RESEARCH

Greatest Eigenvalue Time Vector Approach for Blind Detection of Denial of Service Attacks and Port Scanning

Danilo F Tenório[1*]
, João PCL da Costa[1]
, Edison P de Freitas[1,2]
, Thiago PB Vieira[1]
and Rafael T de Sousa Júnior[1]

*Correspondence:
????????????????
[1]Department of Electrical Engineering, University of Brasilia (UnB), , 70910-900 Brasília-DF, Brazil
Full list of author information is available at the end of the article
†Equal contributor

Abstract

The development of techniques for malicious traffic detection in computer networks is crucial to allow quick decisions to be taken regarding the implementation of safety countermeasures and to protect network devices, including end-user computers.

This work proposes an innovative signal processing technique for automatic blind detection of malicious traffic, by taking into account anomalies in monitored traffic on a network. First, we model our acquired data as a superposition of three traffic types: legitimate traffic related to user applications, noise traffic not associated with user, and malicious traffic. In practice, the three traffics are mixed, and it is hard to analyze them separately. Thus, considering this model, we propose the Greatest Eigenvalue Time Vector (GETV) approach, which successfully and blindly detects only the malicious traffic. Since our scheme is blind, no training is necessary. Moreover, no human intervention is required. We validate our proposed approach by detecting denial of service (synflood and fraggle) attacks and scan of communication ports (port scan), using a real computer network.

Keywords: Intrusion Detection System; Eigenvalue; Principal Component Analysis; Model Order Selection

1 Intrduction

The need for information security is a fact that has transcended the limits of productivity and functionality in computer systems. While the speed and the efficiency in all business processes are a competitive advantage, the lack of security that compromises speed, efficiency and other network properties can result in major damage and lack of new business opportunities. The methods of defense used by an organization can be effective against certain types of attacks, but perhaps fail against new malicious techniques [1].

In this context, a major challenge in a communication network is to guarantee integrity, availability and confidentiality to security related data. In the terms of security, there are both technical and procedural aspects. The former includes equip-

ment and security systems, while the latter corresponds to security polices and staff awareness campaigns.

Intrusion detection and intrusion prevention systems are security systems used respectively to detect (passively) and prevent (proactively) threats to computer systems and computer networks. Such systems use several ways of working, such as: signature-based, anomaly-based or hybrid [2].

In the context of anomaly-based schemes, in this work, we propose an automatic blind malicious traffic detection technique, for using on any computer in a network. Note that the term “automatic” means that human intervention is not necessary to assess whether or not there was an attack. The term “blind” refers that it is not necessary prior information, such as attack signatures or learning periods, to detect the attack.

In [3, 4] the real network traffic data was modeled into three components: legitimate traffic, malicious traffic and noise. Inspired by [3, 4], this work models the network traffic using a signal processing formulation as a composition of three components: legitimate traffic, malicious traffic and noise, taking into account the incoming and outgoing traffic in certain types of ports (TCP or UDP).

Our proposed technique is based on eigenvalue decomposition, however, in contrast with [3, 4], we consider the time variation of the eigenvalues. To the best of our knowledge, the time variation of the eigenvalues was not applied before in the literature. We show, through experiments applying our greatest eigenvalue variation scheme, that attacks such as synflood, fraggle and port scan can be detected in automatic and blind fashion.

The main contributions of this work are: general network traffic modeling by applying signal processing concepts, development of the greatest eigenvalue time vector (GETV) technique, and its validation by detecting synflood, fraggle and port scan attacks.

This paper is organized as follows. In Section 2, related works are discussed. The mathematical notation used in the following sections is presented in Section 3. Section 4 presents mathematical concepts and the concepts of eigenvalues and eigenvectors, principal component analysis (PCA), and model order selection (MOS), including the main MOS schemes and their differences. Section 5 presents the data model used during our experiments. Section 6 describes the proposed greatest eigenvalue time vector (GETV). Section 7 discuss the experimental validation and presents the corresponding experimental results. In Section 8 are presented the final remarks and suggestions for future works.

2 Related Works

Several methods have been proposed for identification and characterization of malicious activity in computer networks. Classical methods typically employ data mining [5, 6] and regular file analysis [7] to detect patterns that indicate the presence of specific attacks in traffic analysis.

Data mining is often used to describe the process of extracting useful information from large databases. Multiple methods of data mining are used in [5] to analyze data flow in a network, with the aim of identifying characteristics of malicious traffic in large scale environments. Researchers have applied data mining techniques in log

analysis [6] to improve the intrusion detection performance. However, data mining techniques requires the prior collection of large data sets, what is a weakness of several schemes for online or low latency analysis.

Regular file analysis [7] consists in use traffic analysis for detecting known patterns that indicate the presence of specific attacks, applying statistical analysis on the the study of collected traffic. An essential feature of this method is that it depends on prior knowledge about details of the attacks to be identified, and also depends on the previous collection of logs for applying traffic analysis and reducing false positives.

PCA is a statistical technique commonly used for dimensionality reduction, it uses an orthogonal transformation to convert a set of correlated variables into a set of linearly uncorrelated variables, where the first principal components has the largest variance. PCA can be used in attack detection [8], however, if PCA is used without combination with any other technique, such as model order selection (MOS), it is necessary the subjectiveness of the human intervention, making it prone to errors, such as false positive cases, and inefficient for automatic detection systems.

Blind automatic detection of malicious traffic techniques has been developed to honeypots in [3, 4]. However, traffic on honeypot is simpler than real traffic, because there are no legitimate applications running. A honeypot emulates behavior of a host within a network to deceive and lure attackers [9].

The data collected in honeypot systems, such as captured traffic and operating system logs, are analyzed to obtain information about attack techniques, general trends of threats and exploits. Due to honeypots do not generate legitimate traffic, the amount of data captured in honeypots is significantly lower in comparison to a network IDS, which captures and analyzes the largest possible amount of network traffic [3].

The use of model order selection for blind detection in network traffic, to identify malicious activities in honeypots, was proposed by [3]. Criteria for selecting the model order are usually evaluated through simulations and comparing the order of the resulting model with the true model order [10].

Our approach treats a more complex network traffic, which is composed of legitimate, noise and attack signals. In contrast with [5–7], our approach does not require either significant amount of logs to detect attacks, nor the prior data collection, in order to make comparisons and evaluate the existence of malicious traffic. Moreover, in contrast with [8], the proposed attack detection approach is automatic and require no human intervention.

3 Mathematical Notation

In this paper the scalars are denoted by italic letters (*a, b, A, B, α, β*), vectors by lowercase bold letters (**a, b**), matrices by uppercase bold letters (**A, B**), and $a_{i,j}$ denotes the (*i, j*) elements of the matrix **A**. The superscripts ^{T} and $^{-1}$ are used for matrix transposition and matrix inversion, respectively.

4 Mathematical Concepts

We present the following mathematical concepts used in this study in order to network attack detection: eigenvalues and eigenvectors, principal components analysis (PCA) and model order selection (MOS).

In the context of malicious traffic detection, the superpositions of matrices can represent the different types of traffics associated with each communication port. Eigenvalues and eigenvectors, which are usually used in linear algebra, can reveal important information about matrix data structure, and can be used for noise reducing or spectral decomposition.

Principal Components Analysis is a multivariate analysis technique that uses an orthogonal transformation to convert a set of correlated variables into a set of linearly uncorrelated variables. PCA has been widely used in different research areas, such as: dimensionality reduction, Internet traffic analysis, economy, image processing, and genetics. PCA is mainly used to reduce the data set size, using uncorrelated variables for this, which are called the principal components (PC). This transformation of variables into a reduced set of variables, occurs with low information loss or even reduces variables that represent noise signals [11].

The principal components, generated through PCA, are a linear combination of the original variables, which are orthogonal and ordered by variance. Thus, the first principal component has the greatest variance of the original data. Although the resulting number of principal components can result on no feature reduction, preserving the original number of variables, most of the variation in the original data set can be retained by the first principal component, what can reduce the size of the problem [12] to principal components.

4.1 Model Order Selection (MOS)

The model order selection is a key point in many digital signal processing applications, including radar, sonar, communications, channel modeling, medical imaging, among others. MOS allows analysis of reduced data set, through separating noise components of the main components, for example. Moreover, the model order is crucial for many parameter estimation techniques [13], since the amount of parameters to be estimated depends on the model order.

The model selection procedure chooses the "best" model of a finite set of models, according to some criterias [14]. Therefore, given some data set, it is chosen a model which was evaluated as the best model to describe the specified data set.

The state of the art regarding estimation techniques of model order based on eigenvalues includes: Akaike's Information Theoretic Criterion - AIC [15, 16]; Minimum Description Length - MDL [16, 17]; Efficient Detection Criterion - EDC [18]; Stein's Unbiased Risk Estimator - SURE [19]; RADOI [20] and Exponential Fitting Test - EFT [3, 21, 22].

In AIC, MDL and EDC techniques, the information criterion is a function of the geometric mean *g(k)* and the arithmetic mean *a(k)* relating to smaller k eigenvalues, where k is a candidate value for the model order d [13].

Basically, the difference between the AIC, MDL and EDC schemes is the penalty function *p(k, N, α)*, so these techniques can be written in general as [13]:

$$\hat{d} = \arg\min_{k} \; J(k), \tag{1}$$

where

$$J(k) = -N(\alpha - k) \log \left(g(k)/a(k)\right) + p(k, N, \alpha), \tag{2}$$

where $\hat{d}$ is an estimate d of the model order, N is the number of samples, $\alpha = M$ and means the number of variables of the problem, and $0 \leq k \leq \min[M, N]$. Penalty functions for AIC, MDL and EDC are given by the Table 1.

Table 1 Penalty functions for the schemes AIC, MDL and EDC

Scheme	Penalty function $p(k, N, \alpha)$
AIC	$k(2\alpha - k)$
MDL	$0.5k(2\alpha - k)\log(N)$
EDC	$0.5k(2\alpha - k)\sqrt{N\ ln(lnN)}$

The Exponential Fitting Test (EFT) can effectively be used in cases where the number of samples N is small. This technique is based on observations of data contaminated only with white noise, where the profile of eigenvalues can be approximated by a exponential decaying [21].

Given λ_i be the i-th eigenvalue, the exponential model can be expressed by:

$$E\{\lambda_i\} = E\{\lambda_1\} \cdot q(\alpha, \beta)^{i-1}, \tag{3}$$

where $E\{\cdot\}$ is the expectation operator, and it is considered that the eigenvalues are ordered in the that λ_1 represents the largest eigenvalue. The term *q(α, β)* is defined as:

$$q(\alpha, \beta) = \exp\left\{-\sqrt{\frac{30}{\alpha^2+2} - \sqrt{\frac{900}{(\alpha^2+2)^2} - \frac{720\alpha}{\beta(\alpha^4+\alpha^2-2)}}}\right\}, \tag{4}$$

where $0 < q(\alpha, \beta) < 1$. According to [22], if $M \leq N$, then $\beta = N$.

Figure 1 shows a typical profile of eigenvalues. The last P – 1 eigenvalues are used to estimate the (M - P)-th eigenvalue, denoted by the yellow rectangle. The EFT method considers the discrepancy between the actual value and the estimated value obtained [10].

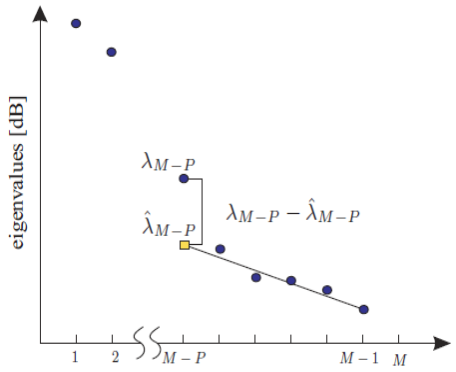

Figure 1 EFT Example. Example of application of EFT [10].

5 Data Model

5.1 Data Collection

The network traffic log, of a computer connected to a network, is formed by timestamp, protocol, source IP address, source port, destination IP address, destination port and additional information, according to the type of transport protocol used.

In order to exemplify the collected data, the following TCP traffic log can be considered:

```
21:00:34.099289 IP 192.168.1.102.34712 > 200.221.2.45.80: Flags
[S], seq 2424058224, win 14600, options [mss 1460, sackOK,TS val
244136 ecr 0,nop,wscale 7], length 0
```

and the following UDP traffic log:

```
21:24:42.484858 IP 192.168.1.102.68 > 192.168.1.1.67: BOOTP/DHCP,
Request from 00:26:9e:b7:82:be, length 300
```

In this paper, it is considered only the following information from the log data: timestamp, port type and port number.

5.2 Modeling Data

The network traffic ($\mathbf{X}$) can be characterized as a superposition of three components: legitimate traffic ($\mathbf{S}$), noise ($\mathbf{N}$) and malicious traffic ($\mathbf{A}$), according to the following expression:

$$\boldsymbol{X}^{(q)} = \boldsymbol{S}^{(q)} + \boldsymbol{N}^{(q)} + \boldsymbol{A}^{(q)}, \tag{5}$$

where q represents the q-th period of time.

According to Figure 2, the data collected were divided into q periods of N samples, where each sample is collected at a given time, according to a sampling period.

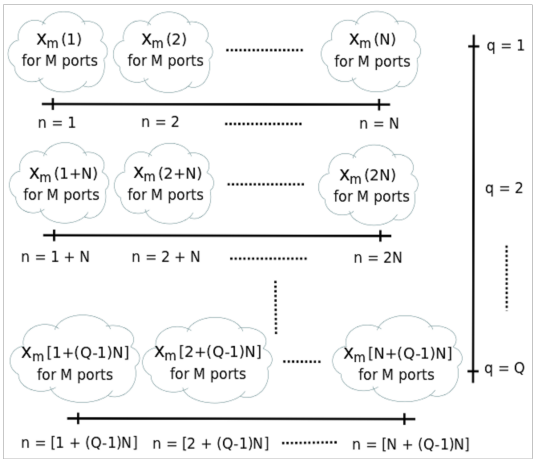

Figure 2 Traffic Matrix. Obtaining the traffic matrix $\boldsymbol{X}^{(q)}$.

The matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{mxn}$ consists of M rows and N columns, where each row is represented by a variable, in this case a communication port (TCP port or UDP port), and each column a second time. Each element $x_{m,n}^{(q)}$ represents the number of times that the port m appears in the n-th instant, in the q-th time period.

The legitimate traffic $\boldsymbol{S}^{(q)}$ is characterized by traffic associated directly to the operations performed by the user. When a user accesses a web page, for example, there is the corresponding TCP/IP traffic to request the page as well as to the traffic due to name resolution (DNS). Figure 3 presents the legitimate traffic obtained during experiments.

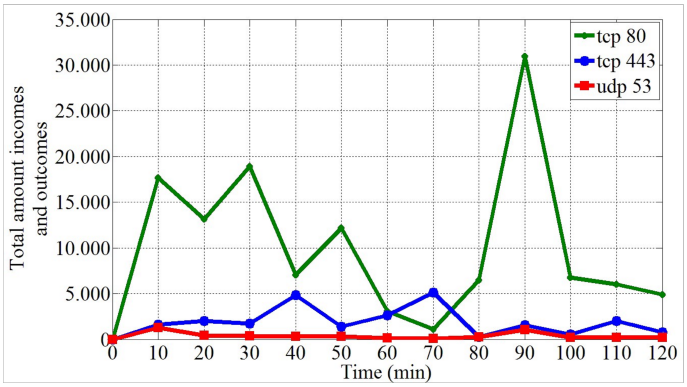

Figure 3 Legitimate Traffic. The legitimate traffic.

All traffic that is not directly associated with operations performed by the user, but it is not a malicious traffic, is considered as noise $\boldsymbol{N}^{(q)}$. The automatic acquisition service of logical IP network address (DHCP) is an example of noise. Independently of any user operation, the machine will receive an IP address, since it is configured to perform a DHCP address request. Figure 4 shows the noise during simulations.

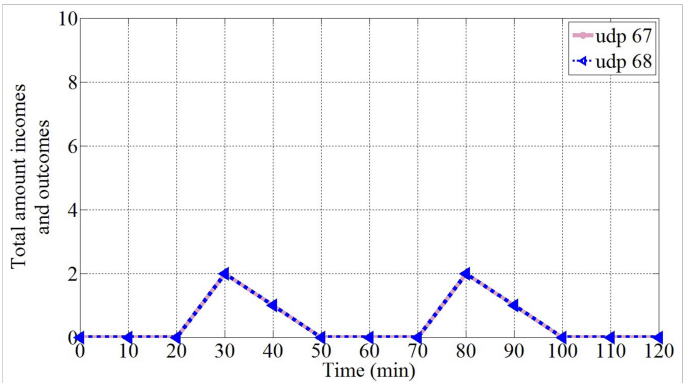

Figure 4 Noise Traffic. The noise traffic.

The traffic coming from a malicious activity, such as a synflood or fraggle attack, is represented by the matrix $\boldsymbol{A}^{(q)}$. For this work ee only consider the traffic from port scanning and flood attacks, which aims to cause denial of service (DoS). We

defined that if the obtained rank $\{\boldsymbol{A}^{(q)}\} \neq 0$, then there is malicious traffic, on the other hand, if the rank $\{\boldsymbol{A}^{(q)}\} = 0$, then there is no malicious traffic. This paper shows how to detect the rank $\{\boldsymbol{A}^{(q)}\}$, given only the matrix $\boldsymbol{X}^{(q)}$, in order to identify malicious network traffic.

5.3 Synflood, Fraggle and Port scan

The kind of network attacks focused by this work are: synflood, fraggle and port scan. The first two attacks can be qualified as denial of service attacks, while the last one is a port scanning attack.

The TCP protocol is a connection-oriented protocol, then a virtual connection must be established between two computers for a end-to-end TCP communication. This virtual connection requires a “handshake”, that occurs in three steps, known as three-way handshake. If a computer needs to communicate with another computer, the requester sends a packet communication synchronization (SYN) to a specific destination port, which is in listening state. If the destination is active, running and accepting requests, it responds to the requester with a SYN/ACK confirmation message. After receiving this message, the requester sends an ACK message to the destination and then the connection is established.

On synflood attacks, the attacker sends a large quantity and concurrent successive SYN requests to a target, in order to consume resources and cause a denial of service. Figure 5 represents the synflood attack carried out during our simulations. In a interval of ten minutes, were sent more than 210,000 packets as a synflood attack. This network traffic behavior can be considered an unusual data traffic on a network, especially because it is concentrated in a short period of time and due to the similarity of the outstanding traffic.

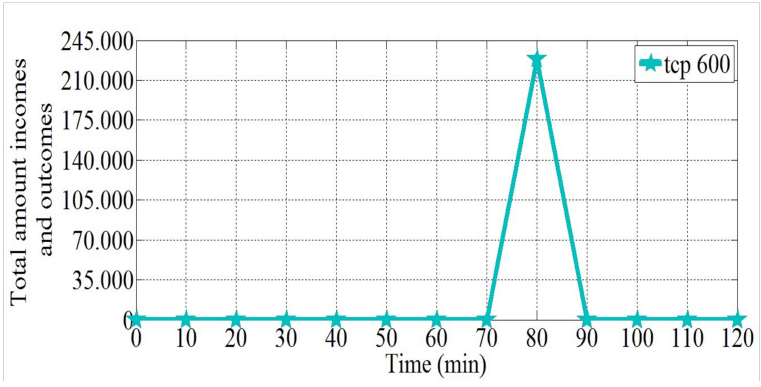

Figure 5 Synflood Traffic. The traffic characterized by synflood.

In the fraggle attack, large packets with “UDP echo” segments are sent to the broadcast address of a network, with every sent packet previously modified to have the source address of the victim, this source address modification is a spoofing technique. Thus, each host receives a huge amount of requests “UDP echo” and all of them replies to the IP address of the victim, in order to cause a denial of service. This attack can affect the entire network, because all hosts receive many requests “UDP echo” and respond with the ICMP protocol, then each host acts as

an "amplifier" of the attack. Thus, the victim that has the fake IP address receives packages from all these hosts, being unable to perform their normal activities and suffering a denial of service. This last part of the fraggle attack will not be taken into account in this work, because the victim receives ICMP (network layer) packets originated from the hosts that were attacked with flooding packet "UDP echo". This occurs due to the UDP be not able to know if the segment sent has reached its destination, i.e. as UDP is connectionless, no confirmation is sent back.

Figure 6 depicts the fraggle attack carried out during the experiments. More than 6,000,000 malicious packets can be counted in an interval of ten minutes, which can be considered an unusual network traffic, especially due to the concentrated traffic in a short period of time and the similarity of the outstanding traffic.

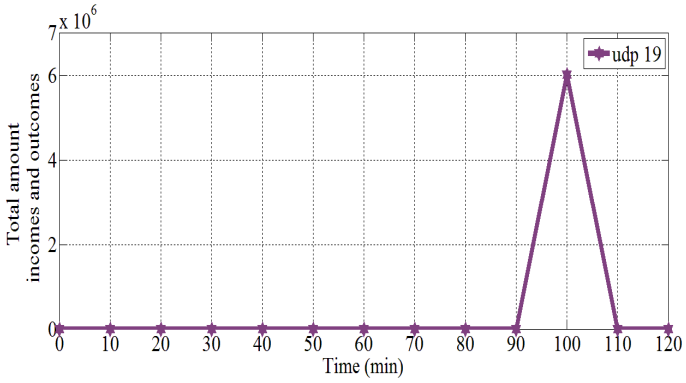

Figure 6 Fraggle Traffic. The traffic characterized by fraggle.

Port scan is the process of connecting to TCP and UDP ports of targets to identify what services are running, or which are in the state of listening. Identifying listening ports is crucial to determine the type of the victim's operating system and running applications. There are several available port scanning techniques, including: TCP SYN scan, TCP ACK scan, UDP scan, etc. This work makes use of TCP SYN scan and UDP scan.

The TCP SYN scan technique is called half-open scanning, because there is no full TCP connection. In this scan, a SYN packet is sent to the destination port and two types of response may occur: SYN/ACK is received or RST/ACK packet is received. In the first case, the destination port is in listening state, in the second case, the destination port is not listening. Then, a RST/ACK packet is sent by the system that is performing the port scan, at the end of each port scanning. Thus, a full connection or a complete three-way handshake is never established. This approach makes more difficult the detection of the attack sender, since it is not registered on the target system.

The technique of UDP scan sends UDP packets to the destination port, if the port responds with a message "ICMP port unreachable" indicates that the scanned port is closed. If a message is not received, then the port can be considered as open. UDP is a connectionless protocol and the efficacy of this technique is dependent on many factors related to network and system resources configurations. This type of scanning is also very slow and can produce uncertain results.

Figure 7 depicts the port scan attack that was experimented. It is possible to observe that the traffic is composed of two packets for each TCP port and a UDP packet to each port. These practical results are aligned with what was explained about this attack. It is important to observe the high correlation of TCP and UDP traffic, separately, since the traffic related to the TCP ports are similar and the traffic related to the UDP ports are also similar. The mentioned similarity of the traffics refers to the amount of incoming and outgoing packets for each port.

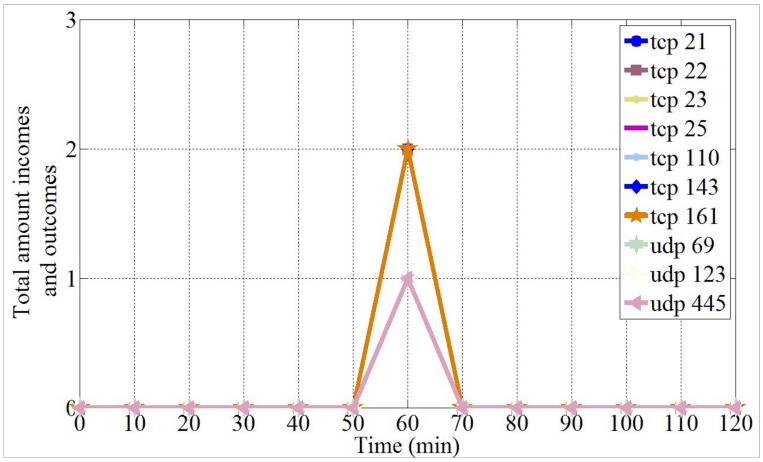

Figure 7 **Port scan Traffic.** The traffic characterized by port scan.

6 Greatest Eigenvalue Time Vector (GETV)

In this section, we present the GETV technique to detect synflood, fraggle and port scan attacks in any computer. The Table 2 shows a comparison between the discussed related works and the presented approach [23], showing the proposed new technique (GETV), which is detailed throughout this work.

Table 2 Comparison between malicious traffic detection schemes

Related Works	Techniques				
	Data Mining	Regular analysis of files	PCA	MOS	GETV
(He et al, 2008)	x	-	-	-	-
(Raynal et al, 2004)	-	x	-	-	-
(Almotairi et al, 2009)	-	-	x	-	-
(da Costa et al, 2012)	-	-	x	x	-
(Proposed)	-	-	x	x	x

The Table 2 shows the techniques used by related works and our proposed approach in order to malicious traffic detection. The techniques were classified according to the adopted approaches, as data mining, regular file analysis, PCA, and MOS. Additionally, a column with the GETV technique was added, showing that our proposed approach is the unique work that combines PCA, MOS and the Greatest Eigenvalue Time Vector approach.

The GETV attack detection technique can be described as a algorithm for malicious traffic detection, that receives as input a data set containing network traffic log, applies a pre-processing and different initial steps for detecting DoS or port

scan attacks, and then executes common tasks for attack detection of these kind of malicious traffic.

The attack detection algorithm starts by the data pre-processing, that receives a traffic log of network connections, providing information above protocols, IP, ports and timestamp of senders and receivers, and then performs a data classification and grouping, where the network connections are classified and counted according to the origin and destination ports, and grouped by q periods of time.

With the data grouped into q periods, it is possible start an iteration at $q = 1$ over the q values until $q = Q$, in order to obtaining the network traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{MxN}$ and to perform the algorithm for detecting the desired type of attack.

According to the behavior of denial of service and port scan attacks, it is possible to characterize denial of services attacks as a covariance aware attack [24] and port scan attacks as correlation aware attack [25]. These characteristics are substantiated by the results obtained through the principal component analysis described below, which shows that the main components of DoS attacks are dominated by the variables with more variance and that the traffic associated with port scan attack does not generate many logs, however port scan attack presents a highly correlated traffic.

Thus, in our approach for detection of denial of service attacks, specifically syn-flood and fraggle attacks, it is necessary to calculate the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$. To obtain the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$ it is necessary, for each variable (we adopted network ports as our variables), to calculate the deviations of the respective elements in relation to the average, which is done by the equation 6.

$$\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)} \tag{6}$$

The set of obtained vectors $\boldsymbol{y}_m^{(q)}$ composes the matrix $\boldsymbol{Y}^{(q)}$, then the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$ can be calculated through the equation 7.

$$\boldsymbol{S}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T} \tag{7}$$

For the port scan attack detection, it is necessary to calculate the correlation matrix $\boldsymbol{R}_{xx}^{(q)}$, instead of the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$ used for DoS detection, since the main components are not dominated by the variables with large variance. The traffic associated with port scan attack does not generate many logs, however port scan attack presents a highly correlated traffic. To obtain the correlation matrix $\boldsymbol{R}_{xx}^{(q)}$ it is required, for each variable, to calculate the deviations of the respective elements in relation to the average, divided by the standard deviation, this calculation is done by the equation 8.

$$\boldsymbol{y}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}} \tag{8}$$

The set of vectors $\boldsymbol{y}_m^{(q)}$ composes the matrix $\boldsymbol{Y}^{(q)}$, then the correlation matrix $\boldsymbol{R}_{xx}^{(q)}$ can be calculated through the equation 9.

$$\boldsymbol{R}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T} \tag{9}$$

Once the $\boldsymbol{S}_{xx}^{(q)}$ and $\boldsymbol{R}_{xx}^{(q)}$ has been obtained for DoS and port scan attack detection, respectively, the next step of the algorithm is the eigenvalue decomposition (EVD), calculated through the equation 11 in order to obtain the vector of eigenvalues $\boldsymbol{E}_m^{(q)}$ associated with each matrix.

$$\boldsymbol{E}_m^{(q)} = \boldsymbol{E}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{E}^{(q)^T} \tag{10}$$

The obtained vector of eigenvalues $\boldsymbol{E}_m^{(q)}$ is composed by $\lambda_m^{(1)}, \lambda_m^{(2)}, \lambda_m^{(3)}, ..., \lambda_m^{(q)}$. The obtained eigenvalues should be sorted in descending order, as defined by $\lambda_m^{(1)} > \lambda_m^{(2)} > \lambda_m^{(3)} > ... > \lambda_m^{(q)}$, to make possible the selection of the first eigenvalue in the obtained sequence, represented by $\lambda_m^{(1)}$, which is the greatest eigenvalue of the period of time evaluated for attack detection.

The matrix of eigenvalues of $\boldsymbol{S}_{xx}^{(q)}$ or $\boldsymbol{R}_{xx}^{(q)}$ can be represented as the matrix $\boldsymbol{K} \in \mathbb{R}^{MxQ}$, as shown in equation 11.

$$\boldsymbol{K} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}, \tag{11}$$

The process of obtaining the $\boldsymbol{X}^{(q)} \in \mathbb{R}^{MxN}$, $q = 1, 2, 3, ..., Q$ and the matrices $\boldsymbol{S}_{xx}^{(q)}$ or $\boldsymbol{R}_{xx}^{(q)}$, finding the greatest eigenvalue for each q-th time period, repeats the specified tasks until $q = Q$. From this process came the term "Greatest Eigenvalue Time Vector", as defined in the title of this paper, which it is related to the greatest eigenvalue for each q-th period of time.

Since $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, the first line of the matrix $\mathbf{K}$ contains the Greatest Eigenvalue Time Vector (GETV) of the time evaluated for attack detection, which is $\lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \cdots, \lambda_1^{(q)}$.

Obtaining the vector GETV, it is possible to apply the schemes of MOS in order to estimate the model order $\hat{d}$, until the tested MOS scheme presents the estimated model ($\hat{d}$) equal to the true model order (d). Thus, if $\hat{d} = d$ then it was discovered which MOS scheme applies to the problem, if the tested MOS scheme do not presents the estimated model, i.e. $\hat{d} \neq d$, it is necessary a new iteration to apply the schemes of MOS to estimate the model and to evaluate the obtained result, this iteration should be done until the estimated model ($\hat{d}$) be equal to the true model order (d).

It is possible to find more than one scheme that applies to the problem, not necessarily only one, according to the selected schemes to be evaluated and according

to characteristics of the evaluated data set. With our approach, the algorithm of attack detection ends when the first model that satisfies the condition $\hat{d} = d$ is found, representing the discovery of what MOS scheme applies to the evaluated problem.

The GETV attack detection algorithm can be understood analyzing the algorithm 1, that represents all tasks of the described technique.

Algorithm 1 GETV Attack Detection

Input: Network Traffic Log

Output: GETV

1: $\mathbf{X}$ = matrix of ports and its occurrences per time {*Data Pre-Processing*}
2: Q = number of periods
3: **loop** $q = 1$ util $q = Q$
4: $\boldsymbol{X}^{(q)} \in \mathbb{R}^{MxN}$
5: **if** $isDosAttack$ **then**
6: $\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}$
7: $\boldsymbol{S}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T}$
8: **end if**
9: **if** $isPortscanAttack$ **then**
10: $\boldsymbol{y}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}}$
11: $\boldsymbol{R}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T}$
12: **end if**
13: $\boldsymbol{E}_m^{(q)} = \boldsymbol{E}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{E}^{(q)^T}$
14: $[\lambda_m^{(1)}, \lambda_m^{(2)}, \lambda_m^{(3)}, ..., \lambda_m^{(q)}] = \boldsymbol{E}_m^{(q)}$
15: $\lambda_m^{(1)} > \lambda_m^{(2)} > \lambda_m^{(3)} > ... > \lambda_m^{(q)}$
16: **end loop**
17: $$\boldsymbol{K} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}$$
18: $\boldsymbol{K}_1^q = \lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \cdots, \lambda_1^{(q)}$
19: $\boldsymbol{GETV} = \boldsymbol{K}_1^q$

7 Experimental Results

In this section, it is presented the analyzed scenario and all obtained results, that can be grouped by eigenvalues, GETV, principal component (PC), and MOS scheme.

7.1 Analyzed Scenario

The environment of the analyzed scenario is composed by two computers and a router with access to Internet and to an internal network (LAN). One of the computers has the role of attacker, while the other is the victim, according to Figure 8.

During the evaluation, the victim performs legitimate activities, that can be characterized mainly by web access. In many organizations this type of access is done

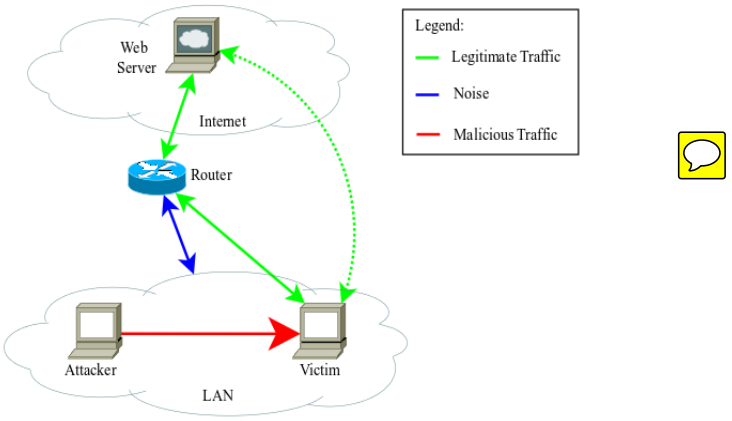

Figure 8 Scenario. Analyzed Scenario.

frequently and is the predominant traffic, since most of corporate services are web-based, such as: access to the webmail services, access to documents and access to intranet or internet pages.

It is possible to cite the traffic of a DHCP service as an example of noise associated with the transport layer of the OSI model. Our malicious traffic is composed by the traffic associated with three types of attacks: synflood, fraggle and port scan. The attacks were simulated using well known professionals security tools. Nmap was used to port scan, Metaspoit was used to synflood attack and Hping was used to lead the fraggle attack.

The total experiment time was one hundred twenty minutes, separated into six periods, with each time period corresponding to twenty minutes. Therefore, as the time of each sampling period is one minute, then $N = 20$.

For each time period q, a traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17x20}$ was obtained, as well as a covariance $\boldsymbol{S}_{xx}^{(q)} \in \mathbb{R}^{17x17}$ and a correlation matrix $\boldsymbol{R}_{xx}^{(q)} \in \mathbb{R}^{17x17}$, assuming that in this paper $q = 1, 2, 3, 4, 5$ and 6. The simulation started at 21:00h, the first period was from 21:00h until 21:20h ($q = 1$), the second was from 21:20h until 21:40h ($q = 2$), the third was from 21:40h to 22:00h ($q = 3$), the fourth was from 22:00h until 22:20h ($q = 4$), the fifth was from 22:20h until 22:40h ($q = 5$), and finally, the sixth was from 22:40h until 23.00h ($q = 6$).

During the simulation, the victim made legitimate access, and the attacker, at certain times, executed the following attacks: at 21:54h ($q = 3$) was performed a port scan, at the interval ranging from 22:10h to 22:20h ($q = 4$) a synflood attack was simulated, and at the interval from 22:30h to 22:40h ($q = 5$) a fraggle attack was performed.

7.2 Eigenvalues

Figure 9 graphically represents the eigenvalues of the matrix used for the detection of synflood. In this figure its possible to see that the greatest eigenvalue, which is related to this attack, stands out from the others.

Figure 10 graphically represents the eigenvalues of the matrix used for the detection of fraggle. In this figure it is possible to see that the greatest eigenvalue, which

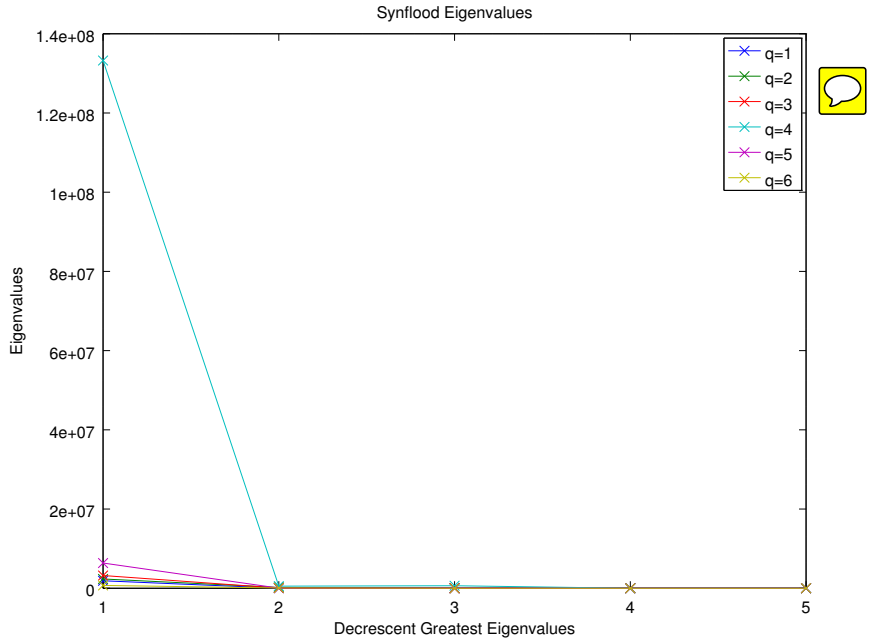

Figure 9 Synflood Eigenvalues. Eigenvalues of the covariance matrix (synflood).

is related to this attack, stands out from the others, as shown in Figure 9 for the synflood attack.

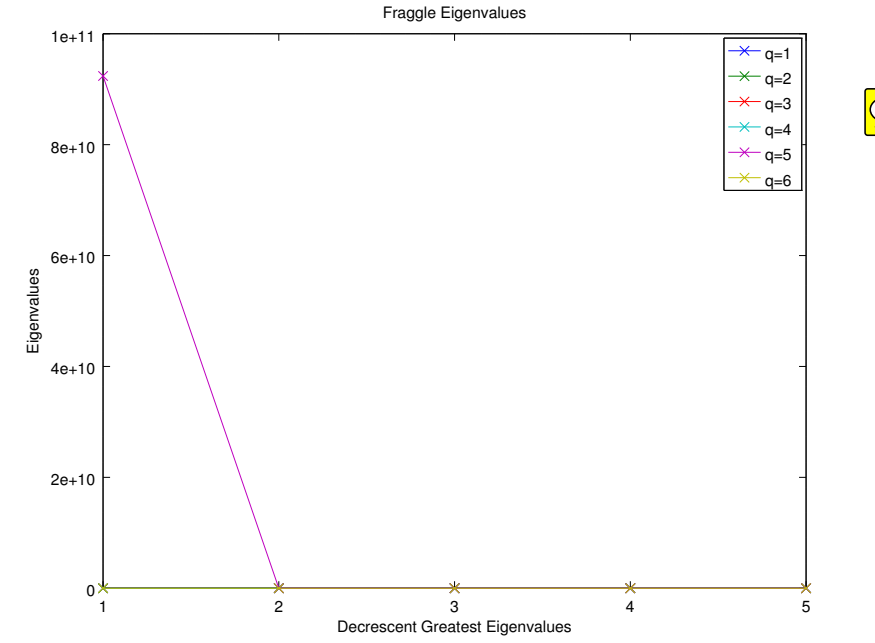

Figure 10 Fraggle Eigenvalues. Eigenvalues of the covariance matrix (fraggle).

Figure 11 graphically represents the eigenvalues of the matrix used for the detection of port scan. The same way as analyzed for the synflood and fraggle attacks, it is possible to observe the greatest eigenvalue, related to this attack, standings out from the others.

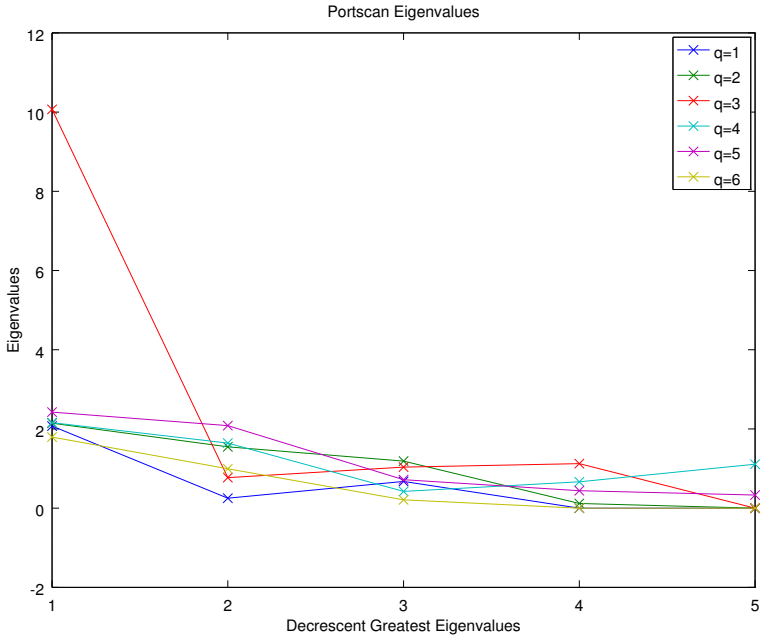

Figure 11 Port scan Eigenvalues. Eigenvalues of the covariance matrix (port scan).

7.3 GETV

Table 3 presents the vectors formed by the greatest eigenvalue time vector. These vectors were used as parameters for model order selection (MOS) and thus to the detection of the proposed attacks.

In Table 3 it is possible to observe the differences of the eigenvalues associated with attacks, in comparison to the others. At $q = 4$, where the synflood attack occurred, the maximum eigenvalue obtained, was approximately 21 times larger than the second one. At $q = 5$, where the fraggle attack occurred, the maximum eigenvalue obtained was about 29,000 times larger than the second one. At $q = 3$, where the port scan attack occurred, the maximum eigenvalue obtained was approximately 4 times larger than the second one. In the last case, although the greatest eigenvalue was not too high, compared to synflood or fraggle attacks, it was entirely sufficient to detect the port scan, as it clearly deviates from the rest of the values.

7.4 Principal Components

As presented before, the principal component analysis is mainly used to reduce the data set size, through the use of uncorrelated variables, called principal components (PC). This data transformation occurs with the least possible loss of information, eliminating only some unique variables that have less information.

Table 3 Greatest Eigenvalue related to attacks detection

Time Period q	Vectors GETV			
	Detection of synflood/fraggle	Detection of synflood	Detection of fraggle	Detection of port scan
1	1887545	1887545	1887545	2,0734
2	2341327	2341327	2341327	2,1451
3	3213867	3213867	3213867	10,0718
4	133238294	133238294	731229	2,1620
5	92384021611	6367983	92384021611	2,4253
6	708335	708335	708335	1,7948

The principal components are a linear combination of the original variables, they are orthogonal and ordered to the first principal component that has the greatest variance, related to the original data. Although the resulting number of principal components can be equal to the original number of variables, most of the variation in the original data set can be retained by the first principal component, thereby reducing the problem size.

According to the evaluated scenario, the variables are communication ports: tcp 80, tcp 443, udp 53, tcp 21, tcp 22, tcp 23, tcp 25, tcp 110, tcp 143, tcp 161, udp 69 , udp 123, udp 445, tcp 600, udp 19, udp 67 and udp 68. Thus, the main components are formed by linear combinations of these variables.

As there are 17 variables, then the data set is 17-dimensional. With PC technique, the data set can be reduced, for example, to two dimensions, presented by the first two principal components. With this, it is possible to reduce the size of the data set without loss of relevant information.

The principal components are obtained from the eigenvectors of the covariance or correlation matrix. As we selected only the first two principal components, it is necessary to select the two eigenvectors related to the two largest eigenvalues of covariance or correlation matrix.

In order to show that attacks present different and dominant behavior, in comparison to other kind of network traffic, it was selected the periods related to these attacks: $q = 3$ for port scan attack, $q = 4$ for synflood attack and $q = 5$ for the fraggle attack.

Synflood attack presented in Figure 12 shows that the variance of PC1 (first PC) is totally dominated by attack components. Attack components are responsible for high value of the eigenvalue associated with this principal component, and consequently for the values of the period $q = 4$. Furthermore, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{S}_{xx}^{(4)}$.

For the fraggle attack represented in Figure 13, the variance of PC1 (first PC) is totally dominated by the components of the attack. These components are responsible for the high value of the eigenvalue associated with this principal component, and consequently to the set of values of the period $q = 5$. Furthermore, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{S}_{xx}^{(5)}$.

For the port scan attack in Figure 14, the variance of PC1 (first PC) is totally dominated by the components of the attack. These components are responsible for the high value of the eigenvalue associated with this principal component, consequently to the set of values of the period $q = 3$. As already discussed, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{R}_{xx}^{(3)}$.

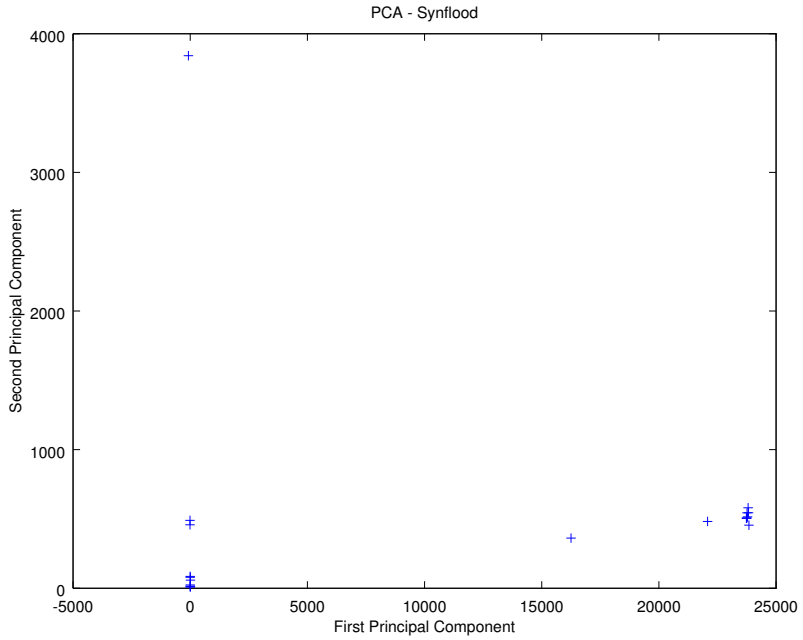

Figure 12 Synflood Components. The two first principal components (synflood).

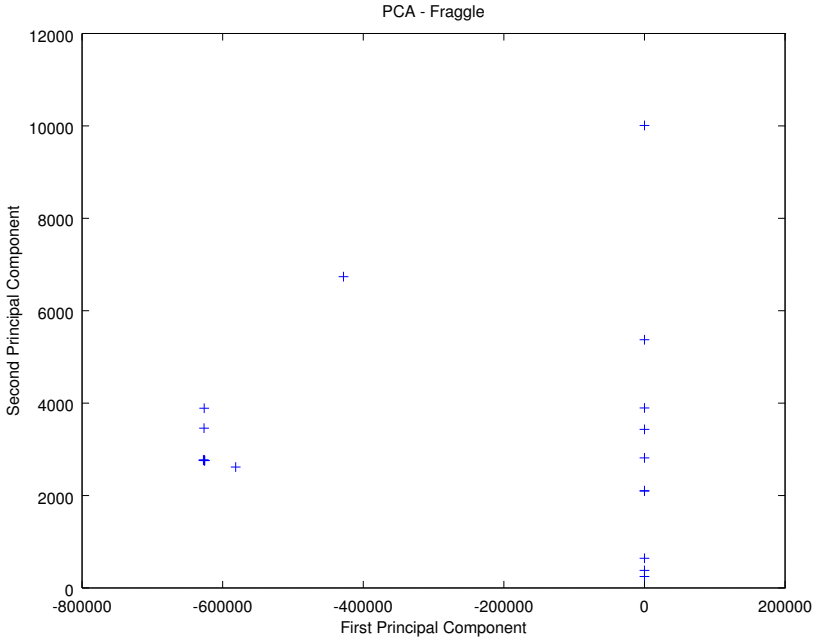

Figure 13 Fraggle Components. The two first principal components (fraggle).

7.5 Model Order Selection Schemes

It is relevant to apply MOS schemes to automate the process, taking into account the profile of the evaluated eigenvalues.

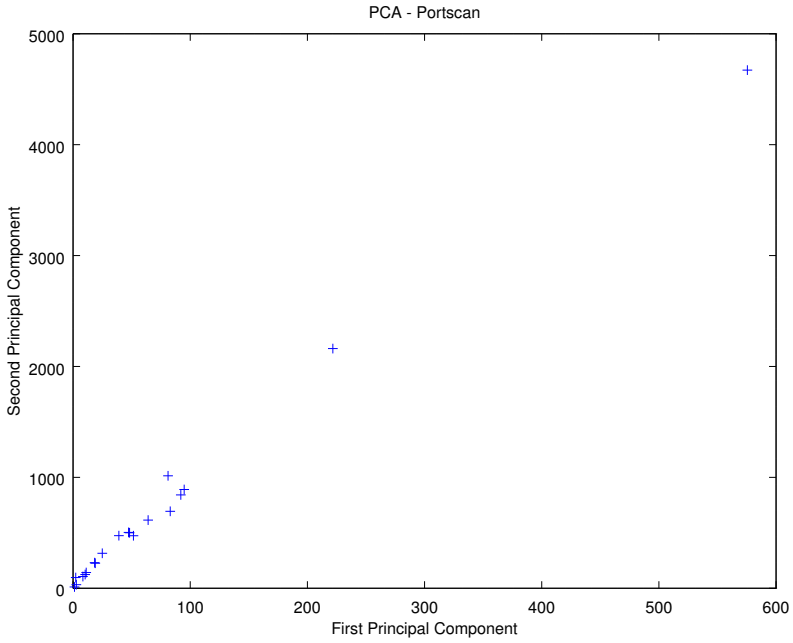

Figure 14 Port scan Components. The two first principal components (port scan).

According to Figure **??**, once obtained the GETV vector, it is possible to apply the MOS schemes to estimate the model order. Table 4 presents the results obtained from the use of the following MOS schemes: AIC, MDL, EDC, RADOI, EFT and SURE.

Table 4 MOS schemes applied to the GETV

Type of analysis q	MOS schemes (estimated model order $\hat{d}$)						Real value (d)
	AIC	MDL	EDC	RADOI	EFT	SURE	
Detection of synflood (presence of attack)	2	1	**1**	5	**1**	4	**1**
Detection of synflood (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of fraggle (presence of attack)	1	1	**1**	5	**1**	4	**1**
Detection of fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of port scan (presence of attack)	1	1	**1**	1	**1**	9	**1**
Detection of port scan (absence of attack)	0	0	**0**	1	**0**	1	**0**
Detection of synflood/fraggle (presence of attack)	2	2	**2**	5	**2**	5	**2**
Detection of synflood/fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**

With the results shown, it is possible to observe that two schemes stand out from the others. Efficient Detection Criterion (EDC) and Exponential Fitting Test (EFT) are the most effective schemes, returning values greater than or equal value to 1

(one), and indicating that there was an attack. Values equal to 0 (zero) indicating the absence of attacks.

The AIC and MDL schemes are satisfactory only for detecting the port scan. The SURE and RADOI schemes did not show effective results for either case.

It was expected the value 1 as the model order value when there was an attack. Due to the presence of the principal component, the component that stands out from the others is used as the reference to detect attacks.

Values greater than 1, returned by the scheme, indicates that there was more than one attack. An example of this could be seen when the eigenvalues related to the synflood and fraggle attacks are grouped in a same GETV vector, showing the presence of the two attacks, as indicated in the second column of Table 3. This vector carries information from two denial of service attacks. The EDC and EFT schemes returned values equal to 2, indicating the presence of two attacks. According to the modelling adopted, this problem can be interpreted as a single denial of service attack that spanned over a period of time.

8 Conclusion and Future Works

This paper proposed the Greatest Eigenvalue Time Vector Approach (GETV) approach for detecting port scan, synflood and fraggle attacks, showing that GETV can be applied to attacks involving port scanning, and denial of service. For these types of attack, the technique proved to be quite effective.

In order to make automated detection, schemes for selecting the model order were evaluated. Through some experiments, it was concluded that the GETV combined with EFT and EDC schemes presented favorable results for the analyzed problem. Moreover, our scheme is blind, which means that no training is required.

As a future work, GETV technique can be tested in other layers of the OSI network model, since this work only evaluated protocols of the transport layer. Furthermore, the GETV can be combined with other techniques, such as data mining and regular files analysis, to detect attacks that slightly escape from the behavior shown in this work. We highlight that the GETV technique can be also applied to scientific areas, since it is a general concept about eigenvalues variation.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
The authors thank the Brazilian Ministry of Planning, Budget and Management for the support during the development of this work.

Author details
[1]Department of Electrical Engineering, University of Brasilia (UnB), , 70910-900 Brasília-DF, Brazil. [2] Electrical Engineering Department, Federal University of Rio Grande do Sul (UFRGS), , 98400-000 Frederico Westphalen-RS, Brazil.

References

1. GUES, P., NAKAMURA, E.: Segurança de redes em ambientes cooperativos. BERKELEY BRASIL (2002)
2. Mudzingwa, D., Agrawal, R.: A study of methodologies used in intrusion detection and prevention systems (idps). In: Southeastcon, 2012 Proceedings of IEEE, pp. 1–6 (2012). IEEE
3. David, B.M., da Costa, J., Nascimento, A.C., Amaral, D., Holtz, M., de Sousa Jr, R.: Blind automatic malicious activity detection in honeypot data. In: The International Conference on Forensic Computer Science (ICoFCS) (2011)
4. da Costa, J., de Freitas, E.P., David, B.M., Serrano, A.R., Amaral, D., Júnior, R.S.: Improved blind automatic malicious activity detection in honeypot data. In: The International Conference on Forensic Computer Science (ICoFCS) (2012)

5. He, W., Hu, G., Yao, X., Kan, G., Wang, H., Xiang, H.: Applying multiple time series data mining to large-scale network traffic analysis. In: 2008 IEEE Conference on Cybernetics and Intelligent Systems, pp. 394–399 (2008)
6. Ghourabi, A., Abbes, T., Bouhoula, A.: Data analyzer based on data mining for honeypot router. In: Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference On, pp. 1–6 (2010). IEEE
7. Raynal, F., Berthier, Y., Biondi, P., Kaminsky, D.: Honeypot forensics. In: Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, pp. 22–29 (2004). IEEE
8. Almotairi, S., Clark, A., Mohay, G., Zimmermann, J.: A technique for detecting new attacks in low-interaction honeypot traffic. In: Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference On, pp. 7–13 (2009). IEEE
9. Zakaria, W.Z.A., Kiah, M.L.M.: A review on artificial intelligence techniques for developing intelligent honeypot. In: Proceeding Of: 8th International Conference on Computing Technology and Information Management, At Seoul, Korea (2012)
10. da Costa, J.P.C., Haardt, M., Romer, F., Del Galdo, G.: Enhanced model order estimation using higher-order arrays. In: Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference On, pp. 412–416 (2007). IEEE
11. Jolliffe, I.: Principal Component Analysis. Wiley Online Library, ??? (2005)
12. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.-i.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. John Wiley & Sons, ??? (2009)
13. Da Costa, J., Thakre, A., Roemer, F., Haardt, M.: Comparison of model order selection techniques for high-resolution parameter estimation algorithms. In: Proc. 54th International Scientific Colloquium (IWK'09), Ilmenau, Germany (2009)
14. Rajan, J., Rayner, P.: Model order selection for the singular value decomposition and the discrete karhunen–loeve transform using a bayesian approach. IEE Proceedings-Vision, Image and Signal Processing **144**(2), 116–123 (1997)
15. Akaike, H.: A new look at the statistical model identification. Automatic Control, IEEE Transactions on **19**(6), 716–723 (1974)
16. Wax, M., Kailath, T.: Detection of signals by information theoretic criteria. Acoustics, Speech and Signal Processing, IEEE Transactions on **33**(2), 387–392 (1985)
17. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. Information Theory, IEEE Transactions on **44**(6), 2743–2760 (1998)
18. Zhao, L., Krishnaiah, P., Bai, Z.: On detection of the number of signals in presence of white noise. Journal of Multivariate Analysis **20**(1), 1–25 (1986)
19. Ulfarsson, M.O., Solo, V.: Rank selection in noist pca with sure and random matrix theory. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference On, pp. 3317–3320 (2008). IEEE
20. Radoi, E., Quinquis, A.: A new method for estimating the number of harmonic components in noise with application in high resolution radar. EURASIP Journal on Applied Signal Processing **2004**, 1177–1188 (2004)
21. Grouffaud, J., Larzabal, P., Clergeot, H.: Some properties of ordered eigenvalues of a wishart matrix: application in detection test and model order selection. In: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference On, vol. 5, pp. 2463–2466 (1996). IEEE
22. Quinlan, A., Barbot, J.-P., Larzabal, P., Haardt, M.: Model order selection for short data: An exponential fitting test (eft). EURASIP Journal on Advances in Signal Processing **2007** (2006)
23. Tenório, D.F., da Costa, J.P.C., de Souza Júnior, R.T.: Greatest eigenvalue time vector approach for blind detection of malicious traffic. In: The International Conference on Forensic Computer Science (ICoFCS) (2013)
24. Jin, S., Yeung, D.S.: A covariance analysis model for ddos attack detection. In: Communications, 2004 IEEE International Conference On, vol. 4, pp. 1882–1886 (2004). IEEE
25. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Computer Communication Review, vol. 35, pp. 217–228 (2005). ACM