Available online at www.sciencedirect.com

journal homepage: www.elsevier.com/locate/cose

ELSEVIER

Computers
&
Security

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

***Danilo Fernandes Tenório*[a], *João Paulo Carvalho Lustosa da Costa*[a], *Edison Pignaton de Freitas*[a,b] *and Rafael Timóteo de Sousa Júnior*[a]**

[a]*Department of Electrical Engineering, University of Brasilia (UnB), 70910-900, Brasília-DF, Brazil*
[b]*Department of Technology, Federal University of Santa Maria (UFSM), 98400-000, Frederico Westphalen-RS, Brazil*

ARTICLE INFO

Article history:
Received 00 December 00
Received in revised form 00 January 00
Accepted 00 February 00

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

ABSTRACT

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

This work proposes an innovative technique for automatic blind detection of malicious traffic by taking into account anomalies in the monitored traffic on a network. First we model our acquired data as a superposition of three types of traffics: legitimate traffic related to user applications, the noise traffic not associated with the user and the 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 the 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 communication ports (portscan) attacks using a real computer network.

1. Introduction

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 developed malicious techniques (Geus et al, 2010).

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. Examples of equipment or security systems that can be employed are firewalls, intrusion detection systems and intrusion prevention systems (CERT.br, 2010).

The firewalls act as the first line of defense in 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. The incoming and outgoing Internet is inspected by network firewalls. Based on a set of rules, they can allow or block incoming or outgoing traffic. Thereby, network firewalls work based on rules that sequentially interrogate the packages, rule by rule, until a match is found and the same is dropped or released to proceed to the destination (Salah et al, 2012).

* *Corresponding author.* Tel.:+556181511711.
E-mail address: daniloftenorio@gmail.com (D. Tenório), joaopaulo.dacosta@ene.unb.br (J. P. da Costa), edisonpignaton@unb.br (E. Pignaton), desousa@unb.br (R. Timóteo)

Parts of this paper have been published in D. F. Tenório, J. P. C. L. da Costa, and R. T. de Sousa Jr., "Greatest Eigenvalue Time Vector Approach for Blind Detection of Malicious Traffic," The International Conference on Forensic Computer Science (ICoFCS) 2013, Brasília, Brazil, best paper award.

Intrusion detection and intrusion prevention systems are security systems that are used respectively to detect (passive) and prevent (proactive) threats to computer systems and computer networks. Such systems use several ways of functioning, such as: signature-based, anomaly-based or hybrid (Mudzingwa et al, 2012).

In this work, it is proposed an automatic blind malicious traffic detection technique to be used in any computer of a network. In (David et al, 2011; da Costa et al, 2012) the real network traffic data is modeled into three components: the legitimate traffic, the malicious traffic and the noise.

Note that the term "automatic" means that it is not necessary human intervention to assess whether or not there was an attack. The term "blind" refers to the fact that it is not necessary prior information, such as attack signatures or learning periods, to detect the attack.

Inspired by (David et al, 2011; da Costa et al, 2012) the network traffic in this paper is modeled 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). Thus, the modeling is concerned only within the transport layer, so that the scope of attack detection is confined to this layer.

Our proposed technique is based on the eigenvalue decomposition, however, in contrast to (David et al, 2011; da Costa et al, 2012), we consider the time variation of the eigenvalues. To the best of our knowledge, such key consideration has not been applied before in the literature. We show by means of experiments that based on the greatest eigenvalue variation, attacks such as synflood, fraggle and portscan can be detected in automatic and blind fashion.

The main contributions of this work are summarized as: 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 related to eigenvalues and eigenvectors, Principal Component Analysis (PCA), and Model Order Selection (MOS), including the main MOS schemes and their differences. Section 5 synflood, fraggle and portscan attacks are characterized, as well as the data collection, data modeling and attack detection. In Section 6 describes the experimental validation with real data, and the evaluation of several MOS schemes presenting the corresponding experimental results validating 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 (He et al, 2008; Ghourabi et al, 2010) and regular analysis of files (Raynal et al, 2004) to detect patterns that indicate the presence of specific attacks in traffic analysis.

Multiple series of data mining are used in (He et al, 2008) 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. Aiming to improve the performance of intrusion detection task, researchers have applied data mining techniques on log analysis (Ghourabi et al, 2010). However, the requirement of the prior collection of large volumes of data is a weak point in the process.

The use of regular analysis of files (Raynal et al, 2004) consists in detecting patterns that indicate the presence of specific attacks in traffic analysis, beyond the statistical study of traffic data collected. An essential feature of this method is the fact that it depends on prior knowledge of the attacks that are intended to be identified, and also the collection of significant amounts of logs for the method to work properly, reducing false positives.

The use of PCA can be seen employed in detecting attacks (Almotairi et al, 2009), but only using PCA without being combined with any other technique, such as Model Order Selection (MOS), requires the subjective character of human intervention, making it impractical for automatic systems besides being prone to errors, such as false positive.

Blind automatic detection of malicious traffic techniques have been developed to honeypots in (David et al, 2011; da Costa et al, 2012). However, traffic on honeypot is simpler because there are no legitimate applications running. It emulates behavior of host within a network in order to deceive and lure attackers (Zakaria et al, 2012).

The data collected in honeypot systems such as traffic capture and operating system logs are analyzed in order to obtain information about attack techniques, general trends of threats and exploits. Because honeypots do not generate legitimate traffic, the amount of data captured is significantly reduced in comparison to a network IDS that captures and analyzes the largest possible amount of network traffic (David et al, 2011).

The use of schemes of Model Order Selection for blind detection in network traffic to identify malicious activities in honeypot was proposed in (David et al, 2011). Criteria for Selecting Model Order are usually evaluated in simulations by comparing the order of the resulting model with the true model order (da Costa, 2007).

Our approach treats a more complex traffic composed of legitimate signal, noise and attack. In contrast with (He et al, 2008; Ghourabi et al, 2010) and (Raynal et al, 2004), our scheme does not require either significant amount of logs in order to detect attacks nor prior data collection to make comparisons and decide the existence of malicious traffic. Moreover, in constrast with (Almotairi et al, 2009), the attack detection is automatic, requiring no human intervention.

In (R. Puttini et al, 2006) an alternative technique to the one presented in this paper, through a general explanation of a system intrusion detection based on detection of traffic anomalies and a discussion of false positives and false negatives.

The Table 1 shows a comparison between the related works and the current proposed in this paper (Tenório et al, 2014) showing that the present study proposes a 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 (**a**, **b**), matrices by uppercase bold letters (**A**, **B**), and $a_{i,j}$ denotes the (*i, j*) elements of the matrix **A**. The superscripts ^{T} and $^{-1}$ are used for matrix transposition and matrix inversion, respectively.

4. Mathematical Concepts

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

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

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

4.1. Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors, commonly used in linear algebra, can reveal very 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.

Complex systems can be represented by matrices that in certain cases are sometimes difficult to handle, requiring a large computational effort and great amount of memory.

Let a square matrix **G** with real-valued elements be decomposable into two matrices **F** and **G**,

$$\mathbf{G} = \mathbf{F}\mathbf{B}\mathbf{F}^{\mathrm{T}}, \tag{1}$$

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

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

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

Note that the eigenvalues provide information about rank of the matrix **G** and that the vectors of the matrix **F** must necessarily be linearly independent.

Being $\mathbf{G}^{\mathrm{T}}\mathbf{G} \in \mathbb{R}^{n\times n}$ and $\mathbf{G}\mathbf{G}^{\mathrm{T}} \in \mathbb{R}^{m\times m}$ symmetric matrices, the eigenvectors of $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ are orthogonal to each other as well as the eigenvectors of $\mathbf{G}\mathbf{G}^{\mathrm{T}}$ are orthogonal to each other.

4.2. Correlation and Covariance Data

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

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

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

The definition of sample covariance matrix is:

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

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

The definition of sample correlation matrix is:

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

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 areas of research such as: analysis of Internet traffic, economy, image processing and genetics. PCA is mainly used to reduce the size of a data set, using for this uncorrelated variables, called the principal components (PC). This transformation into another set of variables occurs with the least possible loss of information or even variables that contain only noise (this process is called denoising) (Jollif, 2002).

The principal components generated are a linear combination of the original variables, 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 (Cichocki et al, 2009).

4.4. Model Order Selection (MOS)

In many applications of digital signal processing, 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 from the main components, applying reduced data set analyzed. Moreover, for many parameter estimation techniques the model order is crucial (da Costa et al, 2009), 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 such that some criterion is satisfied (Rajan et al, 1997). 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: Akaike's Information Theoretic Criterion - AIC (Akaike, 1974; Wax et al, 1985); Minimum Description Length - MDL (Rissanen et al, 1998; Wax et al, 1985); Efficient Detection Criterion - EDC (Zhao et al, 1986); Stein's Unbiased Risk Estimator - SURE (Ulfarsson et al, 2008); RADOI (Radoi et al, 2004) and Exponential Fitting Test - EFT (Goffraud et al, 1996; Quinlan et al, 2007; da Costa et al, 2011).

In AIC, MDL and EDC schemes, 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 (da Costa et al, 2009).

Basically, the difference between the AIC, MDL and EDC schemes is the penalty function $p(k, N, \alpha)$, so these techniques can be written in general as (da Costa et al, 2009) :

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

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 Table 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 scheme 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 and the profile of the eigenvalues can be approximated by a decaying exponential (Grouffaud et al, 1996).

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

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

where E{·} is the expectation operator, and it is considered that the eigenvalues are ordered so that λ_1 represents the largest eigenvalue. The term $q(\alpha, \beta)$ 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 (Quinlan et al, 2007) if $M \leq N$, then $\alpha = M \; and \; \beta = N$.

The Fig. 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 (da Costa et al, 2007).

Fig. 1 - Example of application of EFT (da Costa et al, 2007)

5. Proposed Solution

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

5.1. Data Collection

The log information of a computer connected to the 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 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:

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

where q represents the q-th period of time.

Thus, according to the Fig. 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.

Fig. 2 – Obtaining the traffic matrix $\mathbf{X}^{(q)}$.

The matrix $\mathbf{X}^{(q)} \in \mathbb{R}^{M \times N}$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 at the n-th instant, in the q-th time period.

The legitimate traffic $\mathbf{S}^{(q)}$ 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 Fig. 3 presents the legitimate traffic obtained during experiments.

Fig. 3 – The legitimate traffic.

It is considered as noise $\mathbf{N}^{(q)}$ all traffic that is not directly associated with operations performed by the user, but it is not also a malicious traffic. An example of noise is the service of automatic acquisition of logical IP address network (DHCP). Independent of any user operation, the machine will receive an IP address, since it is to perform configured this address acquisition. Fig. 4 shows the noise obtained during simulations.

Fig. 4 – The noise traffic.

The traffic coming from a malicious activity is represented by the matrix $\mathbf{A}^{(q)}$. In this work it is considered only the traffic originating from flood attacks, aiming to cause denial of service, and port scanning attacks. If the rank $\{\mathbf{A}^{(q)}\} \neq 0$, there is malicious traffic, on the other hand, if the rank $\{\mathbf{A}^{(q)}\} = 0$, there is no malicious traffic. This paper shows how to detect the rank $\{\mathbf{A}^{(q)}\}$ given only the matrix $\mathbf{X}^{(q)}$.

5.3. Synflood, Fraggle and Portscan

The attacks focused in this work are: synflood, fraggle and portscan. The first two attacks are classified as denial of service attacks, while the last one is a port scanning attack.

Because of the TCP protocol is a connection-oriented protocol, 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 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 Fig. 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.

Fig. 5 – 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 IP address of the victim (IP spoofing). With the broadcast, each network host receives a huge amount of requests "UDP echo", passing them all to reply to the source address, which is fake, the IP address of the victim. This attack can affect the entire network, because all their hosts receive many requests "UDP echo" and respond with the ICMP protocol, passing each one to act as an "amplifier" of the attack, in relation to the host that had the fake IP. Thus, the victim had fake the IP address, receives packages from all these hosts, being unable to perform their normal activities, thus 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 is due to the fact that UDP does not by itself be able to know if the segment sent has reached its destination, , i.e. as UDP is connectionless, no confirmation is sent back.

The Fig. 6 depicts the fraggle attack, which was carried out during the experiments. In time interval of ten minutes, was more than 6,000,000 packets related to the attack can be counted, an unusual traffic on a data network, especially by being concentrated in a short period of time.

Fig. 6 – The traffic characterized by fraggle.

Portscan is the process of connecting to TCP and UDP ports on targets of interest in order to determine what services are running or which are in the state of listening. Identifying ports in the state of listening is crucial to determine the type of the victim's operating system and applications in use.

There are several available scanning techniques, including: TCP SYN scan, TCP ACK scan, UDP scan, etc. This work makes use of scanners TCP SYN scan and UDP scan.

The TCP SYN scan technique is called half-open scanning, because there is not a full TCP connection. In this scan, a SYN packet is sent to the destination port, 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 on. 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 door is closed. If a message is not received, then door 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 Fig. 7 depicts the portscan attack that was experimented. It is possible 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.

Fig. 7 – The traffic characterized by portscan.

5.4. Attack Detection

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

The attack detection process starts at $q = 1$, obtaining the matrix $\mathbf{X}^{(1)} \in \mathbb{R}^{M \times N}$, step (A).

For detection of synflood and fraggle (denial of service) attacks, it is necessary to calculate the covariance matrix $\mathbf{S}_{xx}^{(q)}$, step (C), since in this case the main components are dominated by the variables with more variance, in accordance with what has been discussed in Section 4.3. To obtain the covariance matrix $\mathbf{S}_{xx}^{(q)}$ it is essential for each variable (in the case of this work, each port), the calculation of the deviations of the respective elements in relation the average, step (B) of Fig. 8.

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

Once the $\mathbf{S}_{xx}^{(q)}$ and $\mathbf{R}_{xx}^{(q)}$ has been obtained, proceeds with the eigenvalue decomposition (EVD) - steps (D) and (G) respectively - in order 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 $\mathbf{X}^{(q)} \in \mathbb{R}^{M\times N}$, q = 1, 2, 3, ..., Q and the matrices $\mathbf{S}_{xx}^{(q)}$or $\mathbf{R}_{xx}^{(q)}$, 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, in which it is related to the greatest eigenvalue for each q-th time period.

Fig. 8 – Proposed GETV attack detection approach

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

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

By obtaining the vector GETV, $\lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)}, \dots, \lambda_1^{(Q)}$, 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), 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, the process of detecting attacks ends.

6. Experimental Results

In this section it is presented the analyzed scenario and all the 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 the Internet and to the internal network (LAN). One of the computers has the role of attacking while the other is the victim, according to Figure 9.

Fig. 9 – Analyzed Scenario.

The victim performs legitimate activities, mainly web access. In many organizations this type of access is done very often, since most of corporate services are web-based, such as: access to the e-mail server, access to documents protocol and access to intranet page.

As an example of noise associated with the transport layer of the OSI model, it is possible to cite the traffic associated with the DHCP service. In the case of malicious traffic, it is composed by the traffic associated with three types of attacks: synflood, fraggle and portscan, detailed in Section 5.1. The attacks were simulated using well known professionals security tools. For portscan was used nmap, to synflood attack the metasploit and to lead the fraggle attack the hping tool.

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 $\mathbf{X}^{(q)} \in \mathbb{R}^{17\times 20}$ was obtained, as well as a covariance $\mathbf{S}_{xx}^{(q)} \in \mathbb{R}^{17\times 17}$ and a correlation matrix $\mathbf{R}_{xx}^{(q)} \in \mathbb{R}^{17\times 17}$, 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), 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 his 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 Fig. 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.

Fig. 10 – Eigenvalues of the covariance matrix (synflood).

The Fig. 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 Fig. 10 for the synflood attack.

Fig. 11 – Eigenvalues of the covariance matrix (fraggle).

The Fig. 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.

Fig. 12 – Eigenvalues of the covariance matrix (portscan).

6.3. GETV

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

In Table 3 it is possible to observe how different the eigenvalues associated with attacks are relative to the others. At $q = 4$, where the synflood attack occurred, the maximum eigenvalue obtained in this period is approximately 21 times larger than the second one. At $q = 5$, where the fraggle attack occurred, the maximum eigenvalue obtained in this period is about 29,000 times larger than the second one. At $q = 3$, where the portscan attack occurred, the maximum eigenvalue obtained in this period is approximately 4 times larger than the second one. In the last case, although the greatest eigenvalue is not too high, compared to synflood or fraggle attacks, it is 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, using for this uncorrelated variables, called principal components (PC). This transformation into another set of variables 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	923840216 11	6367983	923840216 11	2,4253
6	708335	708335	708335	1,7948

The principal components are a linear combination of the original variables, they 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.

According to the scenario of this work, 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 size of the set is 17-dimensional. With a PC the dataset can be reduced, for example, 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 towards other traffic, obviously it will be selected for analysis the time periods related to these attacks: $q = 3$ for portscan attack, $q = 4$ for synflood attack and $q = 5$ for the fraggle attack.

To the synflood attack in Fig. 13 that the variance of PC1 (first PC) is totally dominated by the components of the attack, in other words, 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 = 4. Furthermore, as discussed in Section 4.3, the variance of PC1 is equals to the largest eigenvalue of the matrix $\mathbf{S}_{xx}^{(4)}$.

For the fraggle attack in Fig. 14 that the variance of PC1 (first PC) is totally dominated by the components of the attack, in other words, 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 = 5. Furthermore, as discussed in Section 4.3, the variance of PC1 is equals to the largest eigenvalue of the matrix $\mathbf{S}_{xx}^{(5)}$.

For the portscan attack in Fig. 15 that the variance of PC1 (first PC) is totally dominated by the components of the attack, in other words, 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 $\mathbf{R}_{xx}^{(3)}$.

Fig. 13 – The two first principal components (synflood).

Fig. 14 – The two first principal components (fraggle).

Fig. 15 – The two first principal components (portscan).

6.5. Model Order Selection Schemes

Although the effect of the attack through the PCA, it is relevant the application of MOS schemes in order to make the process automated, taking into account the profile of the eigenvalues.

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

With the results shown below, it is possible observe that two schemes stand out from the others. The Efficient Detection Criterion (EDC) and the Exponential Fitting Test (EFT) are the most consistent, 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.

Table 4 – MOS schemes applied to the GETV

Type of analysis	**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 AIC and MDL scheme is satisfactory only in detecting the portscan. The SURE and RADOI schemes did not show consistent results for either case.

The value of the model order equal to one when there is the attack is expected. Due to the presence of the principal component, the component that stands out from the others,, is used as the reference to detect the attack.

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

7. Conclusion and Future Works

This paper proposes the Greatest Eigenvalue Time Vector Approach (GETV) approach for the detection of portscan, synflood and fraggle attacks. More generally, the technique 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 the detection automated, the schemes for selecting the model order were. Through some experiments it is concluded that the GETV combined with EFT and EDC schemes presented more consistent results for the proposed 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 discusses only such technique on the transport layer. Furthermore, the GETV can be also combined with other techniques, such as data mining and analysis of regular files, in order to detect attacks that slightly escape from the behavior shown in this work. We highlight that the GETV technique can be also applied in other scientific areas, since it is a general concept about eigenvalues variation.

Acknowledgements

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

REFERENCES

H. Akaike, “A New Look at the Statistical Model Identification,” in IEEE Transactions on Automatic Control, vol. 19, pp. 716-723, December 1974.

S. Almotairi, A. Clark, G. Mohay and J. Zimmermann, “A Technique for Detecting New Attacks in Low-Interaction Honeypot Traffic,” in Fourth International Conference on Internet Monitoring and Protection, pp. 7-13, May 2009.

A. Cichocki, R. Zdunek, A. H. Phan and S. I. Amari, “Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation,” First Edition, Wiley, USA, 2009.

J. P. C. L. da Costa, E. P. de Freitas, B. M. David, A. M. R. Serrano, D. Amaral and R. T. Sousa Júnior, “Improved blind automatic malicious activity detection in honeypot data,” in International Conference on Forensic Computer Science (ICoFCS), pp. 46-45, Sepember. 2012.

J. P. C. L. da Costa, A. Thakre, F. Röemer and M. Haardt, “Comparison of model order selection techniques for high-resolution parameter estimation algorithms,” in 54th International Scientific Colloquium (IWK), pp. 07-10, October 2009.

J. P. C. L. da Costa, M. Haardt, A. Thakre, F. Röemer and G. D. Galdo, “Enhanced Model Order Estimation Using Higher-Order Arrays,” in Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers (ACSSC), pp. 412-416, November 2007.

D.X. Dan, D.Y. Ming, Y. Tao and L. Rong, “Evaluation of AR model order selection approaches,” in International Forum on Information Technology and Applications (IFITA), vol. 1, pp. 704-707, May 2009.

B. M. David, J. P. C. L. da Costa, A. C. A. Nascimento, M. D. Holtz, D. Amaral and R. T. Sousa Júnior, “Blind automatic malicious activity detection in honeypot data,” in International Conference on Forensic Computer Science (ICoFCS), pp. 142-152, October. 2011.

A. Ghourabi, T. Abbes and A. Bouhoula, “Data analyzer based on data mining for honeypot router,” in International Conference on Computer Systems and Applications (AICCSA), pp. 1-6, May 2010.

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

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

D. Mudzingwa and R. Agrawal, “A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS),” in Proceedings of IEEE Southeastcon, pp. 1-6, March 2012.

E. T. Nakamura and P. L. de Geus, “Segurança de Redes em Ambientes Cooperativos,” Novatec Editora Ltda, Brasil, 2010.

R. Puttini, M. Hanashiro, F. Miziara, R. T. Sousa Júnior, L. J. García-Villaba and C. J. Barenco, “On the Anomaly Intrusion-Detection in Mobile Ad Hoc Network Environments,” Personal Wireless Communications, Lecture Notes in Computer Science, Volume 4217, pp. 182-193, January 2006.

A. Quinlan, J. P. Barbot, P. Larzabal and M. Haardt, “Model order selection for short data: An exponential fitting test (EFT),” in EURASIP Journal on Applied Signal Processing, 2007.

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

J. J. Rajan and P. J. W. Rayner, “Model order selection for the singular value decomposition and the discrete Karhunen-Loève transform using a Bayesian approach,” in IEEE Proceedings Vision, Image and Signal Processing, vol. 144, pp. 116-123, April 1997.

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

A.C. Rencher, “Methods of Multivariate Analysis,” Second Edition, Wiley, USA, 2002.

K. Salah, K. Elbadawi and R. Boutaba, “Performance Modeling and Analysis of Network Firewalls,” in IEEE Transactions on Network and Service Management, vol. 9, pp. 12-21, March 2012.

M. O. Ulfarsson and V. Solo, “Rank selection in noisy PCA with SURE and random matrix theory,” in International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3317-3320, April 2008.

M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” in IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP), vol 33, pp. 387-392, April 1985.

W.Z.A. Zakaria and M.L.M. Kiah, “A review on artificial intelligence techniques for developing intelligent honeypot,” in 8th International Conference on Computing Technology and Information Management (ICCM), vol. 2, pp. 696,701, 24-26, April 2012.

L. C. Zhao, P. R. Krishnaiah and Z. D. Bai, “On detection of the number of signals in presence of white noise,” in Journal of Multivariate Analysis, 1986.