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 protect network devices, including end-user computers, and to allow quick decisions to be taken regarding the implementation of safety countermeasures.

This work proposes an innovative 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 impossible to analyze them separately. Then, by considering this model, we propose the Greatest Eigenvalue Time Vector (GETV) approach which successfully detects malicious traffic. Since our scheme is blind, no training is necessary. Moreover, no human intervention is also required. We validate our proposed approach by detecting denial of service (synflood and fraggle) and scan of communication ports (portscan) attacks using a real computer network.

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

1 Intrduction

The need for 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 mean 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 defense arsenal used by an organization can work against certain types of attacks, but perhaps fail against new malicious techniques developed [1].

In this context, a major challenge in a communication network is the guarantee of security related to the data integrity, availability, and confidentiality. There are several ways to provide security, such as taking into account both technical aspects using equipment or security systems, and establishing security policies and staff awareness campaigns. Firewalls, intrusion detection systems and intrusion preven-

tion systems are examples of equipment or security systems that can be employed (CERT.br, 2010).

Firewalls act as the first defense line for protecting servers and network resources from unauthorized access and malicious traffic. Firewalls are typically deployed at the network edge or at the entry point of a private network. Network firewalls inspect the incoming and outgoing traffic of the Internet. Firewalls can allow or block incoming or outgoing traffic based on a set of defined rules. Thereby, network firewalls work based on rules that sequentially interrogate the packages, rule by rule, until a match is found and this match is dropped or released to proceed to destination [2].

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 [3].

In this work, we propose an automatic blind malicious traffic detection technique, for using on any computer in a network. In [4, 5] the real network traffic data was modeled into three components: legitimate traffic, malicious traffic and noise.

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.

Inspired by [4, 5], this work models the network traffic 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). Therefore, the modeling concerns only on the transport layer, which is the layer selected as scope of attack detection.

Our proposed technique is based on the eigenvalue decomposition, however, in contrast to [4, 5], 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 based on the greatest eigenvalue variation, that attacks such as synflood, fraggle and portscan 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 attacks such as synflood, fraggle and portscan.

This paper is organized as follows. In Section 2, related works are discussed. In Section 3, the mathematical notation used in the following sections is presented. Section 4 presents 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 characterizes synflood, fraggle and portscan attacks, as well as the data collection, data modeling and attack detection. Section 6 describes the experimental validation, which uses real data, also describes the evaluation of several MOS schemes, presents the corresponding experimental results and validate the proposed approach. In Section 7 final remarks are made and future works are suggested.

2 Related Works

Several methods have been proposed for the identification and characterization of malicious activity in computer networks. Classical methods typically employ data mining [6, 7] and regular file analysis [8] to detect patterns that indicate the presence of specific attacks in traffic analysis.

Multiple series of data mining are used in [6] to analyze data flow in a network with the aim of identifying characteristics of malicious traffic in large scale environments. Data mining is often used to describe the process of extracting useful information from large databases. Researchers have applied data mining techniques in log analysis [7] to improve the intrusion detection performance. However, the requirement of the prior collection of large data sets is a weakness of this process.

Regular file analysis [8] consists in detecting patterns that indicate the presence of specific attacks through traffic analysis, and applying the statistical study of traffic data collected. An essential feature of this method is that it depends on prior knowledge of the attacks to be identified, and also depends on the previous collection of logs for applying traffic analysis and reducing false positives.

The PCA technique can be used in attack detection [9], however, if used PCA without combination with any other technique, such as Model Order Selection (MOS), it is necessary the subjective character of human intervention, making it impractical for automatic systems and being prone to errors, such as false positive.

Blind automatic detection of malicious traffic techniques has been developed to honeypots in [4, 5]. However, traffic on honeypot is simpler, because there are no legitimate applications running. A honeypot emulates behavior of a host within a network to deceive and lure attackers [10].

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 [4].

The use of Model Order Selection for blind detection in network traffic, to identify malicious activities in honeypots, was proposed by [4]. Criteria for Selecting Model Order are usually evaluated through simulations and comparing the order of the resulting model with the true model order [11].

Our approach unless a more complex traffic, which is composed of legitimate, noise and attack signals. In contrast with [6–8], our approach does not require either significant amount of logs to detect attacks, nor the prior data collection, to make comparisons and evaluate the existence of malicious traffic. Moreover, in contrast with [9], the attack detection is automatic and require no human intervention.

In [12] it was proposed an alternative technique to the one presented in this paper, through a general explanation of an intrusion detection system based on traffic anomalies detection, presenting a discussion of false positives and false negatives.

The 1 shows a comparison between the related works and the approach proposed in this paper [13], showing the proposed new technique (GETV), which is presented throughout this work.

3 Mathematical Notation

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

4 Mathematical Concepts

We present the following mathematical concepts used in this study to detect attacks: eigenvalues and eigenvectors, correlation and covariance data, Principal Components Analysis (PCA) and Model Order Selection (MOS).

4.1 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors, commonly used in linear algebra, can reveal important information about matrix data structure. In the context of malicious traffic detection, matrices can be used to represent the amount of traffic associated with each communication port at a given time, for example.

Table 1 Comparison between malicious traffic detection schemes of related works and of this paper

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

Complex systems can be represented by matrices, however in certain cases are sometimes difficult for compute processing, requiring a large computational effort and great amount of memory.

Let a square matrix $\mathbf{G}$, with real-valued elements, be decomposable into two matrices $\mathbf{F}$ and $\mathbf{G}$,

$$\boldsymbol{G = FBF^T}, \tag{1}$$

where $\mathbf{B}$ is a diagonal matrix similar to the matrix $\mathbf{G}$, and formed by the eigenvalues of the matrix $\mathbf{G}$. The matrix $\mathbf{F}$ that diagonalizes the matrix $\mathbf{G}$ is formed by the eigenvectors of the matrix $\mathbf{G}$, where each column of the matrix $\mathbf{F}$ is formed by an eigenvector of the matrix $\mathbf{G}$, according to the expression below:

$$\boldsymbol{F = [v_1|v_2|v_3|...|v_n]}, \tag{2}$$

where $\boldsymbol{v_1, v_2, v_3 ... v_n}$ represents the eigenvectors corresponding to the matrix $\mathbf{G}$.

Note that the eigenvalues provides information about rank of the matrix $\mathbf{G}$ and that the vectors of the matrix $\mathbf{F}$ must necessarily be linearly independent.

Being $\boldsymbol{G^TG} \in \mathbb{R}^{nxn}$ and $\boldsymbol{GG^T} \in \mathbb{R}^{mxm}$ symmetric matrices, the eigenvectors of $\boldsymbol{G^TG}$ are orthogonal to each other, as well as the eigenvectors of $\boldsymbol{GG^T}$ are orthogonal to each other.

4.2 Correlation and Covariance Data

To apply the mathematical concepts used in this work, such as PCA and MOS, it is necessary to previously structure the collected data into matrices, and subsequently to calculate its correlation and covariance matrices.

Let $\mathbf{X}$ be a matrix of sample data consisting of p variables, observed n times simultaneously. Thus, it can be represented as follows:

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{p,1} & \cdots & x_{p,n} \end{bmatrix}_{pxn}, \tag{3}$$

The definition of sample covariance matrix is:

$$\boldsymbol{S}_{xx} = \begin{bmatrix} s_{1,1} & \cdots & s_{1,p} \\ \vdots & \ddots & \vdots \\ s_{p,1} & \cdots & s_{p,p} \end{bmatrix}_{pxp}, \tag{4}$$

where $s_{i,k}$ represents the sample covariance between two variables.

The definition of sample correlation matrix is:

$$\boldsymbol{R}_{xx} = \begin{bmatrix} \frac{s_{1,1}}{\sqrt{s_{1,1}}\sqrt{s_{1,1}}} & \cdots & \frac{s_{1,p}}{\sqrt{s_{1,1}}\sqrt{s_{p,p}}} \\ \vdots & \ddots & \vdots \\ \frac{s_{p,1}}{\sqrt{s_{p,p}}\sqrt{s_{1,1}}} & \cdots & \frac{s_{p,p}}{\sqrt{s_{p,p}}\sqrt{s_{p,p}}} \end{bmatrix}_{pxp}, \tag{5}$$

Depending on the problem under concern, the sample covariance matrix or the sample correlation matrix can be used.

4.3 Principal Components Analysis (PCA)

Principal Components Analysis is a multivariate analysis technique that has been widely used in different research areas, such as: Internet traffic analysis, economy, image processing, and genetics. PCA is mainly used to reduce the data set size, using uncorrelated variables for this, called the principal components (PC). This transformation into another variable set occurs with the least possible information loss or even variables that contain only noise (this process is called denoising) [14].

The principal components generated are a linear combination of the original variables, and are orthogonal and ordered so that the first principal component has the greatest variance of the original data. Although the resulting number of principal components is equal to the original number of variables, most of the variation in the original set can be retained by the first principal component, thereby reducing the size of the problem [15].

4.4 Model Order Selection (MOS)

In many digital signal processing applications, including radar, sonar, communications, channel modeling, medical imaging, among others, the selection of the model order is a key point. It allows separating, for example, noise components of the main

components, applying reduced data set analyzed. Moreover, for many parameter estimation techniques the model order is crucial [16], 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 [17]. Therefore, given some data, it is chosen a model where it is believed to be the best to describe the dataset in question.

The state of the art regarding to the estimation techniques of model order based on eigenvalues includes is: Akaike's Information Theoretic Criterion - AIC [18, 19]; Minimum Description Length - MDL [19, 20]; Efficient Detection Criterion - EDC [21]; Stein's Unbiased Risk Estimator - SURE [22]; RADOI [23] and Exponential Fitting Test - EFT [4, 24, 25].

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 of (4) or (5), where k is a candidate value for the model order d [16].

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 [16]:

$$\hat{d} = arg\ min_k\ J(k), \tag{6}$$

where

$$J(k) = -N(\alpha - k)\ log\ (g(k)/a(k)) + p(k, N, \alpha), \tag{7}$$

where $\hat{d}$ is an estimate d of the model order, N is the number of samples, $\alpha = M$, the number of variables of the problem, $0 \leq k \leq \min[M, N]$ and penalty functions for AIC, MDL and EDC are given by the 2.

Table 2 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 be effectively used in cases where the number of samples N is small. This technique is based on observations contaminated only with white noise, the profile of eigenvalues can be approximated by a decaying exponential [24].

Given λ_i be the i-th eigenvalue of (4) or (5), the exponential model can be expressed by:

$$E\{\lambda_i\} = E\{\lambda_1\} \cdot q(\alpha, \beta)^{i-1}, \tag{8}$$

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

as:

$$q(\alpha,\beta)=\exp\left\{-\sqrt{\frac{30}{\alpha^2+2}-\sqrt{\frac{900}{(\alpha^2+2)^2}-\frac{720\alpha}{\beta(\alpha^4+\alpha^2-2)}}}\right\}, \tag{9}$$

so that: $0<q(\alpha,\beta)<1$. According to [25] if $M\leq N$, then $\beta=N$.

The 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 [11].

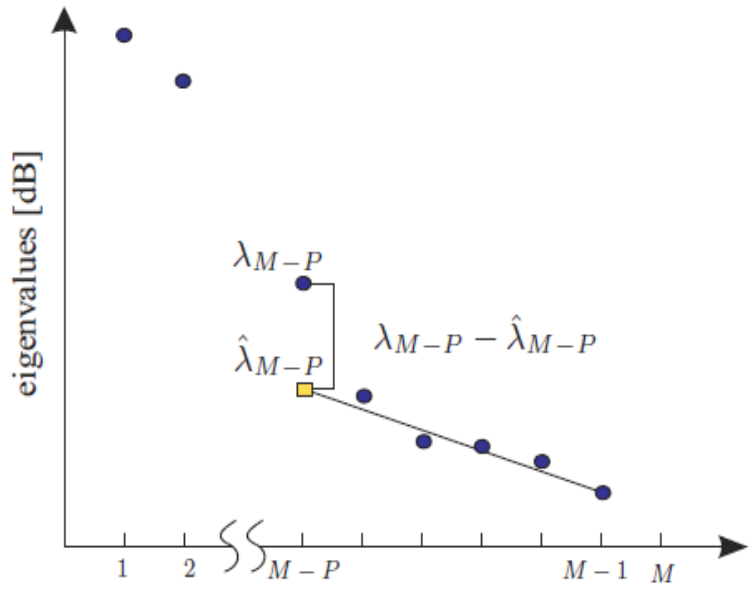

Figure 1 EFT Example. Example of application of EFT [11].

5 Proposed Solution

In this section, we propose the GETV technique to detect the synflood, fraggle and portscan attacks in any computer.

5.1 Data Collection

The log information 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, depending on 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:

$$\boldsymbol{X}^{(q)} = \boldsymbol{S}^{(q)} + \boldsymbol{N}^{(q)} + \boldsymbol{A}^{(q)}, \tag{10}$$

where q represents the q-th period of time.

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

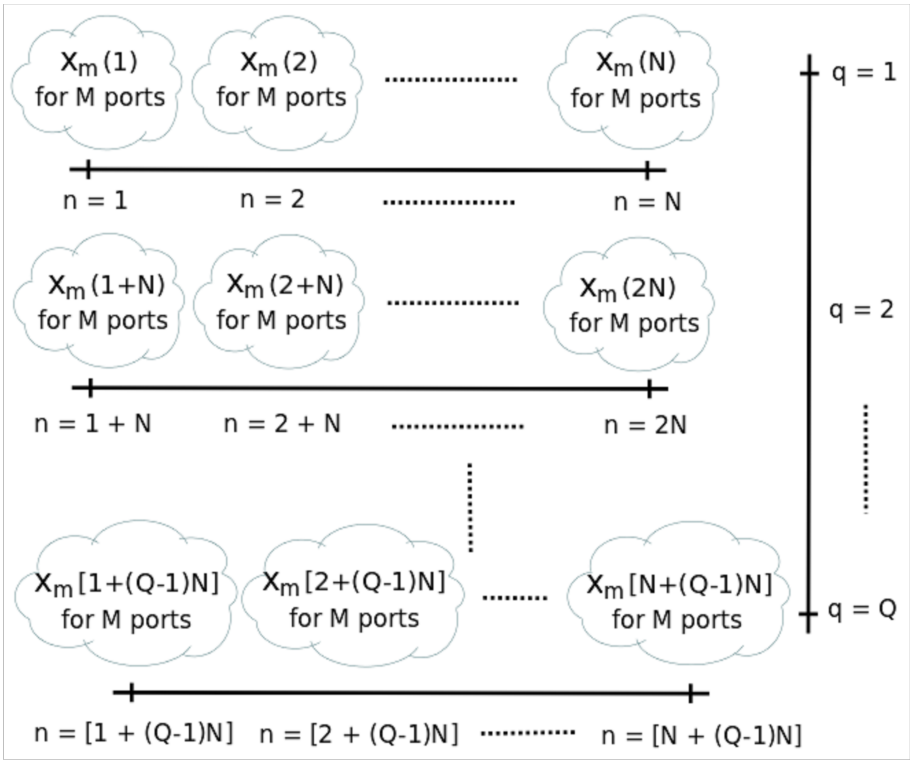

Figure 2 Traffic Matrix. Obtaining the traffic matrix $\boldsymbol{X}^{(q)}$.

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). The 3 presents the legitimate traffic obtained during experiments.

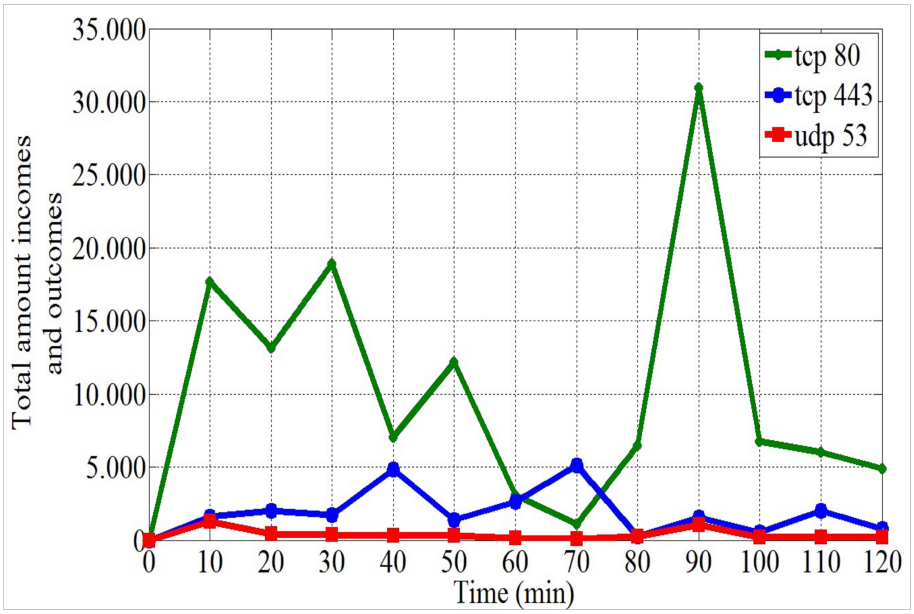

Figure 3 Legitimate Traffic. The legitimate traffic.

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 acquisition. 4 shows the noise during simulations.

The traffic coming from a malicious activity is represented by the matrix $\boldsymbol{A}^{(q)}$. This work only considers the traffic from port scanning and flood attacks, which aims to cause denial of service. If the 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 $\boldsymbol{X}^{(q)}$.

5.3 Synflood, Fraggle and Portscan

The attacks focused by this work are: synflood, fraggle and portscan. The first two attacks are 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 is set up between two computers when it is used. This virtual connection requires a "handshake" and occurs in three ways. If a computer needs to communicate with another computer, the requester sends a packet communication synchronization (SYN) to a specific port on the destination, which is in a listening state. If the destination is active, running and accepting requests, it responds to the requester

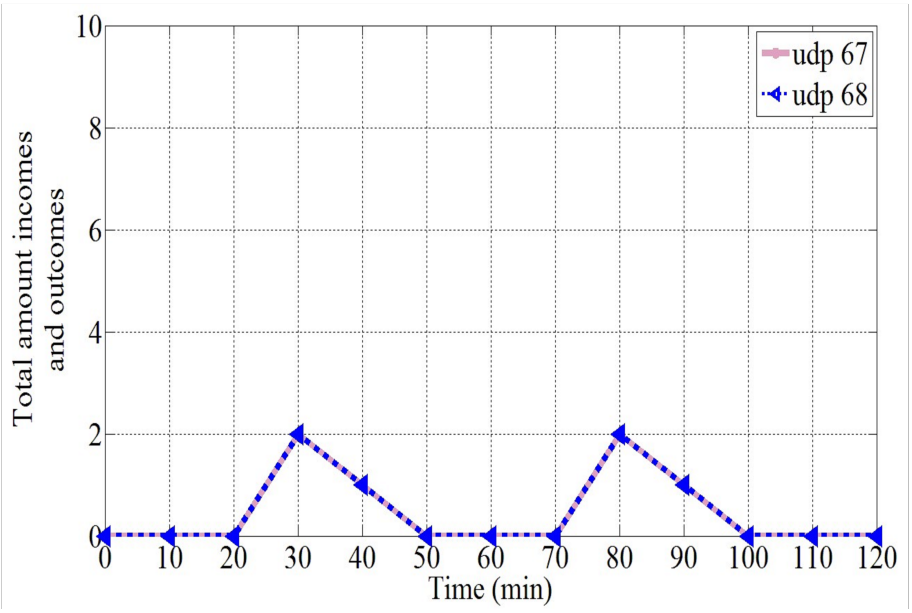

Figure 4 Noise Traffic. The noise traffic.

with a confirmation message SYN/ACK. After receiving this message, the requester sends an ACK message to the destination and the connection is established.

The 5 represents the synflood attack, which was carried out during the simulations. In a time interval of ten minutes there were more than 210,000 packets related to the attack, unusual data traffic on a network, especially because it is concentrated in a short period of time.

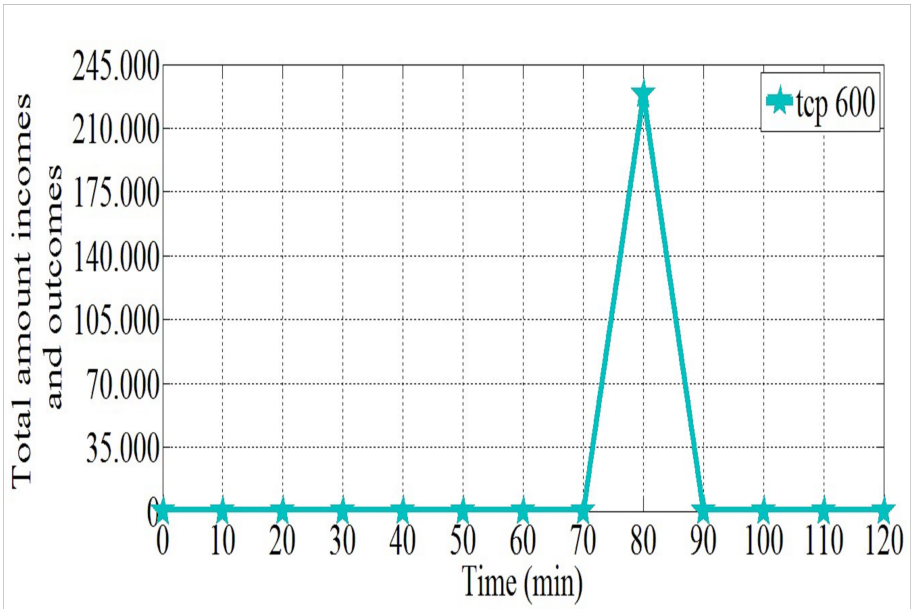

Figure 5 Synflood Traffic. The traffic characterized by synflood.

In the fraggle attack, large packet traffic with ”UDP echo” segments is sent to the IP broadcast address of the network, with the source address of the victim (IP spoofing). With the broadcast, each host receives a huge amount of requests ”UDP echo” and all of them replies to the IP address of the victim. 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 fake the 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 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 is not able to know if the segment sent has reached its destination, , i.e. as UDP is connectionless, no confirmation is sent back.

The 6 depicts the fraggle attack, which was 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.

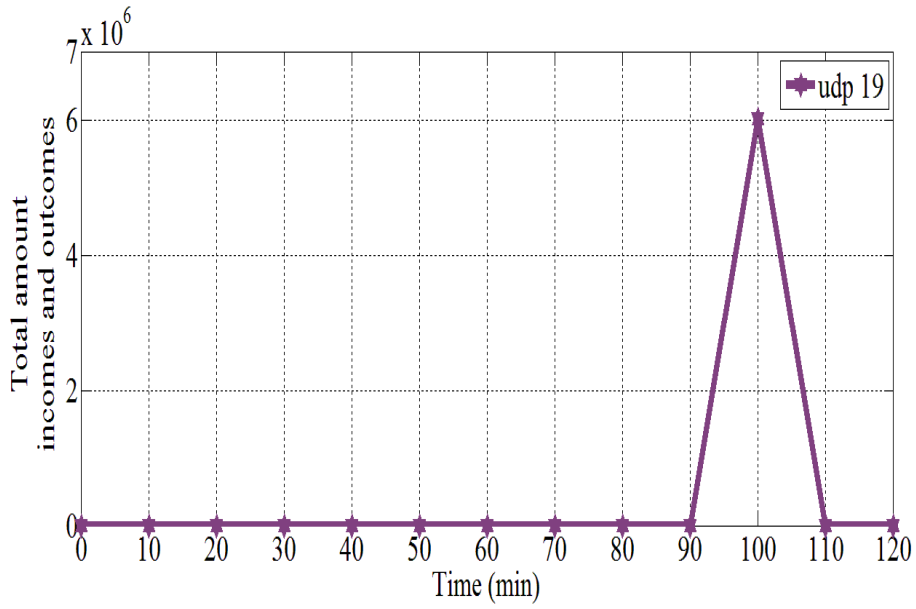

Figure 6 Fraggle Traffic. The traffic characterized by fraggle.

Portscan 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. In this type of scan, a RST/ACK packet is sent by the system that is performing the portscan, at the end of each port scanning. Thus, a full connection is never established. This makes the origin of the attack be more difficult to be detected, 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”, the port is closed. If a message is not received, then the port is open. UDP is known as a connectionless protocol and the efficacy of this technique is dependent on many factors related to network and system resources. This type of scanning is also very slow and can produce uncertain results.

The 7 depicts the portscan attack that was experimented. It is possible to observe that it is composed of two packets for each TCP port and a UDP packet to each port. These practical results are perfectly in line with what was explained about this attack. Important to note the high correlation of TCP and UDP traffic, separately, since the traffic related to the TCP ports are equal and the traffic related to the UDP ports are also equal. The equality of the traffics mentioned here refers to the amount of incoming and outgoing packets for each port.

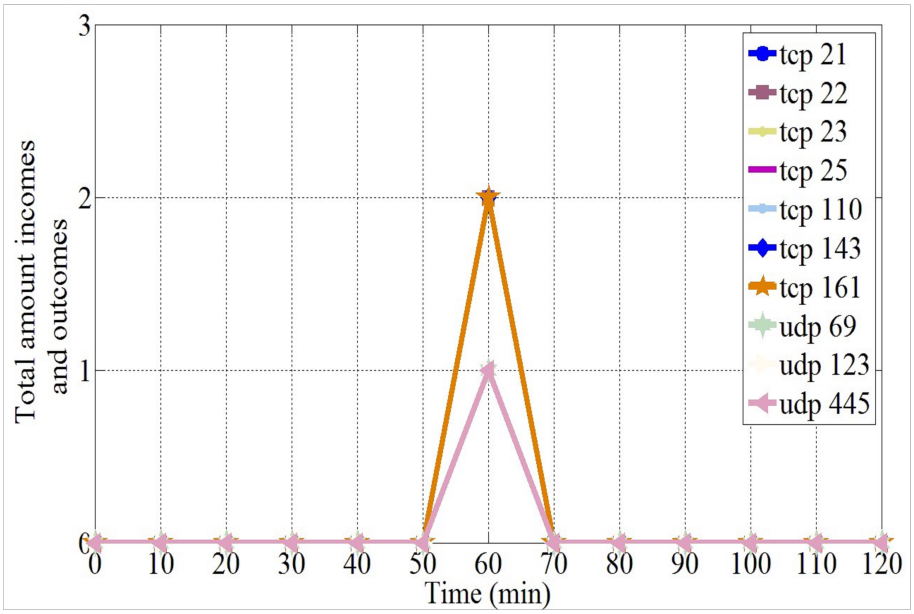

Figure 7 Portscan Traffic. The traffic characterized by portscan.

5.4 Attack Detection

The attack detection can be better understood analyzing the 8. All steps are numbered and explained below.

The attack detection process starts at $q = 1$, obtaining the matrix $\boldsymbol{X}^{(1)} \in \mathbb{R}^{MxN}$, step (A). For detection of synflood and fraggle (denial of service) attacks, it is necessary to calculate the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$, step (C), since the main components

are dominated by the variables with more variance, in accordance with discussed in Section 4.3. To obtain the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$ it is essential, for each variable (in the case of this work, each port), to calculate the deviations of the respective elements in relation to the average, step (B) of 8.

For the portscan attack detection, it is necessary to calculate the correlation matrix $\boldsymbol{R}_{xx}^{(q)}$, step (F), instead of the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$, since the main components are not dominated by the variables with large variance. The traffic associated with portscan attack does not generate many logs, as happens in denial of service, however portscan attack presents a highly correlated traffic. To obtain the correlation matrix $\boldsymbol{R}_{xx}^{(q)}$ it is essential, for each variable, to calculate the deviations of the respective elements in relation to the average divided by the standard deviation, step (E).

Once the $\boldsymbol{S}_{xx}^{(q)}$ and $\boldsymbol{R}_{xx}^{(q)}$ has been obtained, proceeds with the eigenvalue decomposition (EVD) - steps (D) and (G) respectively - to obtain the eigenvalues associated with each matrix, step (H). It is necessary to order the eigenvalues in descending order, step (I), and then select the first eigenvalue of the sequence, which is consequently the greatest one, step (J).

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, is repeated 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 time period.

Thus, we build the matrix $\boldsymbol{K} \in \mathbb{R}^{MxQ}$ formed by the eigenvalues of $\boldsymbol{S}_{xx}^{(q)}$ or $\boldsymbol{R}_{xx}^{(q)}$. Assuming $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, the first line of the matrix $\mathbf{K}$ contains the Greatest Eigenvalue Time Vector (GETV), step (K) of 8.

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

By obtaining the vector GETV, $\lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \cdots, \lambda_1^{(q)}$ it is possible to apply the schemes of MOS to estimate the model order $\hat{d}$, step (L). If the tested MOS scheme presents the estimated model ($\hat{d}$) equal to the true model order (d), it is discovered which MOS scheme applies to the problem. It is possible to find more than one scheme that applies to the problem, not necessarily only one. With this step the attack detection process ends.

6 Experimental Results

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

6.1 Analyzed Scenario

The environment studied 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 attacking while the other is the victim, according to 9.

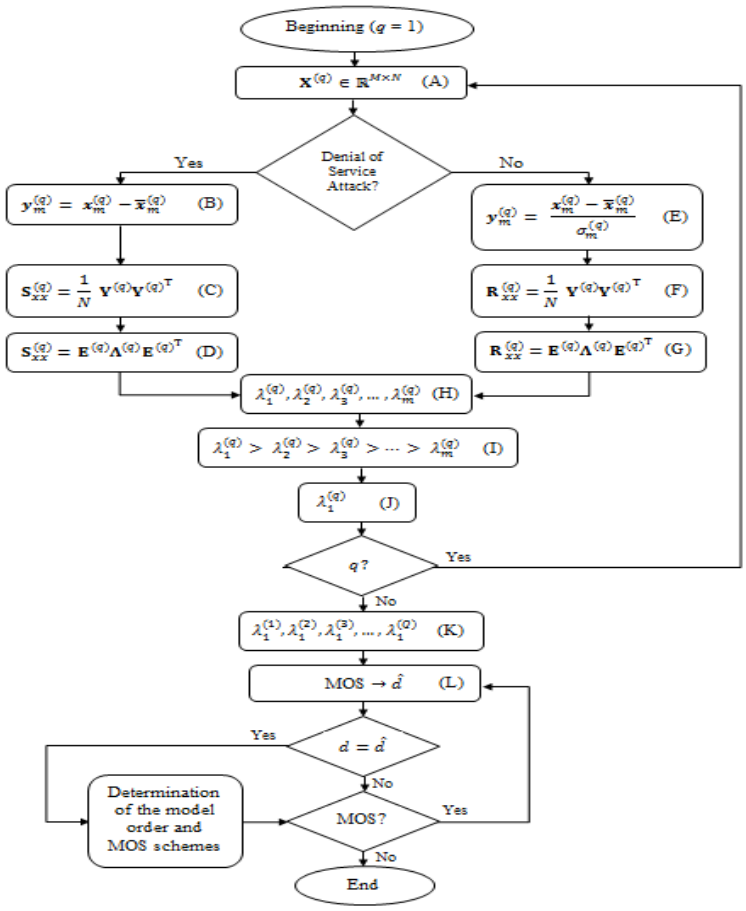

Figure 8 GETV. Proposed GETV attack detection approach.

The victim performs legitimate activities, mainly web access. In many organizations this type of access is done frequently, since most of corporate services are web-based, such as: access to the e-mail server, access to documents protocol and access to intranet 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 portscan, which was detailed in Section 5.1. The attacks were simulated using well known professionals security tools. Nmap was used to portscan, 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, each time period corresponding to twenty minutes. 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 $\boldsymbol{R}_{xx}^{(q)} \in \mathbb{R}^{17x17}$, where in the case of this paper $q = 1, 2, 3, 4, 5$ and 6. The simulation started at 21:00h, the first period is from 21:00h until 21:20h ($q = 1$), the second from 21:20h until 21:40h ($q = 2$), the third from 21:40h to 22:00h ($q = 3$), the fourth from 22:00h until 22:20h ($q = 4$),

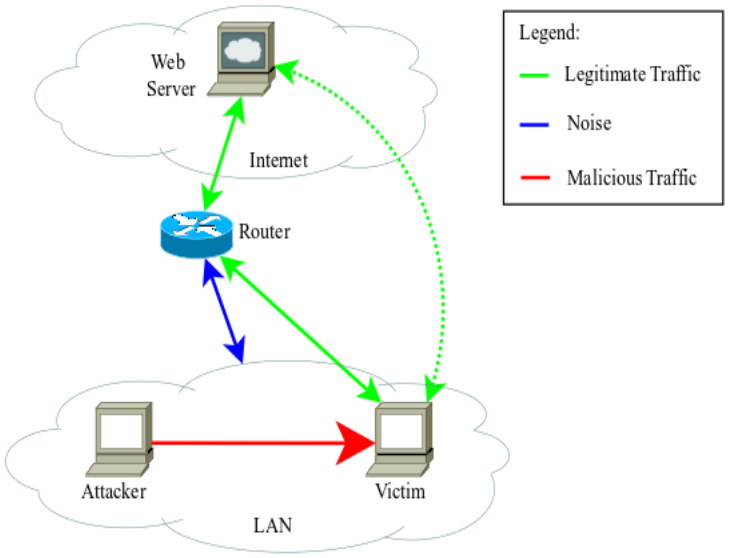

Figure 9 Scenario. Analyzed Scenario.

the fifth from 22:20h until 22:40h ($q = 5$), and finally, the sixth from 22:40h until 23.00h ($q = 6$).

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

6.2 Eigenvalues

The 10 graphically represents the eigenvalues of the matrix used for the detection of synflood. In this figure it can be seen that the greatest eigenvalue, which is related to this attack, stands out from the others.

The 11 graphically represents the eigenvalues of the matrix used for the detection of fraggle. In this figure it can be seen that the greatest eigenvalue, which is related to this attack, stands out from the others, as shown in 10 for the synflood attack.

The 12 graphically represents the eigenvalues of the matrix used for the detection of portscan. 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.

6.3 GETV

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

In 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,

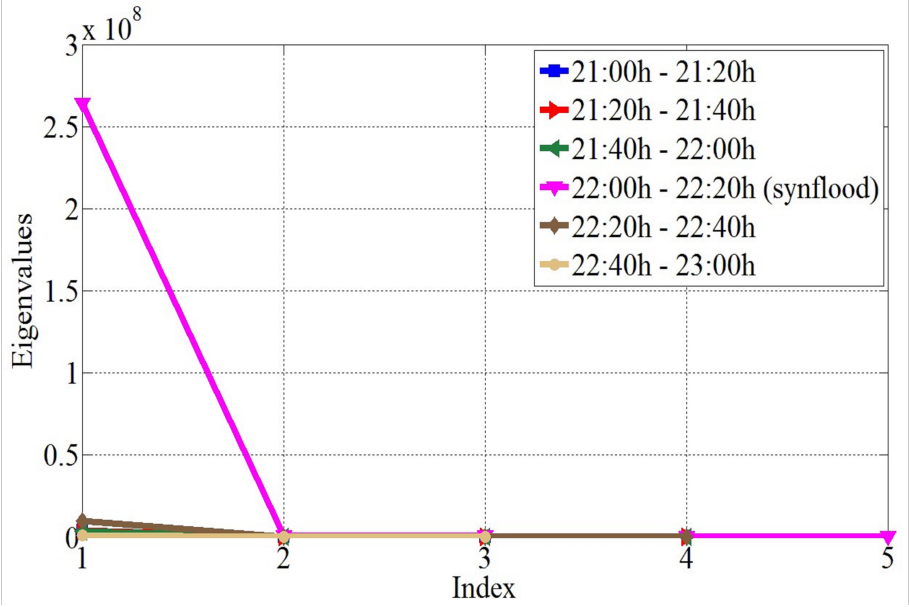

Figure 10 Synflood Eigenvalues. Eigenvalues of the covariance matrix (synflood).

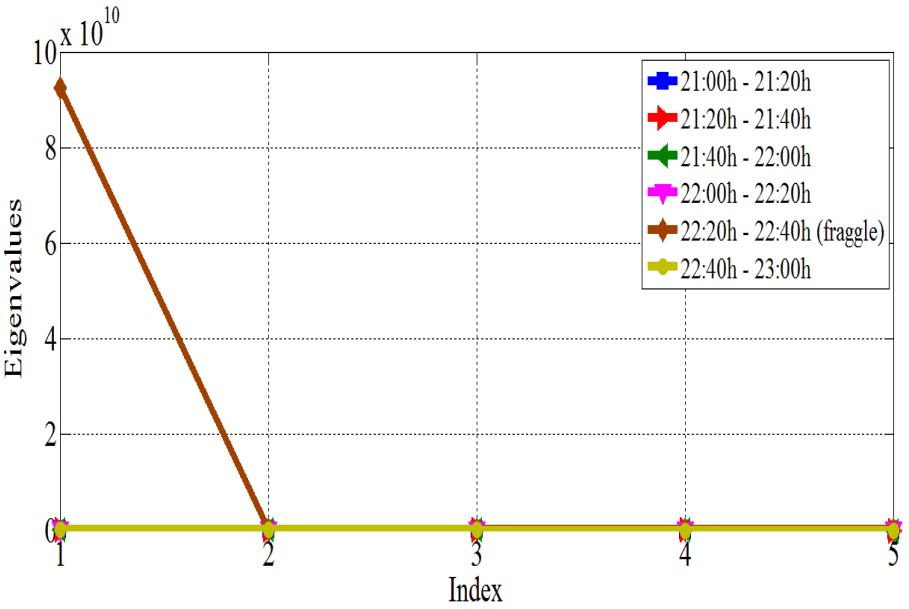

Figure 11 Fraggle Eigenvalues. Eigenvalues of the covariance matrix (fraggle).

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 portscan attack occurred, the maximum eigenvalue obtained was approximately 4 times larger than the second one. In the last case, although the greatest eigenvalue

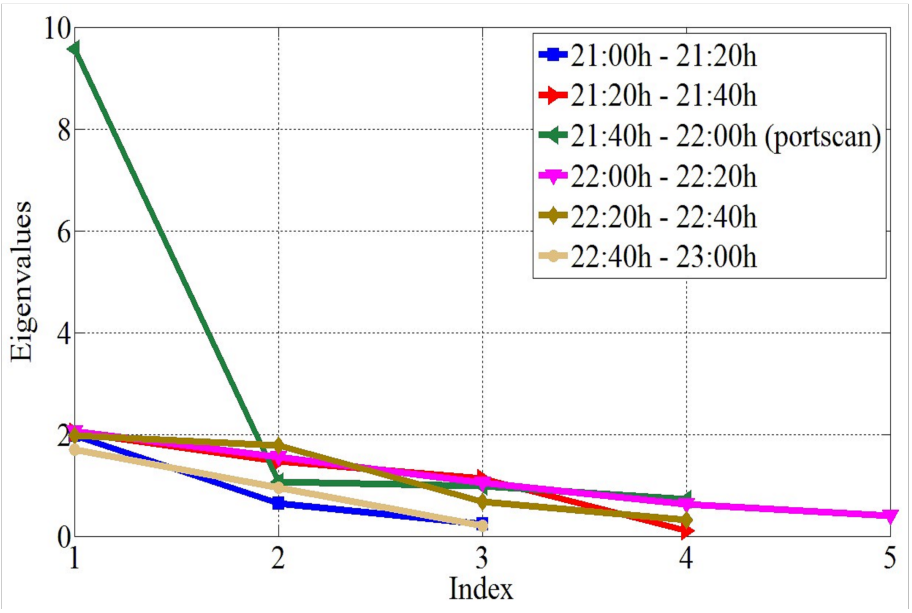

Figure 12 Portscan Eigenvalues. Eigenvalues of the covariance matrix (portscan).

was not too high, compared to synflood or fraggle attacks, it was entirely sufficient to detect the portscan, as it clearly deviates from the rest of the values.

6.4 Principal Components

As presented in Subsection 4.3, the principal component analysis is mainly used to reduce the size of a data set, 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 portscan
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 has the greatest variance, related to the original data. Although the resulting number of principal components is 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 a PC the dataset 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 dataset without loss of information.

The principal components are obtained from the eigenvectors of the covariance or correlation matrix. As it will be 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.

As the intention is to show that the attacks present a different and dominant behavior, in comparison to other traffic, it will be selected to analyse the periods related to these attacks: $q = 3$ for portscan attack, $q = 4$ for synflood attack and $q = 5$ for the fraggle attack.

The synflood attack presented in 13 shows that the variance of PC1 (first PC) is totally dominated by the attack components. Attack components are responsible for the high value of the eigenvalue associated with this principal component and consequently for the set of values of the period $q = 4$. Furthermore, as discussed in Section 4.3, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{S}_{xx}^{(4)}$.

For the fraggle attack in 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 and consequently to the set of values of the period $q = 5$. Furthermore, as discussed in Section 4.3, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{S}_{xx}^{(5)}$.

For the portscan attack in 15, 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 discussed in Section 4.3, the variance of PC1 is equals to the largest eigenvalue of the matrix $\boldsymbol{R}_{xx}^{(3)}$.

6.5 Model Order Selection Schemes

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

According to 8, once obtained the GETV vector, it is possible to apply the MOS schemes to estimate the model order. 4 presents the results obtained from the use of the following MOS schemes: AIC, MDL, EDC, RADOI, EFT and SURE.

With the results shown below, it is possible to observe that two schemes stand out from the others. The Efficient Detection Criterion (EDC) and the Exponential Fitting Test (EFT) are the most effective, returning greater than or equal value to 1 (one), indicating that there was an attack, or value equal to 0 (zero) indicating the absence of attacks.

The AIC and MDL scheme is satisfactory only for detecting the portscan. 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.

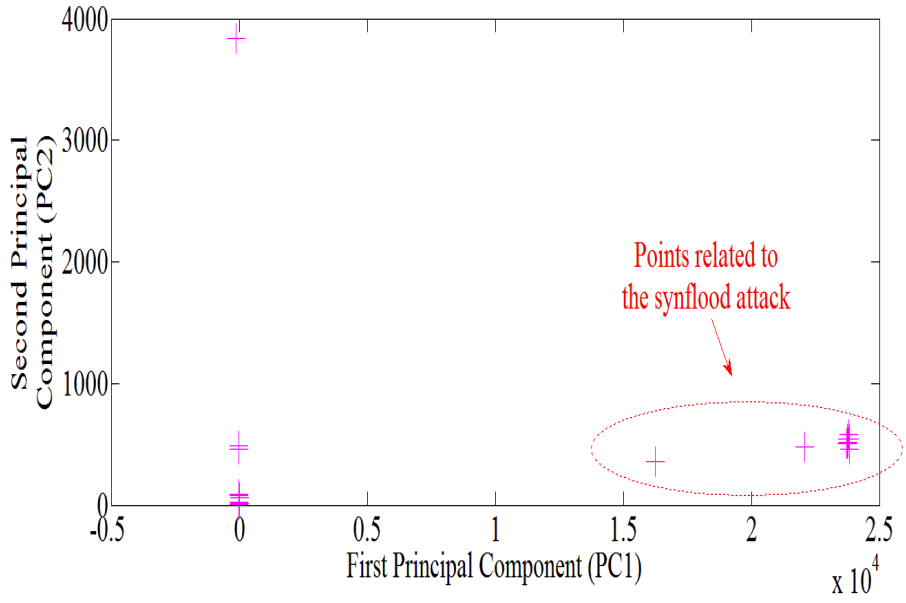

Figure 13 Synflood Components. The two first principal components (synflood).

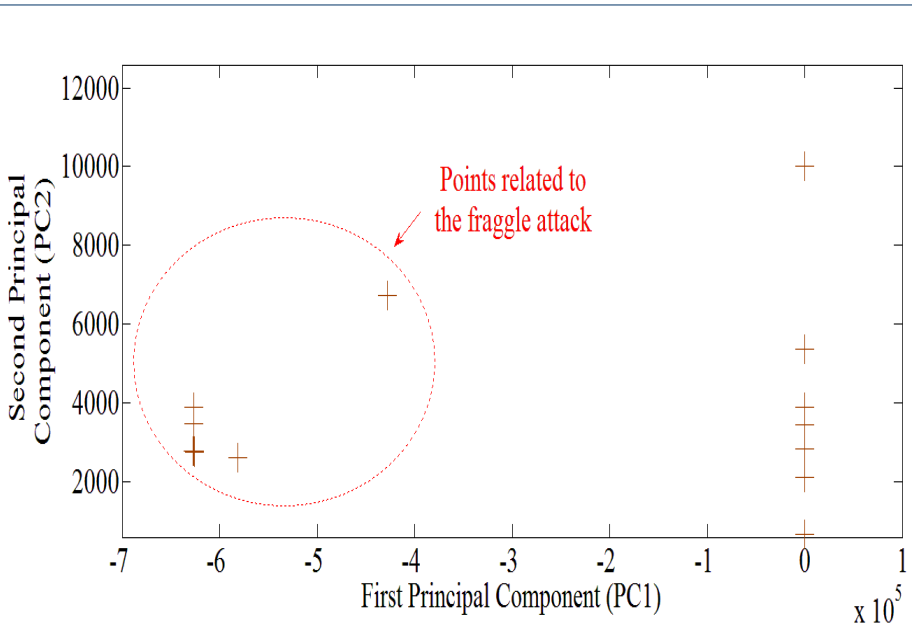

Figure 14 Fraggle Components. The two first principal components (fraggle).

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 3. This vector carries information from two denial of service attacks. The EDC and EFT schemes returned value equal to 2, indicating the presence of the two attacks. According to

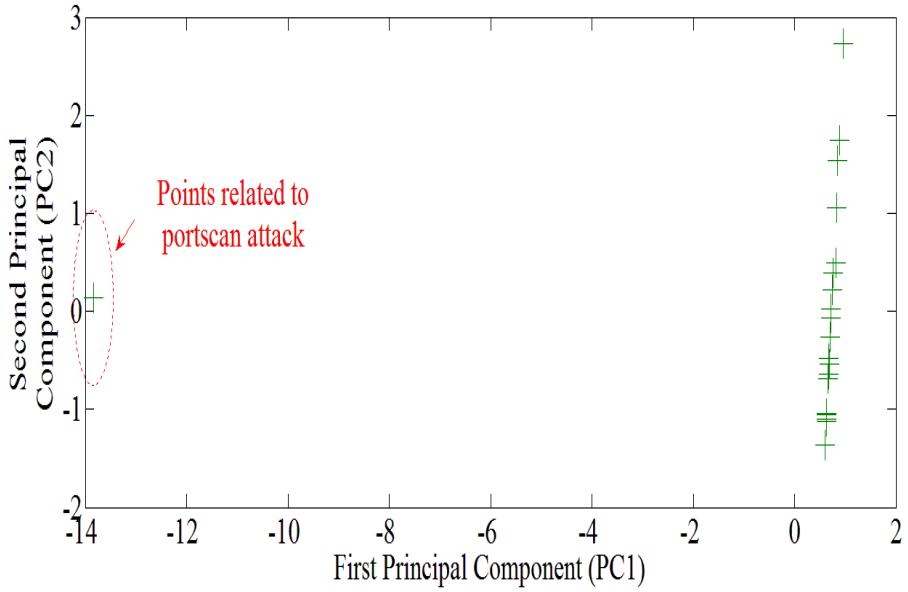

Figure 15 Portscan Components. The two first principal components (portscan).

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 portscan (presence of attack)	1	1	**1**	1	**1**	9	**1**
Detection of portscan (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**

the modelling adopted, this problem can be interpreted as a single denial of service attack that spanned over a period of time.

7 Conclusion and Future Works

This paper proposed the Greatest Eigenvalue Time Vector Approach (GETV) approach for detecting portscan, synflood and fraggle attacks. GETV can be applied to any 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, the schemes for selecting the model order were evaluated. Through some experiments it was concluded that the GETV combined with EFT and EDC schemes presented more consistent 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 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. Salah, K., Elbadawi, K., Boutaba, R.: Performance modeling and analysis of network firewalls. Network and Service Management, IEEE Transactions on **9**(1), 12–21 (2012)
3. 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
4. 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)
5. 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)
6. 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)
7. 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
8. 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
9. 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
10. 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)
11. 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
12. Puttini, R., Hanashiro, M., Miziara, F., de Sousa, R., García-Villalba, L.J., Barenco, C.J.: On the anomaly intrusion-detection in mobile ad hoc network environments. In: Personal Wireless Communications, pp. 182–193 (2006). Springer
13. Tenório, D.F., da Costa, J.P.C., de Souza Júnior, R.T.: Greatest eigenvalue time vector approach for blind detection of malicious traffic. ICoFCS 2013, 46
14. Jolliffe, I.: Principal Component Analysis. Wiley e Library, ??? (2005)
15. 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)
16. 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)
17. 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)
18. Akaike, H.: A new look at the statistical model identification. Automatic Control, IEEE Transactions on **19**(6), 716–723 (1974)
19. Wax, M., Kailath, T.: Detection of signals by information theoretic criteria. Acoustics, Speech and Signal Processing, IEEE Transactions on **33**(2), 387–392 (1985)
20. 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)

21. 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)
22. 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
23. 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)
24. 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
25. 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)