs the index i of the eigenvalues. The total of eigenvalues is $\alpha = m\mathit{in}(M, N) = 29$. The covariance matrix is computed via obtained from the complete preprocessed dataset shown in Fig. 1.

1 The exponential profile of the noise eigenvalues is a characteristic already observed in the literature. [8], [19], [5]

After analysing the eigenvalue profile in Fig 3, the raw collected honeypot network activity logs and the traffic profiles obtained in the preprocessed dataset it is possible to consistently estimate the model order as 4. While the traffic profile and the network activity logs indicate a high level of network activity in certain ports, further analysis of the collected data confirms that the connections to such ports pertain to 4 significant malicious activities, since the 4 destination ports targeted are typically used by commonly probed and attacked services. Furthermore, the break up in the eigenvalue profile of the covariance matrix obtained from the full preprocessed dataset also indicates that the model order is 4. Therefore, we conclude that the model order of the dataset used for the experiments proposed in this section is equal to 4, and consider this value as the correct model order for evaluating the accuracy of the several MOS schemes tested in the remainder of this section.

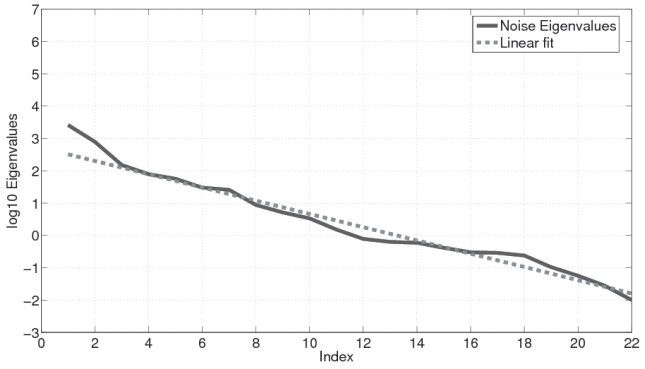

Figure 4: Noise only eigenvalues profile compared to the linear fit. The total of eigenvalues is $\alpha = min(M_r, N) = 22$. The covariance matrix is computed via obtained from the noise only preprocessed dataset shown in Fig. 2

As shown in this subsection, it may be possible to estimate the model order by visual inspection, manually determining the amount of malicious activities present in the dataset. Note that it was necessary to correlate raw collected network data, traffic profiles and information on common attacks in order to verify the correctness of the estimated model order. However, by visual inspection, the model order estimation becomes subjective, i.e., the model order of a same eigenvalue profile may vary for each person who inspects it, introducing an unacceptable uncertainty in the malicious activity identification process. Since the PoD of human dependent MOS schemes varies uncontrollably, it is impossible to guarantee a minimal probability of correctly detecting attacks and an average false positive percentage. Moreover, for real time applications and scenarios involving large quantities of data, it is necessary to employ an automatic scheme to estimate the model order.

I. Model order selection on the preprocessed dataset

In several scenarios it is not possible to visually identify the malicious traffic. However, in our data, this is possible. Therefore, in Section VI-B, we estimate the amount of malicious traffic, *i.e.*, the model order, through human intervention. Once the model order is known for our measured data from Section VI-B, we can apply our model order selection schemes presented in Section V. In this section, we verify the performance of these model order selection schemes, determining that RADOI with zero mean and noise pre-whitening is the most efficient and accurate method for analysing such data.

First, the zero mean zero mean is applied to the preprocessed dataset according to (8). After the application of zero mean (8) in the dataset shown in Fig. 1, the total amount of connections directed and originated from each port assumes negative values, which have no physical meaning but affect the PoD of several MOS schemes. The effect on the eigenvalues profile is almost insignificant when comparing the pure preprocessed dataset to the dataset after the application of zero mean. However, the accuracy of the model order selection schemes may vary when the zero mean is applied, even though it is insignificant for visual inspection purposes.

Note that the eigenvalues profiles obtained for the noise only and full dataset cases after applying the zero mean have similar characteristics to the eigenvalues profiles obtained for the preprocessed data before applying the zero mean, in the sense that the eigenvalues which do not represent malicious activities fit much better to the linear curve than the eigenvalues which indicate outstanding malicious activities. Moreover, it is also possible to clearly estimate the model order as 4 by visual inspection of the signal plus noise eigenvalues profile after zero mean.

Having preprocessed the original network flow dataset, applied the zero mean in the noise only dataset and applied the zero mean in the full dataset, we now proceed to actually estimating the model order of the original dataset. In order to evaluate each MOS scheme the model orders of both the full dataset (containing both noise and outstanding traffic) and the noise only dataset are estimated. In these experiments we estimate the model order using the following MOS schemes: 1-D AIC [26], [6], 1-D MDL [26], [6], efficient detection criterion (EDC) [29], Nadakuditi Edelman Model Order selection scheme (NEMO) [21], Stein's unbiased risk estimate (SURE) [25], RADOI [20] and KN [11].

Finally, the model order of the complete dataset after applying the zero mean is estimated, yielding the results shown in Table I.

Table 1: Model order selection via the eigenvalues of the covariance matrix of the signal plus noise samples.

AIC	MDL	EDC	SURE	RADOI	RADOI w/ PKT	KN	NEMO
21	21	13	11	3	4	11	13

In Table I, note that RADOI with prewhitening returns the correct estimation of the model order while the other MOS schemes fail. In other words, RADOI correctly detects the number of attacks in the analysed dataset. These results

validate our assumption that RADOI can successfully detect attacks in network traffic flow data obtained in honeypot systems, since it correctly estimates the model order as the number of attacks present in the dataset. Hence, we conclude that RADOI has the best performance in real world honeypot network flow data analysis via PCA.

7. Conclusions

In this paper we presented a blind automatic method for detecting malicious activities and attacks in network traffic flow data collected at honeypot systems. First we propose a dataset preprocessing model for network flow data obtained by many honeypot systems and we verify the validity of our approach through simulation results with real log files collected at a honeypot system in operation at the network of a large banking institution. Several model order selection methods were experimented with the preprocessed simulation data, showing that RADOI yields the best results for this type of data.

Honeypot traffic flow data behaves like measurements in signal processing, in the sense that if the traffic in honeypots does not represent significant attacks, the eigenvalues of the covariance matrix of the traffic samples have an exponential profile, linear in log scale. On the other hand, if connections are highly correlated (indicating significant malicious activities), a break appears in the exponential curve of the eigenvalues profile of the traffic samples covariance matrix. This break in the exponential curve profile indicates the model order which, in this case, represents the number of significant malicious activities observed in the honeypot data. The principal components and eigenvalues obtained can also be further analysed for identifying the exact attacks which they represent depending on which ports they are related to.

Since it does not require previous collection of large quantities of data nor adaptive learning periods, the solution proposed in the present work is an interesting alternative to classical honeypot data analysis methods, such as data mining and artificial intelligence methods. Since it is solely based on the correlation between network flows, it is capable of automatically detecting attacks in varying volumes of honeypot traffic without depending on human intervention or previous information. Thus, it eliminates the need for attack signatures and complex rule parsing mechanisms. As a future work, we point out further experimentation with other model order selection schemes in order to obtain an attack detection method that yields correct results even when malicious activities are not present in the analysed dataset (*i.e.* yield model order equal to zero).

References

[1] A. Quinlan and J.-P. Barbot and P. Larzabal and M. Haardt. Model Order Selection 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.

[2] Alata, E. and Dacier, M. and Deswarte, Y. and Kaniche, M. and Kortchinsky, K. and Nicomette, V. and Pham, V. H. and Pouget, F. Collection and analysis of attack data based on honeypots deployed on the Internet. In Gollmann, Dieter and Massacci, Fabio and Yautsiukhin, Artsiom, editors, Quality of Protection in Advances in Information Security, pages 79-91. Springer US, 2006.

[3] Almotairi, S. and Clark, A. and Mohay, G. and Zimmermann, J. A Technique for Detecting New Attacks in Low-Interaction Honeypot Traffic. Proceedings of the 2009 Fourth International Conference on Internet Monitoring and Protection, pages 7-13, Washington, DC, USA, 2009. IEEE Computer Society.

[4] Almotairi, S. and Clark, A. and Mohay, G. and Zimmermann, J. Characterization of Attackers' Activities in Honeypot Traffic Using Principal Component Analysis. Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing, pages 147--154, Washington, DC, USA, 2008. IEEE Computer Society.

[5] 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, :1177--1188, 2004.

[6] Abdallah Ghourabi and Tarek Abbes and Adel Bouhoula. Data analyzer based on data mining for Honeypot Router. Computer Systems and Applications, ACS/IEEE International Conference on, 0:1-6, 2010.

[7] H.-T. Wu and J.-F. Yang and F.-K. Chen. Source number estimators using transformed Gerschgorin radii. IEEE Transactions on Signal Processing, 43(6):1325--1333, 1995.

[8] Weisong He and Guangmin Hu and Xingmiao Yao and Guangyuan Kan and Hong Wang and Hongmei Xiang. Applying multiple time series data mining to large-scale network traffic analysis. Cybernetics and Intelligent Systems, 2008 IEEE Conference on, pages 394 -399, 2008.

[9] Hu, Y.H. Parallel eigenvalue decomposition for Toeplitz and related matrices. Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, pages 1107 -1110 vol.2, 1989.

[10] J. Grouffaud and P. Larzabal and H. Clergeot. Some properties of ordered eigenvalues of a Wishart matrix: application in detection test and model order selection. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96), pages 2463 -- 2466, 1996.

[11] J. P. C. L. da Costa. Parameter Estimation Techniques for Multi-dimensional Array Signal Processing. Shaker, First edition, 2010.

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

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

[14] Youngseok Lee and Wonchul Kang and Hyeongu Son. An Internet traffic analysis method with MapReduce. Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, pages 357 -361, 2010.

[15] Zhang Li-juan. Honeypot-based defense system research and design. Computer Science and Information Technology, International Conference on, 0:466-468, 2009.

[16] Yang Liu and Bouganis, C.-S. and Cheung, P.Y.K. and Leong, P.H.W. and Motley, S.J. Hardware Efficient Architectures for Eigenvalue Computation. Design, Automation and Test in Europe, 2006. DATE '06. Proceedings, pages 1 -6, 2006.

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

[18] Maheswari, V. and Sankaranarayanan, P. E. Honeypots: Deployment and Data Forensic Analysis. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) - Volume 04 in ICCIMA '07, pages 129--131, Washington, DC, USA, 2007. IEEE Computer Society.

[19] Mokube, Iyatiti and Adams, Michele. Honeypots: concepts, approaches, and challenges. Proceedings of the 45th annual southeast regional conference in ACM-SE 45, pages 321--326, New York, NY, USA, 2007. ACM.

[20] Provos, Niels. A virtual honeypot framework. Proceedings of the 13th conference on USENIX Security Symposium - Volume 13 in SSYM'04, pages 1--1, Berkeley, CA, USA, 2004. USENIX Association.
[21] 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, 56:2625--2638, 2008.
[22] Raynal, F. and Berthier , Y. and Biondi, P. and Kaminsky, D. Honeypot forensics. Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, pages 22 - 29, 2004.
[23] Raynal, Frederic and Berthier, Yann and Biondi, Philippe and Kaminsky, Danielle. Honeypot Forensics Part I: Analyzing the Network. IEEE Security and Privacy, 2:72--78, 2004.
[24] S. Kritchman and B. Nadler. Determining the number of components in a factor model from limited noisy data. Chemometrics and Intelligent Laboratory Systems, 94:19--32, 2008.
[25] Spitzner, Lance. Honeypots: Catching the Insider Threat. Proceedings of the 19th Annual Computer Security Applications Conference in ACSAC '03, pages 170--, Washington, DC, USA, 2003. IEEE Computer Society.
[26] Zhi-Hong Tian and Bin-Xing Fang and Xiao-Chun Yun. An architecture for intrusion detection using honey pot. Machine Learning and Cybernetics, 2003 International Conference on, pages 2096 - 2100 Vol.4, 2003.
[27] M. Wax and T. Kailath. Detection of Signals by Information Theoretic Criteria. IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-33:387--392, 1985.
[28] Feng Zhang and Shijie Zhou and Zhiguang Qin and Jinde Liu. Honeypot: a supplemented active defense system for network security. Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on, pages 231 - 235, 2003.
[29] Zhao, L C and Krishnaiah, P R and Bai, Z D. On detection of the number of signals in presence of white noise. J. Multivar. Anal., 20:1--25, 1986.