Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 3 (2012) 97 – 108

Procedia Technology

The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science

Using Entropy Spaces and Mixtures of Gaussian Distributions to Characterize Traffic Anomalies

Pablo Velarde-Alvarado[a]*, Alberto F. Martínez-Herrera[b], Adalberto Iriarte-Solis[a]

[a]*Universidad Autónoma de Nayarit, Ciudad de la Cultura "Amado Nervo", Tepic, Nayarit, Mexico.*
[b]*Instituto Tecnológico y de Estudios Superiores de Monterrey, Garza Sada Sur 2501,Monterrey, Nuevo Leon, Mexico.*

Abstract

In this paper, a technique for detecting anomalous behavior traffic in a computer network is presented. Entropy space method is based on a 3D-space built on a flow-packet level. The complete set of points obtained in the 3D-space can be seen as a data cloud. Each 3D point in the space is a value of the obtained clusters for each slot of the network traffic. The selected features for the set of points are done by applying Pattern Recognition, Principal Component Analysis, and Kernel Density Estimation. At the next stage, the network traffic can be modelled by using Gaussian Mixtures and Extreme Generalized Distributions, which define the behavior of each selected feature. By integrating this model in an Anomaly-based Intrusion Detection System, anomalous behaviour traffic can be detected easily and early. The effectiveness and feasibility of this model was tested in a Local Area Network of a Campus.

Keywords: Intrusion Detection System, Entropy, Pattern Recognition.

1. Introduction

Identification and early detection of anomalous activities on computer networks are the main goals to be performed by a Network Intrusion Detection System, or NIDS. Despite that the firewalls can be sufficient to

* Corresponding author. Tel.:+52 311 211 88 21.
E-mail address: pvelarde@uan.edu.mx

doi:10.1016/j.protcy.2012.03.011

protect a computer network; several times they are not able to prevent a wide type of attacks such as internal attacks. Besides, when someone wants to use a NIDS, software updates against new attacks are not available in a reasonable time. It means that the NIDS is not able to detect zero-day attacks until the database is updated. Thus, these novel attacks cannot be counteracted immediately [1].

This disadvantage is a common characteristic of signature-based NIDS (S-NIDS). Snort [2] is an example of S-NIDS. The operation of S-NIDS is based on a well-known database. This database contains a set of signatures, each signature for each known attack. When an attack arises in a computer network, the S-NIDS extracts a signature of the anomalous traffic and compares this signature against the database. If a match is found in the database, the S-NIDS launches an alarm to the system. It is evident that if a never seen attack arises, S-NIDS won't be able to detect it. To overcome this disadvantage, a system with different approach has been developed. This system is named Anomaly-based NIDS (A-NIDS).To perform its duties, an A-NIDS has to construct a reliable statistical model by extracting selected traffic features. Then, when the current network traffic is captured, it is compared against the obtained traffic model. A significant deviation between the model and the current traffic means an anomalous behavior, but it does not mean an attack itself. The main advantage of A-NIDS relies on the early detection of zero-day attacks by using the network model. An example of A-NIDS is PAYL [3]. Despite of the advantages of A-NIDS, S-NIDS are easier to be configured and deployed rather than A-NIDS.

Entropy-based methods have been widely investigated and discussed in recent years. These methods have shown to be viable for the task of automatic intrusion detection because they offer more fine-grained insights of the structure and composition of network traffic. Entropy-based characterizations are less affected by traffic sampling processes applied in high speed networks. Consequently, entropy methods become more robust for the task of intrusion detection in sampled environment; this is because entropy preserves the structure of an attack [4, 5].

There are different ways to perform anomaly analysis by using entropy techniques. The first one is related to packet-independent treatment. Under this analysis, each feature is extracted from each packet and collected in the corresponding feature set under a defined window-time, thus forming independent feature sets. For instance, if we choose to collect a set of features such as source IP, IP destination, source Port and Port destination, these features are extracted from each packet. Each feature set is formed under a computable and reasonable window-time, normally $t = 0.5\,\mathrm{s}$ for short time periods or $t = 60\,\mathrm{s}$ for long time periods. Such time definition is established according to the detection accuracy because some attacks are more sensible to the window-time than others. After that, the entropy is computed for each feature set. In addition, the window time can be defined as static window-time or sliced window-time [6, 7]. The former means to sample the network traffic by using the window-time. This is done in such way that the beginning and the end of each window-time do not overlap with another, thus forming independent slots. The latter means overlapping each window-time by adding-dropping a packet when a new packet arises in the network traffic. Thus, the entropy is dynamically computed because when a new packet arises, the oldest packet is dropped and the new packet is added. As consequence, the features of the oldest packet are dropped and the features of the new packet are added to the corresponding feature set. An advantage of using sliced window-time is to obtain accurate changes in the network traffic behavior on-the-fly [8].

This type of analysis (and their variants) is reliable and accurate because each packet feature is taken into account. Nevertheless, it is not recommended for large networks with huge amount of traffic. In the case of large networks, there is an anomaly analysis under a flow-packet treatment. It is described throughout the rest of the paper, especially in the next section. The main advantage of this analysis is that the amount of data obtained is classified and shrunk in such way that the relevant data are extracted, so the redundant data are discarded. Thus, the entropy can be obtained by using only relevant data of the network traffic. Nevertheless, it implies to preserve original features of the network traffic to obtain accurate results.

A flow-packet is a set of unidirectional packet sequences that have common values on each header packet: source IP, source port, destination IP, destination port and protocol (5-tuple), even though there is a freedom to define other features to form a flow [9]. A notorious difference regards to packet-independent treatment is that each feature is organized by feature keys (it is described in the next section).

For the rest of the paper, the flow-packet treatment is used. Each flow-packet has an inter-flow gap of 60 seconds. It means that between the first and the last packet the time should not be more than 60 seconds. By using these flow-packets, a methodology is proposed to build a behavior traffic profile of the computer network. The obtained profile helps to perform detection tasks. In the training phase, the flow-packets are organized as groups and represented by 3D entropy values by plotting a set of points. These set of points, seen as a point-clouds, can be treated by using Pattern Recognition. Feature extraction and selection can define a behavior model for normal traffic. After that, it is possible to use supervised classification to identify traffic slots with normal or anomalous traffic. Thus, an A-NIDS is proposed and described. The A-NIDS is able to analyze the network traffic by using packet features under a flow-packet traffic definition.

2. Analysis techniques

In this section we describe some of the techniques that are utilized for processing and analyzing a set of data in an efficiently way.

2.1. Pattern recognition

Pattern recognition (PR) is an analysis tool which allows classifying different objects according to their relevant features. In [10], PR process is described in detail. To understand how PR works, it is needed to describe some important concepts

- *Pattern*. It is a physical representation of a given object, case or sample.
- *Features*. It is a set of data or attributes, obtained from measurements of patterns. These features are useful for their further characterization.
- *Feature selection*. It is a process to determine which set of features is the most appropriate to describe a set of patterns according to their relevance. Feature selection has as the main goal to improve pattern speed processing and their accurate classification.
- *Feature extraction*. To reduce the amount of redundant data given in a set of features, such set should be transformed to obtain its reduced characteristic representation. It is also known as feature vector.
- *Feature vector*. It is the reduced characteristic representation of a given set of features. It stores the relevant data of such features and it is helpful for its further classification. For its easy management, these relevant data are represented in a vector form as shown in (1)

$$\mathbf{r} = [r_1, r_2, ..., r_d] \in R \subset \mathbb{R}^d, \tag{1}$$

where $\mathbf{r}$ is the feature vector and R is the domain of dimension d.

- *Feature space*. It is a given space where each pattern is a point in the space. Each feature vector is represented as a coordinate on such space. Thus, each axis represents each feature and the dimension d should be equal to the number of used features.
- *Class*. It is a set of values and attributes associated with concepts, objects and prototypes.

Given a pattern, its classification process can be done by using one of the following modes

- *Supervised classification*. A given pattern is identified as a member of a given predefined class.
- *Unsupervised classification*. A given pattern is assigned to an unknown and undefined class.

PR main task is to classify patterns in a set of classes, where these classes are defined by a human who designs a system (supervised) or are learnt and assigned by using pattern similarity (unsupervised). In this work in particular, a statistical PR approach (SPRA) is implemented. By using this approach, each pattern is represented by a set of features or measurements in a *p*-space. An adequate feature selection allows establishing disjoint regions. Thus, each class may be distinguished accurately as a different class. Such disjoint regions are obtained by using training sets and each disjoint region represents each class. By using SPRA, each decision region is determined by the probability density function (pdf) of the patterns belonging to each class either in supervised or unsupervised classification [11, 12].

2.2. Principal component analysis

In many cases, a dataset can be shown as a set of points in a *p*-space. Principal Component Analysis (PCA) [13 - 15] is a multivariable method which helps to extract geometric features of such set of points. A dataset is described by a set of correlated variables $(x_1, x_2, ..., x_p)$. Then, PCA helps to analyze that dataset by extracting its relevant information and transforming it as a new set of uncorrelated variables $(y_1, y_2, ..., y_p)$. Each variable is named as Principal Component. This set of variables is sorted by variance value and information that each one possesses. Each y_j, $j = 1, 2, ..., p$ is a linear combination of $(x_1, x_2, ..., x_p)$, then

$$y_j = a_{j1}x_1 + a_{j2}x_2 + ... + a_{j1}x_1 = \boldsymbol{a}_j^T \boldsymbol{x}, \tag{2}$$

where $\boldsymbol{a}_j^T = [a_{1j}, a_{2j}, ..., a_{pj}]$ is a row vector of coefficients, $\boldsymbol{x} = [x_1, x_2, ..., x_p]^T$ is a column vector of variables and *T* is the transposition operation.

To achieve the best possible representation, each variance for each coefficient in $\boldsymbol{a}_j^T$ should be maximized. Thus, $|\boldsymbol{a}_j^T| = 1$ or by using another equivalent representation, $\boldsymbol{a}_j^T \boldsymbol{a}_j = \sum_{k=1}^{p} a_{kj}^2 = 1$. It means that $\boldsymbol{a}_j^T$ must remain orthogonal to maintain its normalized form.

So, the first principal component is obtained from $\boldsymbol{a}_1$ in such manner that y_1 contains the highest variance value under the condition $\boldsymbol{a}_1^T \boldsymbol{a}_1 = 1$. The next principal component is obtained from $\boldsymbol{a}_2$ in such manner that y_2 should be de-correlated of y_1, and so on. Thus, the set of uncorrelated variables $(y_1, y_2, ..., y_p)$ sorted in variance-decreasing order is obtained.

2.3. Entropy

Entropy is used as a mathematical tool to measure the uncertainty of a dataset. Entropy has been studied from the point of view of many disciplines. This concept arisen firstly in physics to study the random movement of particles. Shannon, in his well-known reference [16] proposed the entropy concept as a

mathematical tool to measure the uncertainty of a communication source, thus generating the origin of the Information Theory. Shannon established that each communication source can be modeled as a source of uncertainty, and the uncertainty is described by a pdf. Clearly, some distributions present higher uncertainty than others. In the case of a discrete pdf $p(x_k)$, Shannon's entropy H of a discrete random variable X is given by,

$$H(X) = -\sum_{k=1}^{M} p(x_k) \log_2 p(x_k), \tag{3}$$

where M is the cardinality of the alphabet X and x_k, $1 \le k \le M$ are the elements of the alphabet. A property of (3) is $0 \le H(X) \le \log_2(M)$, given in bits.

2.4. Entropy space method

Entropy spaces are built by using traffic data-flows. A generated 3D space under typical traffic is used to define a behavior profile of network traffic. This method starts with the definition of a trace χ, which is divided in m non-overlapped traffic slots with maximum time duration of t_d seconds (by using static window-time). In a slot i, K_i flows of packets are generated. The flows are defined under a 5-tuple and *inter-flow gap* (IFG) of 60 seconds.

To identify each flow, four fields of each packet were extracted. Each field is labeled as follows: $r = 1$ for source IP address (*srcIP*), $r = 2$ for destination IP (*dstIP*), $r = 3$ for source port (*srcPrt*) and $r = 4$ for destination port (*dstPrt*). After that, the set of flows are clustered (clustering) for each slot i. This clustering process is done with respect to each r field, now named cluster key.

For a given *i*-slot, each cluster is formed by containing flows under the same r field value, but leaving freedom for the rest of r fields. As an example, for $r = 1$ or cluster key *srcIP*; each cluster is formed with all flows that contains the same source IP address field, regardless of the value at the rest of the fields $r = 2, 3, 4$. The rest of the fields are denoted as free-dimensions. It is clear that the complete number of clusters will depend of the cardinality of the alphabet $\left|\boldsymbol{A}_i^{r=1}\right|$, where $\boldsymbol{A}_i^{r=1}$ is the alphabet of source IP addresses shown in slot *i.* The same clustering process is applied for the rest of r fields.

Then, entropy estimation for each cluster key is represented as a 3D Euclidean coordinate, denoted as $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_k$. The number of points in the 3D space for each slot i depends of $\left|\boldsymbol{A}_i^{r=1}\right|$. Thus, the current data-points in a slot i are given by $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_1$, $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_2$, … $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_{|\boldsymbol{A}_i^{r=1}|}$. By plotting those points using scatter plot will form a *Point Cloud Data.* Applying the same method for the rest of r fields and *m* slots, the four entropy spaces of χ are obtained.

Fig. 1 shows entropy spaces by using srcIP field for three traffic traces. The traces were obtained from a LAN campus, where a network segment was leaved unprotected to the attacks of Blaster and Sasser worms. The first figure (1a) shows typical traffic for a trace of 8 hours during the first day of the week, and similar patterns were obtained for the rest of the weekdays. The second (1b) and third (1c) figure shows the behavior of Blaster and Sasser worms during 30 minutes of traffic capture, respectively. Fig. 1 shows that while typical traffic (1a) tends to be concentrated, the behavior of Blaster and Sasser (1b, 1c) tends to spread the points on the entropy space. The entropy spaces are represented by using the vector $\mathbf{X}^p \in \mathbb{R}^3$. PCA is applied to this vector to reduce the dimensionality of it and generate a new vector z-*score* $\mathbf{Z}^r \in \mathbb{R}^d$, $d \leq 3$.

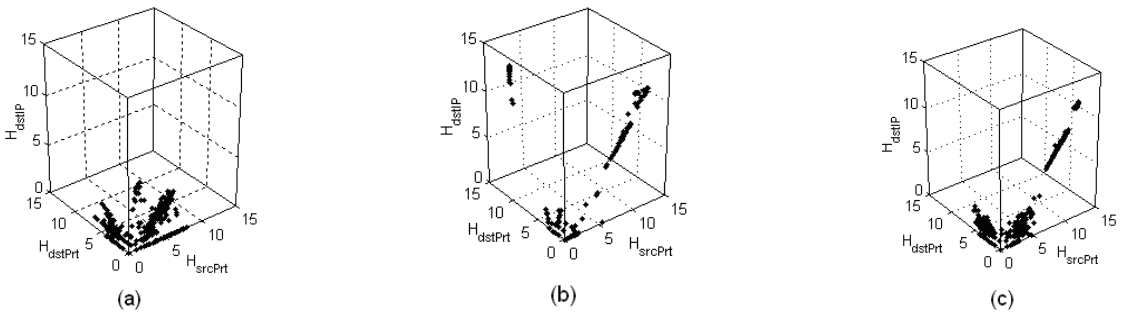

Fig.1. Entropy spaces for *srcIP*. (a) Typical traffic; (b) Blaster worm propagation; (c) Sasser worm propagation

3. Kernel density estimation

To obtain accurate analysis estimation, tools like Kernel Density Estimation (KDE) may be useful. This analysis was applied on data-points at the slot level on the PCA 1 by using a Gaussian kernel of 200 points, with a bandwidth of $h = 1.06\sigma J^{1/5}$, the Silverman's criteria, where J is the number of observations and σ is the standard deviation of the set of observations. KDE shows that the traffic slots have Gaussian bimodality behavior in its pdf. Each mode was labeled as principal mode and far mode respectively, as shown in Fig.2.

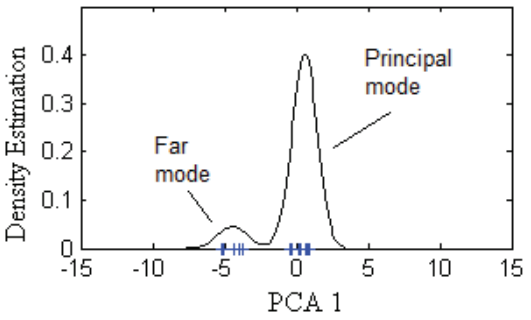

Fig. 2 Density estimation of PCA 1 presents bimodal behavior on an *i* traffic slot.

For PCA 1, the empirical obtained media values were among 4.3 (positive far mode) and –4.2 (negative far mode) units of PCA 1. Negative far mode was the most frequent. In the case of studied attacks, far mode presented anomalous values on slots with anomalous traffic. For instance, in the case of Blaster worm the three first slots showed values of –9 , –11 and –13 PCA 1 units. These values were classified as an anomaly because of each value is lower than the media and the threshold value (–4.2). Thus, this anomalous behavior can be detected easily in an early stage.

A second feature that was used because of its sensitive behavior is the standard deviation of principal mode. The empirical obtained standard deviation in typical conditions was 1.5 units. Then, standard deviation of principal mode was taken to show anomalous behavior. For instance, in the case of Sasser worm, the standard deviation of principal mode on anomalous slots decreased to 0.4 units in average. In another tests but detecting a port-scanning attack, the standard deviation shows 0.7 and 0.4 units on two anomalous traffic slots.

Nevertheless, KDE only provides the density distribution shape but not its parameters. Therefore, a technique to extract these parameters (media of the far mode and standard deviation of principal mode) is needed. It is important because these parameters can be utilized in an A-NIDS. In the next subsection, an auxiliary technique will be described and utilized to obtain these parameters.

4. Gaussian Mixture Model

As it was described in the aforementioned subsection, an extra technique should be utilized to extract KDE parameters. To achieve this goal, Gaussian Mixture Model (GMM) is used [17]. A GMM implementation can be found in Statistic Toolbox of MATLAB. *gmdistribution.fit* toolbox forms groups of data and then fit them on a GMM model of K components. The fitting process is based on *Expectation-Maximization* (EM) algorithm, which is based on maximum likelihood. The algorithm returns the parameters $\theta_k = [\pi_k, \mu_k, \sigma_k]_{k=1}^{K}$, where $K = 3$ is the number of modes of the estimated pdf under KDE for PCA 1; π_k are the proportions of the mixture and μ_k, σ_k are the media and standard deviation that are related to the required information about principal and far modes.

By using PCA 1 and *srcIP* key, we obtain the mixture vector θ_k . From θ_k , principal and far modes can be identified on an *i* slot. Principal mode corresponds to *k* mixture, which has $\mathbf{max}(\pi_k)$ (maximum proportion). Far mode corresponds to mixture with $\mathbf{max}(|\mu_k|)$ (maximum media).

For classification tasks, feature selection chooses far mode media parameter r_1, and the standard deviation of principal mode r_2 . The behavior of the aforementioned features allows establishing that they are good parameters to identify anomalous and typical traffic. Experiments described throughout this document support the aforementioned assumption.

In Fig. 3, typical traffic analysis shows that the behavior of feature r_1 is tri-modal shape, so the GMM description is based on three components.

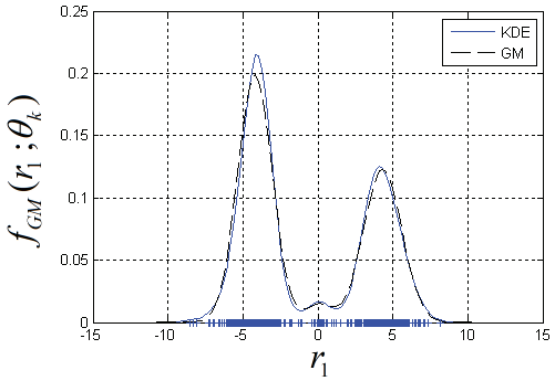

Fig.3. Fitting r_1 feature by using GMM.

As we describe above, for each feature there are 3D free-dimensions that describe the behavior of the network traffic. By using these three free dimensions, a model based on GMM can be constructed. In the case of feature r_1, the profile behavior model obtained is

$$f_{GM}(r_1;\theta_k)=\sum_{k=1}^{3}\pi_k \mathrm{N}\left(r_1 \mid \mu_k,\sigma_k^2\right)=\sum_{k=1}^{3}\frac{\pi_k}{\sqrt{2\pi}\sigma_k}e^{-\frac{(r_1-\mu_k)^2}{2\sigma_k^2}}, \tag{4}$$

where μ_k y σ_k is the media and standard deviation of *k*-esim component, respectively; and π_k are the mixture proportions. The three components form a 3-component vector $\theta_k=[\pi_k,\mu_k,\sigma_k]_{k=1}^{3}$. $\mathrm{N}\left(r_1 \mid \mu_k,\sigma_k^2\right)$ represents a Gaussian multivariate distribution. The fit values of this mixture are shown in Table 1.

Table 1. θ_k parameter values

Parameter	$k=1$	$k=2$	$k=3$
Mixing (π_K)	0.3946	0.5740	0.0314
Mean (μ_k)	4.3239	–4.1988	0.0980
Variance (σ_k^2)	1.6378	1.3208	0.7546

In the case of feature r_2, analysis on the set of typical traffic traces showed that their behavior followed a *Generalized extreme value distribution,* (*GEV*), with the following profile behavior model

$$f\left(r_2;\mu,\sigma,\xi\right)=\frac{1}{\sigma}\left[1+\xi*\left(\frac{r_2-\mu}{\sigma}\right)\right]^{-1/\xi-1}*\exp\left\{-\left[1+\xi*\left(\frac{r_2-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \tag{5}$$

for $1+\xi*\left(x_2-\mu/\sigma\right)>0$, where $\mu\in\square$ is the localization parameter, $\sigma>0$ is the scale parameter and $\xi\in\square$ is the shape parameter.

Fitting parameters for (5) which describe the behavior of r_2 are $\xi=0.2228$, $\sigma=0.1221$ and $\mu=0.9079$. The obtained shape with these values is shown in Fig. 4.

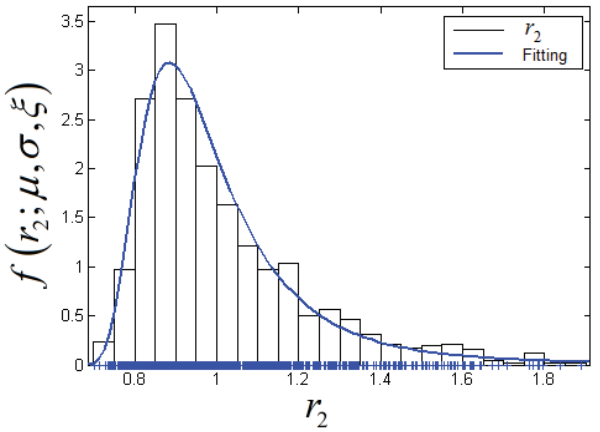

Fig. 4. Fitting curve of feature r_2 by using a GEV distribution

5. Framework architecture

In Fig. 5 an anomaly-based NIDS framework architecture is shown. This framework implements behavior profiles based on entropy spaces. It can be divided in two layers. The first one does training functions with the selected features (r_1 and r_2). In this layer a behavior profile model is obtained by using probabilistic models (eq. 4 and 5). The second one performs detection functions by using the feature values that belongs to each window of traffic under analysis. Then, each window of traffic is compared with typical traffic profile to determine whether an anomaly appears. Each block which forms the framework is described as follows in the case of pre-training layer

- **Pre-processing:** In this stage each typical traffic trace is sanitized. Such process is done to delete duplicated or mis-captured packets. The trace is filtered according to the protocol.
- **Flow generator:** On each traffic slot a flow is generated according to 5-tuple definition.
- ***Flow clustering.*** The generated flows are clustered according to a feature. The number of clusters will depend of the cardinality of the alphabets for each *r* feature.
- **Entropy quantification:** For each free dimension obtained on each *r* feature, entropy value is estimated by using naïve entropy estimation.
- **Dimension reduction:** The set of entropy data-points is transformed by using PCA. From the output of PCA, PCA 1 is chosen for srcIP key. By using this key the features are chosen.
- **Feature identification:** This stage identifies and obtains the features on each PCA 1 by using GMM. It will help to create a profile.

In the case of the detection layer the process is almost the same, with the only difference that a profile is not generated. Instead of this, from each captured traffic slot and after passing the aforementioned listed stages, the PCA 1 features are extracted by using GMM and after these features are compared against typical traffic behavior according to a probabilistic model (as described in the last section).

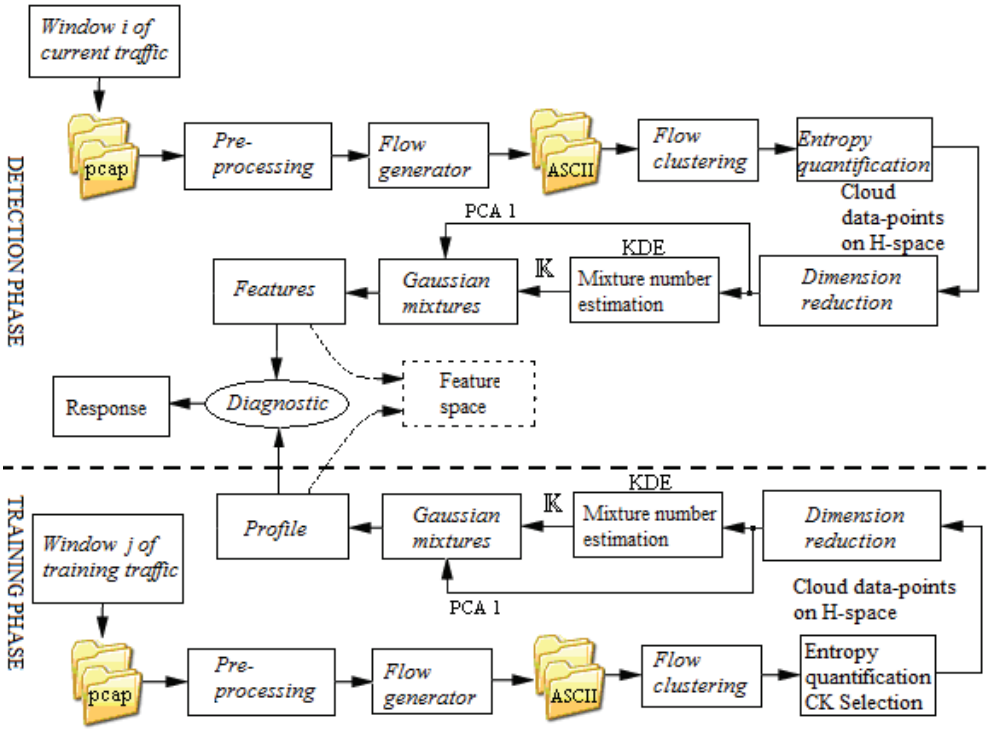

Fig. 5. Diagram framework for each A-NIDS module

As it was shown throughout this document, an anomalous behavior changes the traffic behavior and it can be measured and compared. Thus, the framework, seen as NIDS, can be able to warn about anomalous activities by measuring and comparing deviations against typical values. It is done on each slot of the trace and its prevention depends on the window time size.

Entropy space is formed by data-points, obtained from each traffic feature on each slot. For instance in the case of srcIP feature, the data-points represent the multiple entropy values taken for the rest of the dimensions for the flow clusters. The number of data-points is defined by the number of different srcIP that are in the slot. The first PCA simplifies the analysis of the given dimensions. Two features on the first component can show clearly when an anomalous behavior arises.

Fig. 6a shows the two types of traffic, normal and anomalous traffic produced by Blaster. The former is grouped in three different regions and the latter only in two. It can help to apply supervised classification techniques by dividing their corresponding classes. In Fig. 6b traffic distribution is shown for the feature r_1 on both cases normal and anomalous traffic produced by Blaster worm.

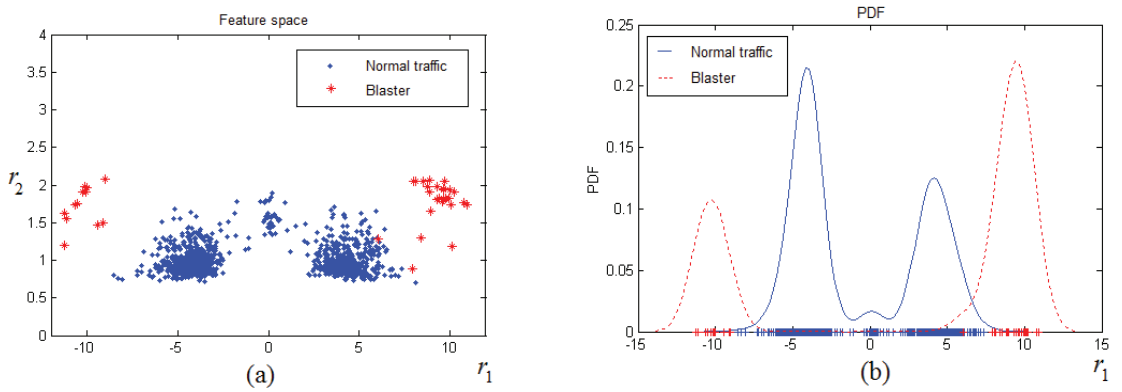

Fig. 6. (a) Feature space for normal traffic and Blaster. (b) r_1 feature distribution for normal traffic and Blaster.

In the case of Sasser worm, traffic slots are grouped in one region as shown in Fig 7a. In Fig. 7b it is shown the difference between normal traffic and anomalous traffic produced by Sasser worm in the case of the feature r_2. For all the aforementioned cases, the regions generated for each type of traffic can be clearly divided.

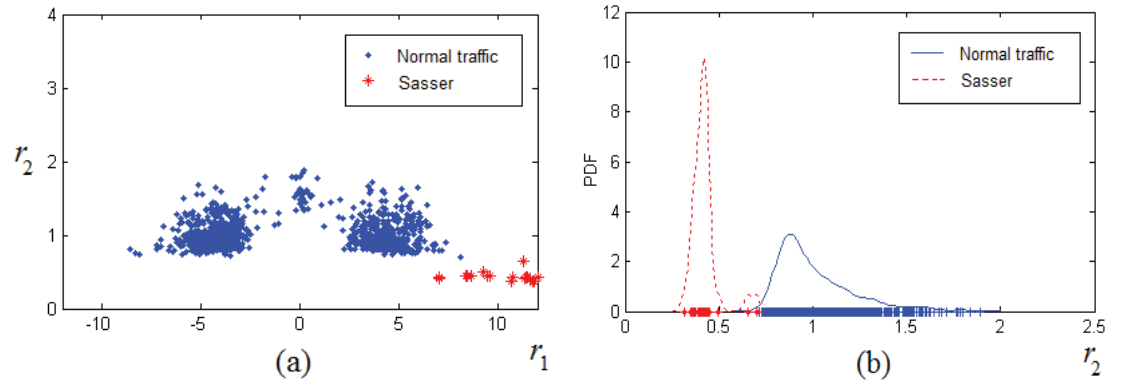

Fig.7. (a) Feature space for normal traffic and Sasser; (b) r_2 feature distribution for normal traffic and Sasser

6. Conclusions

In this work we presented an A-NIDS working on flow-packet level. Techniques such as Entropy Spaces, Pattern Recognition and fitting models were used. Their use allows identifying special features that helps to define anomalous traffic in specific slots and comparing them with normal traffic. Anomalous traffic presents different values than normal traffic in specific parameters in the case of fitting models given by the Gaussian Mixture Model. Mean and variance shows different values when anomalous traffic arises, defining different regions according to the results shown in Figs. 6 and 7. By using these results, framework architecture is proposed to analyze network traffic anomalies. Deviations from a given reference obtained from typical traffic are considered anomalous. The further work contemplates to consider scenarios with high volume of traffic, the evaluation of more attacks and adding a pre-processing stage by using an S-NIDS like Snort. The last issue implies to filter well-known attacks to obtain features that represent normal traffic in a better way.

Acknowledgements

P. Velarde-Alvarado and A. Iriarte-Solis are grateful with PROMEP for financial support given for this research. A. F. Martinez-Herrera is very grateful with CONACyT and ITESM for supporting him to pursuit his PhD at ITESM, Monterrey Campus.

References

[1] Lillard TV. Digital forensics for network, Internet, and cloud computing: a forensic evidence guide for moving targets and data. Elsevier Inc.; 2010

[2] Roesch M. Snort - Lightweight Intrusion Detection for Networks. *Proc. 13th USENIX Conference on System Administration, USENIX*; 1999.

[3] Wang K, Stolfo S. Anomalous Payload-Based Network Intrusion Detection. *Recent Advances in Intrusion Detection*, Springer Editors 2004:203 – 222.

[4] Xu K, Zhang Z, Bhattacharyya S. Internet Traffic Behavior Profiling for Network Security Monitoring. *IEEE ACM T Network* 2008; 16-3:1241 – 1252.

[5] Wagner A, Plattner B. Entropy Based Worm and Anomaly Detection in Fast IP Networks. Proc. of the 14th IEEE International Workshop on Enabling Tech 2005: 172 – 177.

[6] Velarde-Alvarado P, Vargas-Rosales C, Torres-Roman D. Martinez-Herrera A. IP Traffic Anomaly Exposure, an Information Theoretic-based Approach. *J App. Res Technol* 2010; 8-2:159-176.

[7] Velarde-Alvarado P, Vargas-Rosales C, Torres-Roman D, Martinez-Herrera A. Detecting Anomalies in Network Traffic Using the Method of Remaining Elements. *IEEE Comun Lett* 2009; 13- 6:462-464.

[8] Feinstein L, Schnackenberg, D, Balupari, R, Kindred, D. Statistical approaches to DDoS attack detection and response, *DARPA Information Survivability Conference and Exposition (DISCEX 2003)* 2003:303-314.

[9] Malagón C. NetFlow y su aplicación en seguridad. *Ed. Red.es/RedIRIS*. 2009. Boletín No. 87:33-42

[10] Marques de Sa JP. *Pattern Recognition: Concepts, Methods and Application*.1st ed. New York: Springer-Verlang; 2001.

[11] Devroye L, Gyorfi L, Lugosi G. *A Probabilistic Theory of Pattern Recognition*. Berlin: Springer-Verlag; 1996.

[12] Duda RO, Hart PE. *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons; 1973.

[13] Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. Englewood Cliffs New Jersey: Prentice Hall; 1992

[14] Härdle W, Hlávka Z. *Multivariate Statistics: Exercises and Solutions*. New York: Springer Science+Business Media LLC; 2007

[15] Bishop CM. *Pattern Recognition and Machine Learning*. Springer Science+ Business Media LLC; 2006

[16] Shannon JC. A Mathematical Theory of Communication. *AT&T Tech J*. 1948; 27: 379-423.

[17] McLachlan GJ, Peel D. *Finite Mixture Models*. Wiley;2000