RESEARCH

Model Order Selection and Eigen Similarity Analysis for Network Attack Detection

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

Network attack detection is a research problem that requires constant inovative advances in order to deal with novel or adaptive attacks, high throughput networks, false positives, flash crowd and other characteristics that make it harder to detect and avoid malicious behavior by network traffic. Signal processing techniques have been applied to network attack detection, due to their capability to detect anomalies, high processing capacity and low dependency on previous knowledge about the attack behaviour. In this paper, we propose a signal processing technique for detailed detection of network attacks, applying model order selection (MOS), eigen analysis and similarity analysis for network attack identification. To validate our apprach, we adopted a network traffic, composed of denial of service (DoS) and probing attacks, modeled as a signal superposition of legitimate traffic, noise and malicious traffic. Evaluation results show that the proposed approach achieves acuracy on blind detection of time and port under attack.

Keywords: Network Attack Detection; Model Order Selection; Eigen Analysis; Similarity Analysis

1 Introduction

Some methods of cyber defense used by organizations can be effective against certain types of attacks, but can fail against new or modified malicious techniques [1]. In order to be able to detect and avoid well known attacks and their variations, it is necessary to develop or improve techniques to achieve efficiency on resource consumption, processing capacity and response time, it is also necessary to obtain high detections accuracy and capacity to detect variations of malicious patterns. Recently, signal processing schemes have being applied to the detection of malicious traffic in computer networks [2–7], showing advances in network traffic analysis for the detection of malicious activities.

Information security may consists of both technical and procedural aspects. The former includes equipment and security systems, while the latter corresponds to security rules and recommendations. 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, 8].

In the context of anomaly-based schemes, this work proposes a malicious traffic detection approach for computer networks. Inspired by [5, 6], 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 network ports (TCP or UDP). Our proposed technique is based on eigenvalue analysis, model order selection (MOS) and similarity analysis. In contrast with [5–7], we apply MOS and eigenvalue analysis to detect time frames under attack, we also evaluate the accuracy of our proposed novel approach, based on eigen similarity analysis, for extracting detailed information about accurate time and network ports under attack.

We show, through experiments, that synflood, fraggle and port scan attacks can be detected accurately and with great detail in an automatic and partially blind fashion, applying signal processing concepts for traffic modeling and through approaches based on MOS and eigen similarity analysis. The main contributions of this work are: the proposition of an approach based on MOS and eigen analysis to blindly detect time frames under network attack; The proposal and evaluation of the accuracy of eigen similarity analysis for detailed network attack detection.

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.

This paper is organized as follows. In Section 2, related works are discussed. Section 3 presents the data model used for evaluating our proposal. Section 4 describes our proposed approach for blind and automatic detection of malicious traffic. Section 5 discusses the experimental validation and presents the corresponding experimental results. In Section 6 are presented the final remarks and suggestions for future works. The appendices A and B present mathematical concepts and the concepts of eigenvalues and eigenvectors analysis and MOS, including the main MOS schemes and their differences.

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 [9, 10] and regular file analysis [11] to detect patterns that indicate the presence of specific attacks in network traffic.

Data mining is often used to describe the process of extracting useful information from large databases. Multiple methods of data mining are used in [9] 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 [10] to improve intrusion detection performance. However, data mining techniques requires prior collection of large data sets, which is a weakness of several schemes for online or low latency analysis [3].

Regular file analysis [11] consists of using traffic analysis for detecting known patterns that indicate the presence of specific attacks, applying statistical analysis to the study of collected traffic. An essential feature of this method is that it depends on prior knowledge of the details of the attacks to be identified, and also depends on previous log collection 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 have the largest variance. PCA can be used in attack detection [12], however, if PCA is used without the combination of any other technique for abnormalities identification, such as MOS, is necessary because of the subjectiveness of human intervention, making it prone to errors, such as false positive cases, and inefficient for automatic detection systems.

Signal processing techniques have been successfully applied to network anomaly detection, due to their ability in abnormalities detection [2]. Lu and Ghorbani [2] proposed a network anomaly detection model based on network flow, wavelet approximation and system identification theory, however, their work does not addressed problems with no significant outliers, such as port scan attacks. Zonglin *et al.* [4] proposed a method to detect traffic anomaly with correlation analysis, where traffic signals' parameters are computed to extracted their anomalous space via traffic prediction and calculate the correlation between spaces to reveal anomalies Zonglin *et al.* [4] evaluated the correlation analysis for anomaly detection, but the work was not applied to port scan and denial of service (DoS) attack detection, simultaneously.

The data collected from honeypot systems, such as captured traffic and operating system logs, can be analyzed to obtain information about attack techniques, general trends of threats and exploits. Blind automatic detection of malicious traffic techniques has been developed for honeypots in [5, 6], however, traffic on honeypot is simpler than real network traffic, because there are no legitimate applications running, due to the fact that honeypots emulate behavior of a host within a network to deceive and lure attackers [13]. Since 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 [5]. MOS for blind identification of malicious activities in honeypots was proposed by us in [5], which evaluated criterias for selecting the model order, through simulations and comparing the order of the resulting model with the true model order [14].

Lee *et al.* [15] proposed osPCA, which allows one to determine the anomaly of the target instance according to the variation of the resulting dominant eigenvector obtained by similarity analysis and over sampling. In contrast to Lee *et al.*, our approach applies MOS for detection of time frames under attack and similarity analysis to extract details for detection of time and ports under attack. Additionally, Lee *et al.* only evaluated their proposal for covariance analysis, while we adoted covariance and correlation analysis for DoS and probing attacks, respectively.

Our approach does not require either a significant amount of logs to detect attacks, nor prior data collection, in order to make comparisons and evaluate the existence of malicious traffic. Our proposed approach is automatic and blind for identification of time frames under probing and DoS attacks, through MOS and eigen analysis.

Moreover, we apply eigen similarity analysis to extract details of time and ports under network attacks.

3 Data Model

3.1 Data Collection

We used a dataset of network log to validate our approach, as explained following.

A network traffic log is commonly 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. We present the following TCP traffic log in order to exemplify the collected data:

```
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 to exemplify 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
```

However, in this paper it is only considered the following information from the networ traffic log: timestamp, port type and port number.

3.2 Modeling Data

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

$$\boldsymbol{X}^{(q)} = \boldsymbol{S}^{(q)} + \boldsymbol{N}^{(q)} + \boldsymbol{A}^{(q)}, \tag{1}$$

where q represents the q-th time frame, which is a time grouping of network traffic for splitting the analysis and computation.

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

The legitimate traffic $\boldsymbol{S}^{(q)}$ is characterized by the traffic from user's operations. When an user accesses a web page, for example, there is the corresponding TCP/IP traffic to request the page, as well as there is the traffic required to domain name resolution. Figure 1 presents the legitimate traffic obtained during experiments.

All traffic that is not directly associated with users' operations, 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 2 shows the noise during simulations.

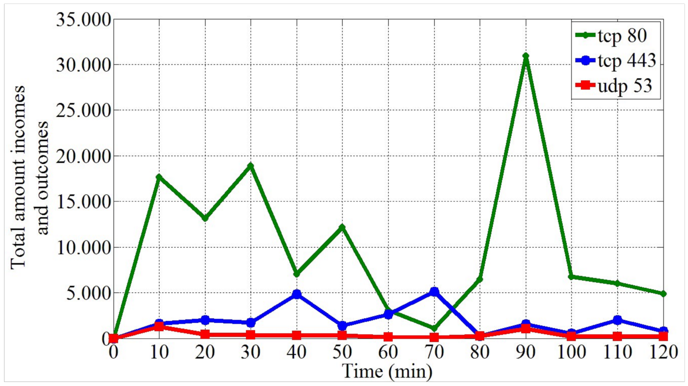

Figure 1 Legitimate Traffic. The legitimate traffic.

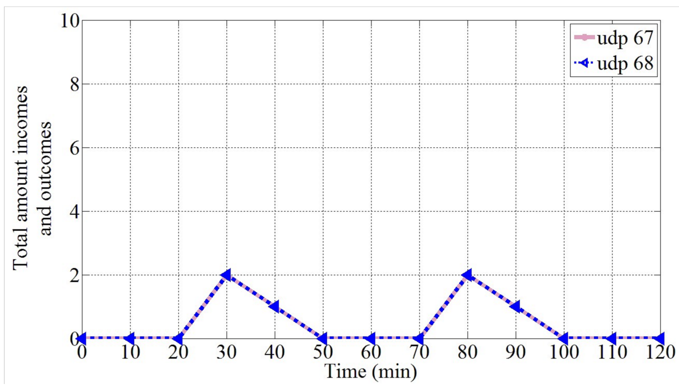

Figure 2 Noise Traffic. The noise traffic.

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 we only consider the traffic from port scanning and flood attacks, which aims to cause denial of service. We defined that: if the obtained rank $\{\boldsymbol{A}^{(q)}\} \neq 0$, then there is malicious traffic in the evaluated time frame q, 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.

3.3 Synflood, Fraggle and Port scan

The kind of network attacks evaluated by this work are: synflood, fraggle and port scan. The first two attacks can be qualified as DoS attacks, while the last one can be qualified as probing or 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 DoS. Figure 3 shows 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 a abnormal behavior of network traffic, especially because it is concentrated in a short period of time and presents similar outstanding traffic diring the time under attack.

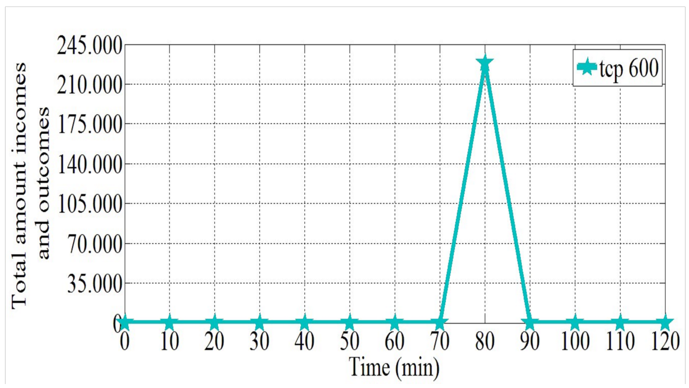

Figure 3 Synflood Traffic. The traffic characterized by synflood.

During the simulated fraggle attack, large packets with “UDP echo” segments were sent to the broadcast address of a network. Previously, every packet was modified to have the source address of the victim, in order to implement the source address spoofing technique. Therefore, each host receives a huge amount of requests “UDP echo” and all of them replies to the IP address of the victim, causing a packet flooding aiming a DoS. This attack can affect the entire network, because all hosts receive several requests “UDP echo” and respond with the ICMP protocol, therefor each host acts as an “amplifier” of the attack. 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.

Figure 4 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 abnormal network traffic, especially due to the concentrated traffic in a short period of time and due to the similarity of the outstanding traffic.

Port scan is the process of try to establish a connection to TCP and UDP ports to identify what services are running or are in the listening state. There are several available port scanning techniques, including: TCP SYN scan, TCP ACK scan and UDP scan. This work evaluates the use of TCP SYN scan and UDP scan.

In TCP SYN scan, a SYN packet is sent to the destination and two types of response may occur: SYN/ACK or RST/ACK. In the first case, the destination port is in the listening state, in the second case, the destination port is not listening. At

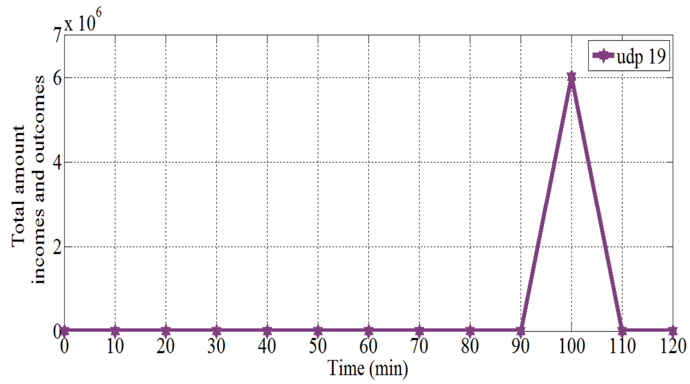

Figure 4 Fraggle Traffic. The traffic characterized by fraggle.

the end of each port scanning, a RST/ACK packet is sent by the system that is performing the port scan, thus a full connection or a complete three-way handshake is never established, which makes the detection of the attack sender more difficult, and requires approaches able to identify probing attacks without connection stablishment. The UDP scan technique sends UDP packets to the destination port, if it responds with a “ICMP port unreachable” message, it indicates that the scanned port is closed. If a message is not received, then the port is considered as open.

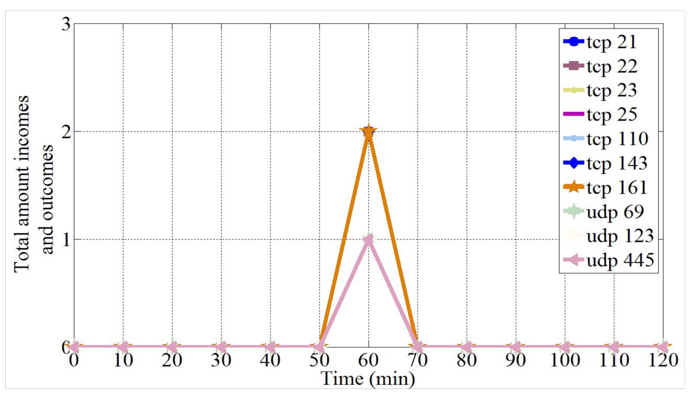

Figure 5 Port scan Traffic. The traffic characterized by port scan.

Figure 5 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 one UDP packet to each port, as explained above. The incoming and outgoing packets analysis, for each port, shows the high correlation and similarity of TCP and UDP traffic during the simulated port scan attack.

4 The proposed algorithm

In this section, we present the proposed technique for synflood, fraggle and port scan attacks detection, which requires a data set containing network traffic log, as input, to apply a pre-processing and initial specifics steps for detecting DoS or port scan attacks, and subsequently to execute common tasks for port scan and DoS detection.

The proposed attack detection algorithm starts by the data pre-processing of a network traffic log containing IP, ports and timestamp of senders and receivers. During this step the desired information is extracted in order to classify and count packets according to the origin and destination ports, and subsequently this information is grouped by minutes and Q time frames.

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

According to DoS and port scan attacks' behavior, it is possible to characterize DoS attacks as a covariance aware attack [16] and port scan attacks as a correlation aware attack [1]. These characteristics are substantiated by the results obtained through the covariation and correlation 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 it presents a highly correlated traffic.

Therefore, in our approach for detection of DoS attacks, it is necessary to calculate the covariance matrix $\boldsymbol{S}_{xx}^{(q)}$, which can be obtained from the deviations of the respective elements in relation to the average, as defined by the equation 2.

$$\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)} \tag{2}$$

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 3.

$$\boldsymbol{S}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T} \tag{3}$$

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 and because port scan attack presents a highly correlated network 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 4.

$$\boldsymbol{y}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}} \tag{4}$$

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 5.

$$\boldsymbol{R}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T} \tag{5}$$

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 6, in order to obtain the vector of eigenvalues $\boldsymbol{E}_m^{(q)}$ associated with each matrix.

$$\boldsymbol{E}_m^{(q)} = \boldsymbol{E}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{E}^{(q)^T} \tag{6}$$

The obtained vector of eigenvalues $\boldsymbol{E}_m^{(q)}$ is composed by $\lambda_m^{(1)}, \lambda_m^{(2)}, \lambda_m^{(3)}, ..., \lambda_m^{(q)}$ eigenvalues. Thes 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 largest eigenvalue of the time frame 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 7.

$$\boldsymbol{K} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}, \tag{7}$$

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 largest eigenvalue for each q-th time frame, should be repeated until $q = Q$, in order to obtain the largest eigenvalue of all time frames. Since $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, then the first line of the matrix $\mathbf{K}$ contains the largest eigenvalues of each q-th time frame, which is the Greatest Eigenvalue Time Vector (GETV) [7], denoted as the equation 8.

$$GETV = \lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \cdots, \lambda_1^{(q)} \tag{8}$$

Once obtained the largest eigenvalue of each q-th time frame, it is possible to apply a selected MOS scheme to estimate the model order $\hat{d}$, which is the estimated number of time frames under attack. Therefore, GETV is used as input parameter for MOS schemes, although some MOS schemes may also require the number of minutes that compose a time frame, to perform its calculation, as following.

$$\hat{d} = MOS(GETV), \qquad \hat{d} = MOS(GETV, Q) \tag{9}$$

On previows work [7], we evaluated the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE schemes for synflood and port scan attack detection, showing that EDC and EFT are effective for detecting this kinds of attacks. In this work we extended that evaluation to also analyse the effectiveness of the listed MOS schemes for fraggle attack detection, which results will be showed in section 5.

Although EDC and EFT presented the same accuracy on our evaluation, the EDC scheme requires less processing time than EFT, which is an important criteria to select EDC as the MOS scheme for DoS and port scan detection on our remain experiments.

The selected EDC scheme can estimate the number of time frames under attack, denoted by $\hat{d}$, but it does not provide information of which q-th time frames are under attack. However, the time frames under attack can be estimated through the largest eigenvalue analysis, which indicated that the $\hat{d}$ time frames with the $\hat{d}$ largest eigenvalues correspond to the q-ths time frames under attack during our simulation. The largest eigenvalue analysis can be expressed by equation 10.

$$\hat{Q} = \arg\max_{\hat{d}} \ GETV(\hat{d}), \tag{10}$$

where $\hat{Q}$ denotes a vector of q-th indexes corresponding to the $\hat{d}$ largest eigenvalues of GETV, which refers to the q-ths time frames under attack.

After to estimate the $\hat{Q}$ time frames under attack, it is necessary to obtain more details of the detected attack, such as the n-th minutes when the attacks happened and the m-th network ports that were attacked. To deal with this problem, we evaluated the adption of a similarity analysis between legitimate traffic and minutes of traffic from time frames classified as under attack, analysing the variation of the most significant eigenvectors caused by the insertion of anomalous traffic [15]. Therefore, we use cosine similarity analysis according to the following equation 11.

$$s = \left|\frac{v \cdot v_n}{\|v\|\|v_n\|}\right| \tag{11}$$

whrere s denotes the absolute similarity degree, v is the most significant eigenvectors of a selected set of minutes without network attack, and v_n is the more significant eigenvector obtained after append the target minute n of traffic to be performed the DoS and port scan attack detection.

If $s = 1$, then the two eigenvectors are similar and no anomaly is detected. Smaller values of s means less similarity and can indicate an anomaly, according to a defined threshold l.

For incremental and detailed timing attack detection through similarity analysis, we adopted a window w as reference of traffic without network attack, selected according to previous attack detection, by time frame, done through MOS schemes and eigen analysis. The reference eigenvectors v is calculated from the selected traffic w, which is composed by z minutes of legitimate network traffic. Thus, the target minutes, of the time frames predicted as under attack, shall be incrementally and individually appended into w, creating a resultant w_n, in order to perform eigen decomposition and obtain the eigenvector v_n, for performing similarity analysis against the reference eigenvectors v. The following equation 12 denotes this step of the algorithm:

$$? \tag{12}$$

Therefore, each minute of the selected $\hat{Q}$ time frames shall be incrementally appended into w for obtaining w_n and the eigenvector v_n, until detect the first n-th minute under attack. Subsequently, w_{n-1} became the new reference of traffic without network attack, $w = w_{n-1}$, and each subsequent minute must be individually evaluated, without incremental append, but doing individual appended into w for obtaining the next eigenvectors v_n for individual time similarity analyis, in order to estimate the vector $\hat{N}$ of the n-ths minutes under attack, as defined by equation 13.

$$? \tag{13}$$

This approach, of incremental similarity analysis followed by individual analysis after an attack detection, allow to identify the attack period, detecting the first and last time under attack, due to the variation of the most significant eigenvector become more significant comparing a traffic under attack against a traffic with no attack, according to results which will be discussed in section 5.

Given $\hat{N}$, which is the set of n-ths minutes under attack, it is still necessary to obtain more details about the identified network attack, such as the network ports that were attacked during each detected minute under attack. Hence, we also applied cosine similarity analysis to identify variation of the most significant eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port during a n-th minute.

For detection of ports under attack, the v last most significant eigenvectors without attack, calculated from w_{n-1}, after the first attack detection, shall be used as reference for similarity against v_n, for each target port m-th of $\hat{N}$ minutes predicted as under attack.

Therefore, v should be calculated from the w_{n-1} last minutes without attack, and v_n should also be calculated from w_{n-1}, but replacing the traffic of the target port m-th of the last minute of w, by the traffic of the same port, during the selected n-th minute under attack. The following equation 14 denotes this approach for detection of ports under attack through similarity analysis:

$$? =? \tag{14}$$

Once v and v_n has been obtained, then the equation 11 shall be used to evaluate the similarity between the most significant eigenvectors v and v_n, with and without port traffic replacement respectively, in order to identify if the traffic replacement highlights the addtion of anomalous traffic by the evaluated port during a minute previously classified as under attack.

Finally, this procedure should be repeated for each m-th target port of $\hat{N}$, in order to individually identify the network ports under network attack during each predicted time.

The algorithm described above can be summarized as following:

Algorithm 1 GETV Attack Detection

Input: Network Traffic Log

Output: GETV

$\mathbf{X}$ = matrix of ports and its occurrences per time $\{$*Data Pre-Processing*$\}$
Q = number of time frames
loop $q = 1$ util $q = Q$
    $\boldsymbol{X}^{(q)} \in \mathbb{R}^{MxN}$
    if $isDosAttack$ **then**
        $\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}$
        $\boldsymbol{S}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T}$
    end if
    if $isPortscanAttack$ **then**
        $\boldsymbol{y}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}}$
        $\boldsymbol{R}_{xx}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^T}$
    end if
    $\boldsymbol{E}^{(q)} = \boldsymbol{E}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{E}^{(q)^T}$
    $[\lambda_m^{(1)}, \lambda_m^{(2)}, \lambda_m^{(3)}, ..., \lambda_m^{(q)}] = \boldsymbol{E}_m^{(q)}$
    $\lambda_m^{(1)} > \lambda_m^{(2)} > \lambda_m^{(3)} > ... > \lambda_m^{(q)}$
end loop

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

$\boldsymbol{K}_1^q = \lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \cdots, \lambda_1^{(q)}$

$\boldsymbol{GETV} = \boldsymbol{K}_1^q$

5 Experimental Results

In this section we present the analyzed scenario, the simulated traffic and all obtained results for MOS schemes evaluation, eigen analysis and similarity analysis for DoS and port scan attack detection.

5.1 Analyzed Scenario

The environment of the analyzed scenario is composed by two computers and one router with access to Internet and to an internal network. Where, we performed a simulation of legitimate traffic and DoS and port scan network attacks. During the simulation and traffic generation, one computer assumes the role of attacker, while the other is the victim, according to scenario represented by Figure 6.

During the simulation we generated a set of network traffic which was modelled as legitimate, noise and malicious traffic, where the victim performs legitimate ac-

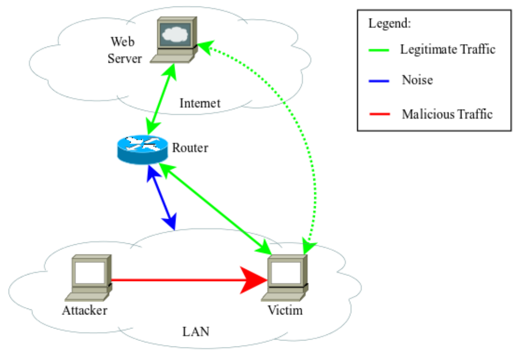

Figure 6 Scenario. Analyzed Scenario.

tivities, that can be characterized by web access. In many organizations this type of traffic is predominant, since most of corporate services are web-based, such as: access to the webmail, web pages, cutstomized web-based systems and cloud services. It is possible to characterize the traffic of a DHCP service as an example of noise associated with the transport layer. For malicious traffic, we evaluated traffic from three types of network attacks: synflood, fraggle and port scan. These attacks were simulated using well known security tools, such as Nmap to port scan, Metaspoit to synflood attack and Hping to lead the fraggle attack.

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

For each time frame q, a traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17x20}$ was obtained, as well as a covariance $\boldsymbol{S}_{xx}^{(q)} \in \mathbb{R}^{17x17}$ (Equation 3) and a correlation matrix $\boldsymbol{R}_{xx}^{(q)} \in \mathbb{R}^{17x17}$ (Equation 4), assuming that in this paper $q = 1, 2, 3, 4, 5$ and 6.

The simulation started at 21:00h, the first time frame 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 performed 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.

5.2 Largest Eigenvalues Analysis

For the evaluation of MOS Schemes acuracy for DoS and port scan detection, our approach defines that it is necessary to obtain the largest eigenvalue of each evaluated time frame, through eigen decomposition from a covariance or correlation matrix calculated from the evaluated network traffic, as the algorithm described in previous section 4.

Through eigenvalue analysis of the network traffic with DoS o porscan attack, it is possible to visualize a significant difference between the largest eigenvalues and the remain eigenvalues, which can indicate a relationship between an outlier and time frames under attack, but still requiring further analysis for detailed and conclusive results, as following discussed.

Figure 7 graphically represents the eigenvalues calculated from covariance matrix of the network traffic used to evaluate the synflood attack identification. In this figure its possible to see that the largest eigenvalue, which is related to the simulated synflood attack ($q = 4$), stands out significantly from the others eigenvalues.

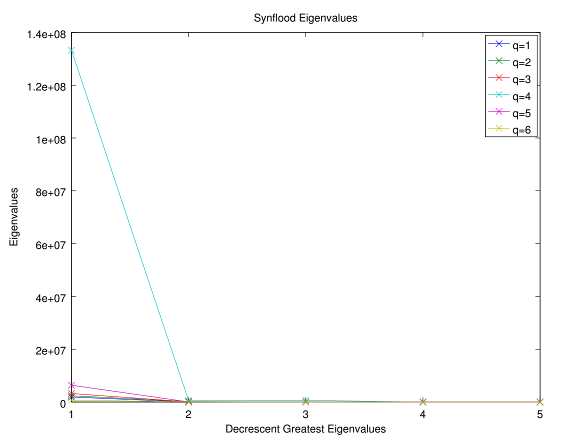

Figure 7 Synflood Eigenvalues. Eigenvalues of the covariance matrix (synflood).

Figure 8 graphically represents the eigenvalues calculated from covariance matrix of the matrix used for fraggle attack detection.

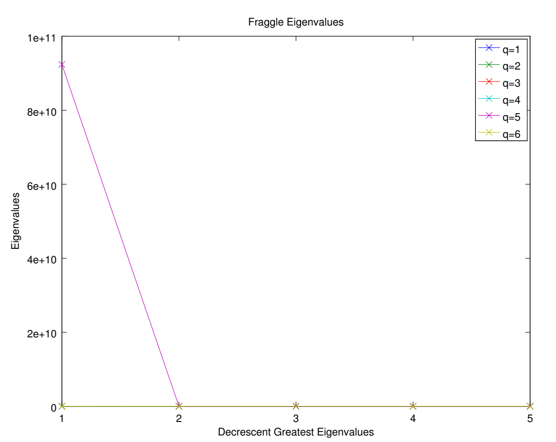

Figure 8 Fraggle Eigenvalues. Eigenvalues of the covariance matrix (fraggle).

In this figure it is possible to see that the largest eigenvalue, which is related to this attack ($q = 5$), stands out significantly from the others eigenvalues, in accordance with the result shown in Figure 7 for the synflood attack analysis.

Figure 9 graphically represents the eigenvalues calculated from correlation matrix of the network traffic matrix evaluated for port scan detection. As analyzed for the synflood and fraggle attacks, it is possible to observe that the largest eigenvalue, related to this attack ($q = 3$), stands out significantly from the others eigenvalues.

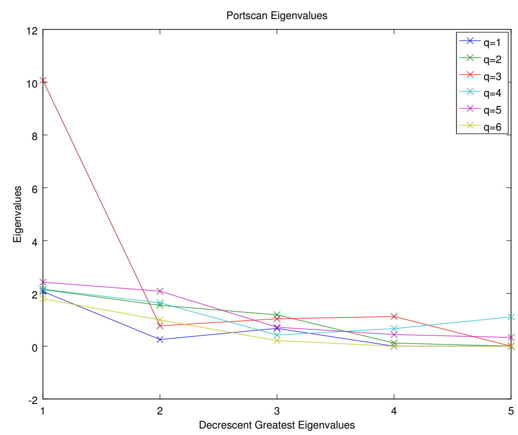

Figure 9 Port scan Eigenvalues. Eigenvalues of the correlation matrix (port scan).

Table 1 presents the values of the largest eigenvalues of each time frame q-th for port scan, synflood and fraggle detection.

Table 1 Largest Eigenvalue related to attacks detection

Time Frame 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

In Table 1 it is possible to observe the significant variationof 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, for port scan attack detection, although the largest eigenvalue presented no too large variance to the second one, if compared to synflood or fraggle attacks, it clearly deviates from the remain largest eigenvalues.

These results righligh that all q-ths time frames, where a network attack was simulated, presented high significat variance between the largest eigenvalue and the remain eigenvalues, obtained from covariance matrix, for DoS detection, or from correlation matrix, for port scan detection. Therefore, we proposed [7] apply the vector of the largest eigenvalues to MOS schemes in order to evaluate their accuracy for identification of time frames under attack, motivated by the fact that it is relevant to apply MOS schemes to automate the attack detection process, taking into account the characteristics of the evaluated eigenvalues.

5.3 MOS Schemes Evaluation

On previous work [7] we evaluated the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE MOS schemes for synflood and port scan attack detection. In this work we extended that evaluation for fraggle attack detection, applying the same schemes to fraggle attack detection over the traffic presented in section 3, as results shown on following Table 2.

Table 2 MOS schemes applied to port scan and DoS detection

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**

It was expected 1 as the model order value when there was only one attack, values greater than 1, returned by one MOS 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 into one vector of largest eigenvalues, showing the presence of two attacks, as indicated by the d real values of Table 2.

With the results shown by Table 2, it is possible to observe that two MOS schemes stand out from the others, EDC and EFT. Efficient Detection Criterion (EDC) and Exponential Fitting Test (EFT) are the most effective schemes, correctly estimating the number of attacks in comparison to the expected values for effetctive attack detection, as defined by the column of real values in Table 2. The AIC and MDL schemes are satisfactory only for port scan detection, however SURE and RADOI schemes did not show effective results for port scan or DoS detection.

According to Table 2, EDC and EFT estimated correctly the number of attacks of a time frame vector, indicating that ocurred $\hat{d}$ network attacks, but not providing additional details, what highlights the necessity of complementary approaches in order to estimate the time and ports under attack. Hence, we propose apply eigen analysis to estimate the q-th time frames under attack and eigen similarity analysis to estimate the minutes and ports under attack.

5.4 Eigenvalue Analysis

According to results presented in Section 5.2, the largest eigenvalue stands out significantly from the others eigenvalues of an evaluated q-th time frame. This behavior can also be observed in the largest eigenvalues analysis, according to results presented in Table 1, where it is possible to observe that the $\hat{d}$ largest eigen values of the time frames under attacks stand out significantly from the others largest eigenvalues.

Therefore, it is possible to observe and conclude that the $\hat{d}$ largest eigenvalues correspond to the respectives q-ths time frames under attack, which is denoted by $\hat{Q}$ and can be calculated through the Equation 10.

5.5 Eigen Similarity Analysis

In this paper we propose to apply eigen similarity analysis for detecting the minutes and ports under attack, from each q-th time frames under attack defined by $\hat{Q}$. Hence, we applied our approach to the time frames where $q = 3$, $q = 4$ and $q = 5$ to respectively evaluate the effectiveness of our approach for port scan, synflood and fraggle attack detection.

5.5.1 Time Analysis

We evaluated three approaches for eigen similarity analysis: for the incremental individualized approach, each minute was incrementally appended into w for obtaining v_n to similarity analysis, until detect the first n-th minute under attack. Subsequently, w_{n-1} became the new reference of traffic without network attack and each subsequent minute must have its similarity individually evaluated; for the incremental approach, each n-th minute must be incrementally appended into w, for obtaining the next eigenvectors v_n for individual time similarity analyis; for the individual approach, each n-th minute must be individually appended into w, without incremental append, but doing individual appended into w for obtaining the next eigenvectors v_n for individual similarity analyis.

The Table 3 presents the results of the evaluation of three approaches for similarity analysis of eigenvectors for port scan detection.

Table 3 Eigen Similarity Analysis for Port Scan Detection

Time Frame q	Time n	Similarity Analysis			Has Attack?
		Incremental Individualized	Incremental	Individual	
3	1	0.9946	0.9946	0.9946	no
3	2	0.9934	0.9934	0.9999	no
3	3	0.9912	0.9912	0.9999	no
3	4	0.9888	0.9888	0.9999	no
3	5	0.9856	0.9856	0.9998	no
3	6	0.9840	0.9840	0.9999	no
3	7	0.9824	0.9824	1.0000	no
3	8	0.9794	0.9794	0.9999	no
3	9	0.9673	0.9673	0.9926	no
3	10	0.9674	0.9674	0.9997	no
3	11	0.9733	0.9733	0.9993	no
3	12	0.9702	0.9702	0.9993	no
3	13	0.9677	0.9677	0.9999	no
3	14	0.9646	0.9646	0.9998	no
3	15	0.0216	0.0216	0.0276	yes
3	16	0.9621	0.0209	1.0000	no
3	17	0.9611	0.0199	0.9998	no
3	18	0.9612	0.0191	0.9999	no
3	19	0.9613	0.0186	0.9998	no
3	20	0.9638	0.0190	1.0000	no

The Table 3 shows the evaluation of the time frame $q = 3$, when the port scan attack was simulated, considering the incremental individualized, incremental and individual approaches for eigen similarity analysis. According to the presented results, it is possible to observe the high similarity between network traffic without attack, which was larger than 0.9610 for all evaluated cases, and emphasize the expressive low similarity when evaluated the traffic with the simulated port scan attack ($n = 15$), which was lower than 0.0276 for all evaluated approaches.

Comparing the evaluated approaches for similarity analysis, it is possible to observe that all evaluated approaches highlight the low similarity when evaluated the traffic under attack. However, the incremental approach figured out low similarity for times without attack, where $n = 16, 17, 18, 19, 20$, what indicates that the incremental approach can produce false posive results. This behaviour occurs because the incremental approaches appends all selected traffic into the reference traffic for comparison against the original reference traffic, what makes more evident the first lack of similarity but reduces the changing detection capability after an attack be evaluated.

The Table 4 presents the results of the evaluation of the similarity analysis of eigenvectors for synflood detection.

Table 4 Eigen Similarity Analysis for Synflood Detection

Time Frame q	Time n	Similarity Analysis			Has Attack?
		Incremental Individualized	Incremental	Individual	
4	1	1.0000	1.0000	1.0000	no
4	2	0.9999	0.9999	1.0000	no
4	3	0.9997	0.9997	0.9999	no
4	4	0.9998	0.9998	1.0000	no
4	5	0.9965	0.9965	0.9908	no
4	6	0.9975	0.9975	1.0000	no
4	7	0.9977	0.9977	1.0000	no
4	8	0.9980	0.9980	1.0000	no
4	9	0.9987	0.9987	0.9999	no
4	10	0.9991	0.9991	1.0000	no
4	11	0.0085	0.0085	0.0284	yes
4	12	0.0162	0.0120	0.0343	yes
4	13	0.0248	0.0158	0.0427	yes
4	14	0.1243	0.0185	0.1041	yes
4	15	0.0082	0.0162	0.0103	yes
4	16	0.0404	0.0070	0.0580	yes
4	17	0.0397	0.0007	0.0573	yes
4	18	0.0408	0.0042	0.0584	yes
4	19	0.0408	0.0079	0.0584	yes
4	20	0.0477	0.0092	0.0757	yes

The Table 4 shows the evaluation of the time frame $q = 4$, when the synflood attack was simulated, considering the incremental individualized, incremental and individual approaches for eigen similarity analysis. According to the presented results, it is possible to observe the high similarity between network traffic without attack, which was larger than 0.9907 for all evaluated cases, and emphasize the expressive low similarity when evaluated the traffic with the simulated synflood attack (between $n = 11$ and $n = 20$), which was lower than 0.1244 for all evaluated approaches.

The incremental approach produced better results if compared with other evaluated approaches, with lower values and maximum of 0.0185 for times under attack, but this approach presents change detection limitation after the first outlier of similarity, as shown in Table 3 for port scan detection.

Comparing the incremental individualized and the individual approaches for eigen similarity analysis, it is possible to observe that the incremental individualized approach obtain lowest values for almost all cases, except for the time $n = 14$, where incremental individualized approach identified a larger similarity than the individual approach. The incremental individualized appends information about each evaluated traffic, therefore it incorporates traffic behaviours that can reduce the outlier capability detection, as ocurred for the time $n = 14$.

The Table 5 presents the results of the eigen similarity analysis evaluation for fraggle detection.

Table 5 Eigen Similarity Analysis for Fraggle Detection

Time Frame q	Time n	Similarity Analysis			Has Attack?
		Incremental Individualized	Incremental	Individual	
5	1	1.0000	1.0000	1.0000	no
5	2	0.9999	0.9999	1.0000	no
5	3	1.0000	1.0000	1.0000	no
5	4	0.9999	0.9999	1.0000	no
5	5	0.9993	0.9993	0.9997	no
5	6	0.9993	0.9993	0.9997	no
5	7	0.9994	0.9994	1.0000	no
5	8	0.9995	0.9995	1.0000	no
5	9	0.9995	0.9995	1.0000	no
5	10	0.9995	0.9995	1.0000	no
5	11	0.0031	0.0031	0.0021	yes
5	12	0.0019	0.0025	0.0009	yes
5	13	0.0030	0.0026	0.0020	yes
5	14	0.0030	0.0027	0.0020	yes
5	15	0.0030	0.0028	0.0020	yes
5	16	0.0012	0.0025	0.0002	yes
5	17	0.0030	0.0026	0.0020	yes
5	18	0.0030	0.0026	0.0020	yes
5	19	0.0030	0.0027	0.0020	yes
5	20	0.0069	0.0023	0.0083	yes

For fraggle attack detection, the lack of similarity between legitimate and malicious traffic was more evident than for the evaluation of synflood and port scan detecion. This becaviour can be explained by the number of packets generated through the fraggle attack simulation, that was significative larger than the number of packets generated during the synflood simulation. Considering the three approaches, the largest value for times under attack was 0.0083, while the shortest value for times without attacks was 0.9993.

Therefore, considering the evaluation for port scan, synflood and fraggle detection, the incremental approach can produce false positive results, while the individual and incremental individualized approaches produce quite similar results, even though the individual approach be more simple and require less memory and processing time.

These results highlight the capabilitiy of change detection based on similarity between legitimate and malicious traffic from DoS or port scan attacks, endorsing the effectiveness and safety for adoption of threshold for attack detection through eigen similarity analysis.

5.5.2 Port Analysis

Given $\hat{N}$, which is the set of estimated n-ths minutes under attack, it is possible to apply cosine similarity analysis to identify variation of the most significant eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port, during a n-th minute.

Therefore, we evaluated the incremental individualized and individual approaches of eigen similarity analysis, for detection of ports under DoS and port scan attacks, according to results presented in following tables. For this evaluation, the v last most significant eigenvectors without attack was used as reference for similarity analysis against each target port m-th.

The Table 6 presents the results of the evaluation of eigen similarity analysis for detection of ports under port scan attack, showing only the time frame $q = 3$ and minute $n = 15$, due to the simulated port scan attack occurred only at this time, although the remain time frame has been completely evaluated and presented high similarity to the reference of traffic without network attack.

Table 6 Eigen Similarity Analysis for Detection of Ports Under Port Scan Attack (q=3 and n=15)

Port p	Approaches		Has Attack?
	Incremental Individualized	Individual	
80	0.9999	0.9999	no
443	0.9999	0.9999	no
53	0.9999	0.9999	no
21	0.9999	0.9997	yes
22	0.0298	0.9997	yes
23	0.0298	0.9997	yes
25	0.0298	0.9997	yes
110	0.0298	0.9997	yes
143	0.0298	0.9997	yes
161	0.0298	0.9997	yes
69	0.0298	0.9997	yes
123	0.0298	0.9997	yes
445	0.0298	0.9997	yes
600	0.9999	0.9999	no
19	0.9999	0.9999	no
67	0.9999	0.9999	no
68	0.9999	0.9999	no

The incremental individualized approach presented more sensibility to anomaly detection than the individual approach, the former produced the identification of a low similarity of 0.0298 for almost all ports under attack, unless the port 21, although our simulation has attacked this port. The individual approach was not able to identify low similarity for ports under attack, resunting in values of 0.9997 for ports with anomalous traffic and 0.9999 for ports without network attack.

For the evaluation of our proposed approaches for identification of ports under synflood and fraggle attack, we analyzed all minutes of each time frame where one attack location was estimated, but we will only show the results of the firt minute where a low simillarity was idenfified, where $n = 11$, since the results obtained for the evaluation of traffic without attack presented high similarity to the reference traffic, with similarities near of 0.9999, and because the evaluation of the other minutes under attack presented results quite similar to the resuts shown in the following tables.

The Table 7 presents the results of the evaluation of eigen similarity analysis for detection of ports under synflood attack, showing only the time frame $q = 4$ and minute $n = 11$.

According to results presented in Table 7, both approches identified low similarity for the traffic of port 600, which was the target port of our simulated synflood

Table 7 Eigen Similarity Analysis for Detection of Ports Under Synflood Attack (q=4 and n=11)

Port p	Approaches		Has Attack?
	Incremental Individualized	Individual	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	nos
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	0.0077	0.0427	yes
19	1.0000	1.0000	no
67	1.0000	1.0000	no
68	1.0000	1.0000	no

attack, but the incremental individualized approach identified the lowest similarity and presented better sensibility to identification of synflood attack through eigen similarity analysis assisted by threshold definition.

The Table 7 presents the results of the evaluation of eigen similarity analysis for detection of ports under fraggle attack, showing only the time frame $q = 5$ and minute $n = 11$.

Table 8 Eigen Similarity Analysis for Detection of Ports Under Fraggle Attack (q=5 and t=11)

Port p	Approaches		Has Attack?
	Incremental Individualized	Individual	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	no
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	1.0000	1.0000	no
19	0.0031	0.0004	yes
67	1.0000	1.0000	no
68	1.0000	1.0000	no

The results for the avaluation of ports under fraggle attack, shown by Table 8 were similar to the results obtained for synflood analysis, with the identification of low similarity for traffic of the port under attack, but for fraggle analysis, the individual approach identified the lowest similarity, that was 0.0004 while the incremental individualized approach obtained a similarity of 0.0031.

The incremental individualized approach was able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches presented false positives or low sensibility to eigen similarity analyis for network attack detection. This approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

According to the shown significant lack of similarity between legitimate and malicious traffic, it is possible to adopt safe thresholds for DoS and port scan detection through eigen similarity analysis.

6 Conclusion and Future Works

This paper extended the evaluation of the Greatest Eigenvalue Time Vector Approach (GETV) approach for detecting fraggle attacks, showing that GETV can be applied to attacks involving port scanning, and DoS. For these types of attack, the technique proved to be quite effective for estimating time frames under attack, but still requiring more information for detailed attack detection. Therefore, we proposed a novel approach for detailed network attack detection, based on eigen similarity analysis.

We showed, through experiments, that synflood, fraggle and port scan attacks could be detected accurately and with great detail in an automatic and partially blind fashion, applying signal processing concepts for traffic modeling and through approaches based on MOS and eigen similarity analysis. The main contributions of this work were: the extension of an approach based on MOS combined with eigen analysis to blindly detect time frames under network attack; The proposal and evaluation of the accuracy of eigen similarity analysis for detailed network attack detection.

The incremental individualized approach of eigen similarity analysis, was able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches presented false positives or low sensibility to eigen similarity analyis for network attack detection. Therefore, the incremental individualized approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

According to the significant similarity difference between legitimate and malicious traffic, it is possible to adopt safe thresholds for DoS and port scan detection through eigen similarity analysis.

Appendix A: 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 [19], 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 [20]. 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 [21, 22]; Minimum Description Length - MDL [22, 23]; Efficient Detection Criterion - EDC [24];

Stein's Unbiased Risk Estimator - SURE [25]; RADOI [26] and Exponential Fitting Test - EFT [5, 27, 28].

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 [19].

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 [19]:

$$\hat{d} = \arg\min_{k} \; J(k), \tag{15}$$

where

$$J(k) = -N(\alpha - k) \log\left(g(k)/a(k)\right) + p(k, N, \alpha), \tag{16}$$

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 9.

Table 9 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 [27].

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

$$E\{\lambda_i\} = E\{\lambda_1\} \cdot q(\alpha, \beta)^{i-1}, \tag{17}$$

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:

$$q(\alpha, \beta) = \exp\left\{ -\sqrt{\frac{30}{\alpha^2 + 2} - \sqrt{\frac{900}{(\alpha^2 + 2)^2} - \frac{720\alpha}{\beta(\alpha^4 + \alpha^2 - 2)}}} \right\}, \tag{18}$$

where $0 < q(\alpha, \beta) < 1$. According to [28], if $M \leq N$, then $\beta = N$.

Figure 10 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 [14].

Figure 10 EFT Example. Example of application of EFT [14].

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. 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
2. Lu, W., Ghorbani, A.A.: Network anomaly detection based on wavelet analysis. EURASIP J. Adv. Signal Process **2009**, 4–1416 (2009). doi:10.1155/2009/837601
3. Huang, C.-T., Chang, R.K.C., Huang, P.: Editorial: Signal processing applications in network intrusion detection systems. EURASIP J. Adv. Signal Process **2009**, 9–192 (2009). doi:10.1155/2009/527689
4. Zonglin, L., Guangmin, H., Xingmiao, Y., Dan, Y.: Detecting distributed network traffic anomaly with network-wide correlation analysis. EURASIP J. Adv. Signal Process **2009**, 2–1211 (2009). doi:10.1155/2009/752818
5. 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)
6. 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)
7. 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)
8. 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
9. 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)
10. 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
11. 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
12. 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
13. 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)
14. 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
15. Lee, Y.-J., Yeh, Y.-R., Wang, Y.-C.F.: Anomaly detection via online oversampling principal component analysis. Knowledge and Data Engineering, IEEE Transactions on **25**(7), 1460–1470 (2013). doi:10.1109/TKDE.2012.99
16. 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
17. Jolliffe, I.: Principal Component Analysis. Wiley Online Library, ??? (2005)
18. 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)
19. 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)
20. 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)
21. Akaike, H.: A new look at the statistical model identification. Automatic Control, IEEE Transactions on **19**(6), 716–723 (1974)

22. Wax, M., Kailath, T.: Detection of signals by information theoretic criteria. Acoustics, Speech and Signal Processing, IEEE Transactions on **33**(2), 387–392 (1985)
23. 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)
24. 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)
25. 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
26. 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)
27. 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
28. 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)