An unsupervised approach for traffic trace sanitization based on the entropy spaces

Pablo Velarde-Alvarado[1] · **Cesar Vargas-Rosales**[2] · **Rafael Martinez-Pelaez**[3] · **Homero Toral-Cruz**[4] · **Alberto F. Martinez-Herrera**[2]

Published online: 31 March 2015

Abstract The accuracy and reliability of an anomaly-based network intrusion detection system are dependent on the quality of data used to build a normal behavior profile. However, obtaining these datasets is not trivial due to privacy, obsolescence, and suitability issues. This paper presents an approach to traffic trace sanitization based on the identification of anomalous patterns in a three-dimensional entropy space of the flow traffic data captured from a campus network. Anomaly-free datasets are generated by filtering out attacks and traffic pieces that modify the typical position of centroids in the entropy space. Our analyses were performed on real life traffic traces and show that the sanitized datasets have homogeneity and consistency in terms of cluster centroids and probability distributions of the PCA-transformed entropy space.

✉ Pablo Velarde-Alvarado
pvelarde@uan.edu.mx

Cesar Vargas-Rosales
cvargas@itesm.mx

Rafael Martinez-Pelaez
rafael.pelaez@uacj.mx

Homero Toral-Cruz
htoral@uqroo.edu.mx

Alberto F. Martinez-Herrera
af.martinez.phd.mty@itesm.mx

[1] Area of Basic Sciences and Engineering, Autonomous University of Nayarit, 63155 Tepic, Nayarit, Mexico

[2] Department of Electrical and Computer Engineering, Tecnológico de Monterrey, Campus Monterrey, 64849 Monterrey, Nuevo Leon, Mexico

[3] Department of Information Technology, Autonomous University of Ciudad Juarez, Chihuahua, Mexico

[4] Department of Sciences and Engineering, University of Quintana Roo, 77019 Chetumal, Quintana Roo, Mexico

Keywords Sanitizing traffic data · Entropy-based anomaly detection · Data mining · *k*-means clustering · Principal component analysis

1 Introduction

One of the most important challenges in computer networks and information systems is the security threats. Recently, there has been a significant incidence of new advanced threats and an increased level of sophistication in the attacks [39]. To address these threats effectively, sophisticated security solutions need to be implemented, for example, robust authentication protocols [20], security in the cloud [2], next-generation perimeter defense [30], etc. For perimeter protection, network intrusion detection systems (NIDS) are often used as additional layers to protect networks by analyzing traffic for suspicious activity, either internal or external. Intrusion detection (ID) is the intelligent process of monitoring and analyzing the events occurring in a computer system or network, to detect signs of violations of security policies [4]. ID approaches can be categorized into signature-based detection, and anomaly-based detection. A signature-based NIDS (S-NIDS) examines the network traffic for patterns of known intrusions. A key advantage of this detection method is that it can accurately and efficiently detect instances of known attacks. However, there are two intrinsic weaknesses of S-NIDS methods. The first is the inability to detect zero-day attacks or polymorphic attacks, either because the database is out of date or because no signature is available yet. The second disadvantage has to do with the time lapse to create signatures for new attacks. An alternative to S-NIDS is the anomaly-based NIDS (A-NIDS), that works on the premise that malicious network traffic is distinguishable from normal traffic, [15,19]. Thus, the typical behavior of the network is captured and modeled, and deviations from the established behavior profiles are interpreted as anomalies. The major advantage of A-NIDS over S-NIDS is that a zero-day attack and variations of attacks for which a signature does not exist can be detected if these fall out of the behavioral profile. Although the capabilities of A-NIDS are significant, they suffer of some drawbacks as well; for example, they tend to generate more false alarms because an anomaly can just be a new behavior generated by the dynamic nature of the network. Moreover, due to the high computational requirements of some anomaly detection algorithms, on-line processing is not suitable in high speed networks [21]. Flow-level analysis is a suitable solution to address scalability problems posed by the high speed networks available today. In [38], the authors propose a metric to detect flow disturbances, e.g., abrupt increases in traffic caused by targeted attacks, for this, a limited number of monitors must be located within the stochastic network to assess communication-flow patterns.

Theoretically, the combination of different ID approaches in one system can produce a more robust NIDS. Two examples of hybrid systems are shown in [3,32].

Data mining techniques improve A-NIDS performance by addressing problems related to detection of sophisticated or novel attacks, data overload, and false positives/negatives. In this sense, Nikolova and Jecheva in [35] proposed a methodology to form normal activity profiles for anomaly-based IDS.

Soft computing techniques have been widely used in the field of intrusion detection [26]. In [1], Ahmad, et. al., use a combination of tools such as principal component analysis (PCA), Genetic Algorithms (GA) and Support Vector Machines (SVM) to improve the detection rate of IDS to 99.6 %.

In this paper, we introduce a system to generate training and benchmark traces for A-NIDS by using traffic trace sanitization applied to raw traffic traces to remove anomalous traffic. The resulting trace improves quality for a training stage of A-NIDS and captures network dynamics. In our research, we apply a data mining technique, k-means clustering, to identify and filter out anomalies in the entropy space of the network traffic. Additionally, the proposed system is supported with statistical tools such as PCA, Kernel Density Estimation (KDE), and Mahalanobis distance (MD). We also show that the application of our method, allows one to generate new databases consisting of attack-free traces from real traffic captured on the same network, which can further be used for the design, testing or training of NIDS specific for the network segment from which the traces were generated. Also, the use of attack-free traces for network characterization and behavior evaluation allows the practitioners to use simulation methods to evaluate new techniques that require testing before being implemented in the actual network. Thus, a solution to the scarcity of suitable dataset for purposes within the field of NIDS is provided, and the resulting NIDS methods will be intrinsically related to the network on which they are implemented. The overall objective of the proposed method is to perform traffic trace sanitization using information theory, unsupervised techniques and statistical tools for the verification of results, thus generating new and better datasets that could help for network intrusion detection purposes and for building behavioral profiles of the network segment analyzed.

The paper is organized as follows: In Sect. 2, we briefly discuss the related work. Section 3 provides a thorough and detailed description of the framework for the traffic trace sanitization methodology we propose. In Sect. 4, the data sets and platform used in the experiments are described. Section 5 contains the experimental results that show the functionality of the methodology proposed, and at the end, Sect. 6 presents the conclusions highlighting important results and possible future use of our findings.

2 Related work

Usually, behavioral profiles have to be learned from days or weeks of anomaly-free traffic traces, this is a practical problem since the training traffic dataset is never guaranteed to be clean when collected from a real network [6,7]. Also, it is evident that the definition of normal behavior plays an important role in the performance of the A-NIDS. It is possible for an A-NIDS to achieve better performance minimizing the false alarm rate and maximizing the detection accuracy using behavioral profiles that represent the legitimate network traffic [5]. The key point to build behavioral profiles representing accurate typical behavior is to have valid datasets, i.e., anomaly-free traffic traces. Traffic datasets may come in two ways: synthetic and real data. The most commonly used synthetic datasets and publicly available are the 1999 KDD Cup dataset and MIT-DARPA evaluation dataset that have been used to test a large number of intrusion detection systems. The MIT-DARPA dataset has been criticized for being a very outdated dataset, unable to accommodate the latest trend in attacks, for example, botnets, spyware, SQL injection, and network worms which are contemporary [42]. Additionally [45], this dataset has the lack of damaged or unusual background packets and uniform host distribution. On the other hand, real traffic traces can be obtained through public repositories like MAWI, NLANR and, CAIDA. A drawback associated with the traffic traces provided by these repositories is their strong *anonymization* of the information, which is a procedure that removes sensitive information from data before being released and put available to the public. Typically, traffic trace anonymization involves the hashing or elimination of various fields from the trace captures. The problem with anonymization is that not only it removes privacy-sensitive information from the traces, but also information which is important and valuable for research. The stripping of information discards vital properties of the captured traffic, rendering the traces unusable for certain network studies; specifically for those with decisions based on traffic feature distributions or stateful analyses. The traffic feature distributions provide an approach to conduct fine-grained network traffic analysis which, compared to volume-based approach, is most effective for intrusion detection tasks. Stateful analyses need to preserve context information on any stream of flows, protocols or packets being analyzed. Additionally, there is a risk associated with anonymization, which consists of attacks that are based on correlated external information with anonymized data and successfully de-anonymized objects with distinctive signatures [8]. In terms of applications such as traffic engineering, financial requests and movements, among others, anonymization must not be used because vital information for decision making processes can be lost.

Due to limitations imposed by the scarcity of suitable datasets for the implementation and evaluation of intrusion detection systems, some studies have chosen to create their own real traffic datasets which guarantee the reliability of the obtained results, [8]. The behavioral profiles constructed based on real traces offer the advantage of reflecting the current trends in the attacks and the actual dynamics of a network, however, an advanced preparation on these datasets is needed in order to remove attacks and reduce the overall noise level of intrusions. This phase performed to remove suspected attacks and other abnormalities from data, and to produce anomaly-free traces is called *traffic trace sanitization*. In traffic analysis, the term *sanitization* can be used in two contexts. One is to perform anonymization to remove private information, and the other is used to remove spurious data. It is the latter that is relevant to our research focus.

Very little research has been conducted on dataset sanitization or dataset generation, especially for intrusion detection purposes. Trace sanitization general principles and characteristics that should be satisfied can be found in [37]. In [17], trace sanitization is used for traffic classification and monitoring, where the IPMON system is proposed and distribution of flows, packet delays and packet sizes are studied. In [47], authors used traffic trace sanitization for an intrusion warning system using message flows. Sanitization is conducted and clusters are formed by port numbers, the system is used to produce warnings for worm attacks. No information theory is used and no production of clean traffic traces for network characterization is carried out. A training dataset sanitization technique for multiple anomaly detection sensors is proposed in [12]. Using a voting scheme, abnormal packets are filtered out from the training dataset. However, because it is a packet-level solution, its implementation is not feasible for large scale networks. In [43], the authors proposed a method for generating network traces in a controlled testbed environment. The attack environment is defined by an α-profile, while a network behavior is encapsulated by a β-profile. These profiles are guidelines that allow the representation of features and events that facilitate the reproduction of real-world behaviors as seen from the network. However, the generated traces are dependent on such profiles, which only capture the behavior of the real network where analysis was performed. In the real world, each network is exposed to different threats; therefore a generalization obtained from a single network has its drawbacks.

Recently, the sanitization process has been reported in most of the available architectures. For instance, patents US 20140283052 A1, and US 8,407,160 B2 [13,22], respectively, include a sanitization module. In US 8,407,160 B2, the sanitization process is included in an IDS, as might be expected, before the generation of the traffic modeling. Here, a previous training should be performed in order to enforce accurate detection rates. In patent US 20140283052 A1, two typical models such as signature-based (known attacks) and machine-learning-based (zero-day attacks) are combined into a single IDS. The combination of both techniques allows the IDS to mitigate false-positives. Nevertheless, this IDS also depends on previous training. A specific application is presented by Wressnegger et. al., in [49], to sanitize HTTP queries of known attacks before modelling the typical behavior of such queries. Other examples such as Narang et. al. [33,34] use sanitization to remove those IP packets which contain an invalid IP header. Gang et. al. [18], show how the sanitization process is critical for generating a reliable signature. After sanitizing, the training process they describe constructs 5-tuples of uniflows or biflows from TCP/UDP traffic. Then, they are able to classify those uniflows or biflows that share the same patterns (for instance, the same payload size, the same IP source, etc.). Each set of grouped flows generate one signature, thus downsizing the process of inspecting network traffic. Chen et. al. [9], use sanitization to remove port-scan activity and they also evaluate how the removed packets affect the behavior of the IDS on training and traffic traces. The IDS is mainly based on “the random moonwalk algorithm” (RMW). They claim that, with some limitations, RMW is able to detect attacks with evasion techniques. They also claim that in some cases the improvements are significant with the sanitization process.

In this paper, we present a novel method that deals with the problems just discussed by introducing a proof of concept that shows the feasibility of the design of a system in which a network generates its own training and benchmark traces for A-NIDS.

3 A flow-level approach to traffic trace sanitization

By using flow-level analysis, no information related to individual packets is captured, but flows of packets. Flow-level models provide a different perspective about the traffic, because the network is considered on a higher level of abstraction, i.e., over a longer time horizon. In a flow-level model, the fundamental unit of information is the flow. A flow is a unidirectional stream of packets of a given protocol from a source IP address and source port to a destination IP address and destination port in a given time interval of size ΔT. The above definition is known as quintuple (5-tuple) for the five "*flow keys*" they share. Two additional flow keys may be the Type of Service (ToS) and the input interface, in fact, different parameters can be used to define a flow.

The flow keys and flow statistics are stored in a *flow record*. Practical implementations for generating flow records can be obtained through the captured data in routers running protocols such as Cisco NetFlow [10], IPFIX [11], or J-Flow, [23]. In this work, a Perl script called `flowanalyzer.pl` was developed, which is an exporter that generates flow records in 5-tuple from the received pcap binary file, i.e., the traffic trace.

Flows can only provide information on the behavior of the connection and not on the payload of the packet. Therefore, attacks that are detected by only analyzing the contents of the packets, in other words through signature matching, are not detectable at the flow level. Such is the case, for example, of SQL injection attacks (SQLi) targeting Web applications. This type of attack is invisible at flow level because the malicious code is contained in the payload packet. However, this might be detectable at the flow level if the attacker executes multiple parallel attacks, because many connections to the targeted server are generated. Although flow level has limited information about network interactions, there is sufficient information to identify patterns among hosts. For this reason, it is important to develop anomaly-based algorithms at the flow level that can be effective in detecting denial of service attacks (DoS), network worms, scanning attacks and floods, or any other type of attack that alters the dynamics of the connections.

The processing and storage costs incurred by network devices become a critical issue as the volume and speed of data increases. Packet-based NIDS are very demanding in time, so they should not be used in high-speed links, except where specialized hardware is used in a given network link. Furthermore, these devices are quite expensive. On the other hand, the amount of space needed to store traffic traces is usually huge and often have prohibitive costs. This huge size affects the efficiency for accessing such a database for analysis of signatures. Another problem facing packet-based schemes in high speed links is that in most cases signature analysis is impossible where the payload is encrypted, which affects the performance of S-NIDS. As we just discussed, the packet-level methods mostly apply to signature-based detections, therefore, such problems affect S-NIDS.

To handle these problems, new flow-based algorithms have been developed looking for the reduction of the volume of data to analyze. Flow-based methods only need to monitor and process a fraction of the amount of data that requires a network packet-based method to about 0.1 % [46]. Usually, the flow records are generated by the exporter, which does not overload the NIDS with computational cost. Because network data at flow-level is lightweight, storage problems that occur at the packet-level can be mitigated. Also, the problem of encrypted payload does not influence the operation of flow-based NIDS. On the contrary, the absence of the payload analysis contributes some advantages for flow-based methods, such as the scalability of high-speed networks, which is a very important aspect to meet the demands that networks face today.

Finally, the supervised techniques present some drawbacks related to their adaptability. Since the nature of computer networks is not static, a trained NIDS will become useless if it is not updated with new traffic information, because traffic properties will change as soon as the network itself, applications, services or protocols, change. This implies that a NIDS should update its own anomaly detection policies in order to mitigate false positives. Additionally, these techniques need time to create new rules and baselines, even though some techniques claim that such period of time and the needed data are mitigated by using auxiliary processing techniques such as machine learning, etc. to help in the classification process [27]. On the other hand, "zero-day" attacks become risky when the NIDS policies are not updated. For instance, an attacker could use evasion techniques to hide damaging attacks in "benign" traffic [27].

In the method proposed in this paper, the entropy is used to perform an abstraction that maps flow-level traffic into a spatiotemporal representation called entropy space. This entropy space is characterized by an unsupervised technique (k-means clustering) which allows the identification of traffic with regular or anomalous behavior. Filtering the portions tagged as anomalous traffic is the basis for the method of traffic trace sanitization. Verification of sanitized datasets under this method is observed by the homogeneity in the centroid position and the probability distribution of the principal components in entropy space. The application of this method within a closed environment, referring to the protection of privacy, enables the generation of new databases of attack-free traces from real traffic captured on the same network, which can be used for the design, testing or training of NIDS. Thus, we provide a solution to the scarcity of suitable datasets of sanitized traces for purposes within the field of NIDS, since such traces would describe typical behavior of the segment being monitored and do not come from unknown segments with different typical behaviors.

3.1 Requirements of a flow-level approach

The overall objective of our proposed method is to perform traffic trace sanitization using Information Theory, unsupervised techniques and statistical tools for the verification of results, thus generating new and better datasets for network intrusion detection purposes.

We recommend that any flow-level approach, which intends to carry out trace sanitization, satisfy certain requirements that are listed in the following

(1) *Economy and simplicity* The method is able to generate useful datasets from the same network being monitored, it is not necessary the implementation of a testbed. This reduces infrastructure costs and time to get results.
(2) *Scalability* It is suggested that the method uses approaches to reduce the amount of information generated by the traffic traces in order to be used on high speed links and dense network segments. The flow-level traffic analysis approach is suitable for high-speed networks.
(3) *Fine-grained analysis* It is recommended that any approach used for sanitization have the adequate sensitivity in order to capture malicious behavior that could be very subtle. Our method carries out traffic characterization based on measurement of entropy, which provides greater sensitivity to reveal anomalous behavior caused by sophisticated attacks.
(4) *Unsupervised* This characteristic is recommended for any approach to be able to detect malicious behavior without a-priori knowledge of the features of such behavior. It is recommended that the approach use unsupervised algorithms to allow the detection of anomalies in noisy and unlabeled data, and 0-day attacks.

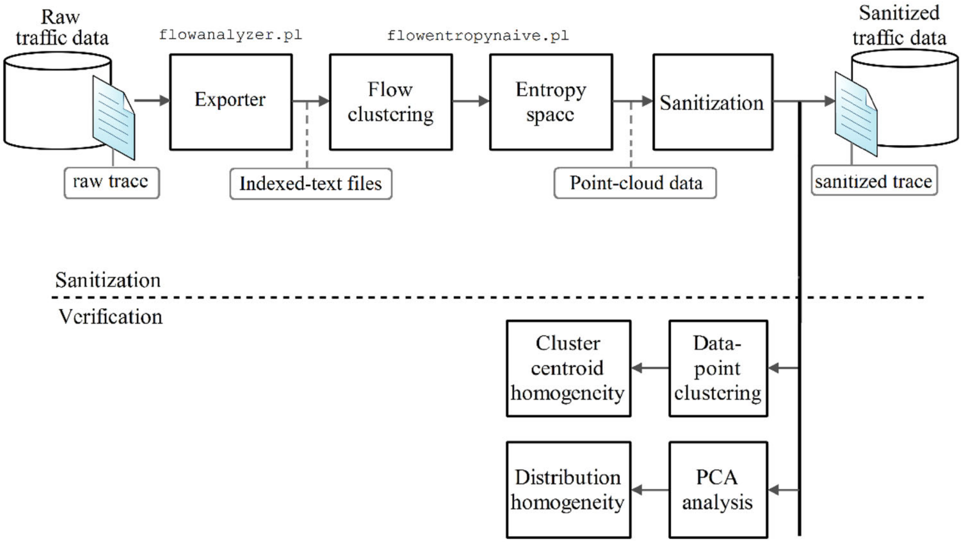
The approach presented in this paper has all these requirements satisfied.

3.2 System architecture proposal

Based on the requirements just listed and the problems discussed in previous sections, an architecture for traffic trace sanitization method is introduced in this section. The proposed architecture is shown in Fig. 1. The main input is the database of raw traffic traces from the network, and the output is a database that contains the sanitized and clean traffic traces that result from our method, hence the complete traffic trace sanitization architecture works offline. The architecture consists of two main layers: sanitization layer and verification layer. The sanitization layer processes raw binary traces in order to represent them through an entropy space. From this abstraction of entropy spaces, the sanitization layer also identifies and filters out portions of anomalous traffic to generate a new sanitized trace. The second layer, the verification layer, confirms that the sanitized trace properties are similar or within certain statistical criteria based on two behaviors: homogeneity in cluster centroids and uniformity in the distribution, we discuss these criteria after presenting in detail the method proposed.

The Sanitization layer starts by receiving a raw traffic trace captured in binary format (pcaplib), which is further cut up using `tracesplit` (from libtrace) for a given elapsed time of 8 hours between 7:00 and 15:00 h. The trace is filtered by selecting only those packets that have TCP traffic. The resulting traffic trace is fed to the *Exporter block*, which creates the flow records by partitioning the trace into traffic slots of duration no longer than $t_d = 60$ seconds. In each

Fig. 1 Block diagram of the traffic sanitization engine



slot i, flows are generated according to the definition of 5-tuple, and its corresponding flow records are stored in an i-indexed text file. A flow record contains the following flow keys: start time, and end time of the traffic flow, source IP address, source port number, destination IP address, destination port number, the number of packets and the amount of bytes in the flow. The contents of the file are used by the *Flow clustering block* to create cluster flows of the i-th traffic slot. Clustering is performed according to a pre-defined cluster key or pivot. Subsequently, information of the clusters flows generated in the i-th traffic slot is sent to the next block, the *Entropy space block.* This entropy space block is responsible for making the spatiotemporal representation of traffic, where Shannon's entropy is estimated in each cluster flow. The defined pivot determines the three flow keys on which the entropy is estimated. Therefore, a 3-dimensional value of entropy is generated. This value is called *entropy data-point* and can be represented graphically in a three dimensional space called *entropy space*. Thus, subsequent cluster flows originate a *cloud of data-points* which represents the traffic trace. Finally, the *Sanitization* block is responsible for the analysis of the data-point clouds using an unsupervised technique, namely k-means. Also, criteria such as the Mahalanobis distance or empirical selection for outlier detection is employed. Such outliers generally correspond to anomalous behavior and this anomalous behavior can be verified by forensic techniques since information of IP addresses, and ports permits the identification of the intruders and the victims. With the information provided by the outliers, it is possible to determine the guidelines for anomalous traffic filtering. Anomalous behavior will present itself by a variation of the clouds of data points within the entropy space. This variation is captured by the Mahalanobis distance with the criterion of the 98-th percentile for typical behavior and the ones left out are considered outliers. The reason for this is that anomalous behavior affects the natural disorder of the network segment and this is captured by the entropy with the clouds of data points, see Sect. 3.4.

In the *Verification* layer, two types of testing on sanitized traces are carried out. The first test (Cluster centroid homogeneity) analyzes entropy space of a sanitized trace by the position of a number of centroids obtained from the point clouds of the traffic trace. The number of centroids is determined by Mojena's rule [31]. The locus used by the centroid of sanitized traces is homogeneous and completely divergent regarding raw traces with anomalous traffic. In the second test (Distribution homogeneity) the entropy space of a sanitized trace is analyzed by PCA, in this case, the probability distributions of the first and the second principal components are homogeneous, while for the corresponding distributions of raw traffic with anomalous traffic, it is not possible to observe statistical similarities.

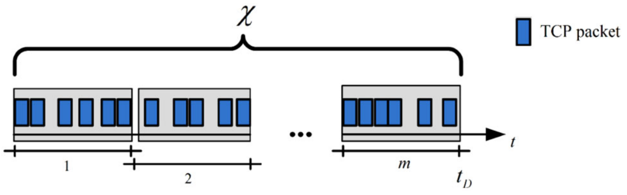


Fig. 2 Partitioning of trace χ in m traffic slots

In the next section, we explain the procedure for traffic abstraction based on entropy estimation of cluster flows.

3.3 Traffic abstraction by entropy spaces

Methods based on entropy of traffic feature distributions for anomaly detection provide fine-grained insights [25,36,48,50]. Through the use of entropy, it is possible to extract the properties of feature distributions and reveal unusual patterns and trends in the traffic behavior that are hidden to traditional methods based on volume.

Entropy is a functional which measures the information content of a dataset or the uncertainty of a random variable. Given a discrete random variable X taking values from the finite alphabet $A := \{x_1, \cdots, x_M\}$ with probability mass function (pmf), $p_X(x_k) = \Pr[X = x_k]$, the Shannon entropy is defined as

$$H\left(p_X(x_k)\right) = -\sum_{k=1}^{M} p_X(x_k) \log p_X(x_k). \tag{1}$$

The appropriate interpretation of the Shannon entropy is the "amount of uncertainty" in a state. Pure states have Shannon entropy 0, i.e., states with no uncertainty. On the other extreme lies the maximally mixed state which gives the maximum entropy value, $H = \log M$, which is obtained when all the elements of the finite alphabet are equally likely to appear. In order to apply this concept to traffic, we generate flows from traffic traces that are captured in a network node. As mentioned previously, the flow-level traffic behavior may be analyzed by an entropy space. The clouds of data-points of this space present regular patterns in typical traffic conditions. However, malicious activity such as DoS attacks, worms and scans significantly alter the pattern of acceptable behavior. This shows that the entropy space is suitable for application in anomaly-based IDS algorithms.

A traffic trace χ of t_D seconds in duration, shown in Fig. 2, is considered to obtain its entropy space. The trace is divided into m non-overlapping traffic slots with a maximum duration of t_d seconds each. Traffic slot i is formed of K_i flows generated according to the 5-tuple defined above. This set of flows is represented as $\mathbb{F}_i = \left\{f_{1,i}, \ f_{2,i}, \ \ldots, \ f_{K_i,i}\right\}$, where $i = \{1, \ 2, \ \ldots, \ m\}$.

Having identified the flows that exist in a slot, we form cluster flows, which can be done in four different ways. Each form depends on a pivot or cluster key (CK) chosen. There are four CK:

- Source IP address CK (CK-1)
- Destination IP Address CK (CK-2)
- Source Port CK (CK-3)
- Destination port CK (CK-4)

A CK acts as a reference for the formation of cluster flows. For example, if we choose CK-1, flow clusters are formed based on source IP addresses that are observed in the i-th traffic slot, $i = \{1,\ 2,\ \ldots,\ m\}$. If we choose CK-2, then destination IP addresses will determine the shape of the cluster, and similarly for CK-3 and CK-4. Once a CK is selected, the alphabet set of the CK in question needs to be identified. The alphabet sets of CKs for a given i-traffic slot are denoted as follows:

$$\mathrm{CK}{-}1 \rightarrow \mathbb{A}_i,\ \mathrm{CK}{-}2 \rightarrow \mathbb{B}_i,\ \mathrm{CK}-3 \rightarrow \mathbb{C}_i,\ \mathrm{CK}{-}4 \rightarrow \mathbb{D}_i$$

These alphabets contain objects of the same class observed during the i-th traffic slot. As an example, the alphabet set $\mathbb{A}_i$ is formed by all different source IP addresses seen in the i-th traffic slot. The next step is to obtain the three-dimensional representation through the entropy spaces. Given traffic slot i and a cluster key CK-k, k $= 1,\ 2,\ 3, 4$, a set of flows $\mathbb{F}_i$ is created. A flow is denoted as $f_{x,i},\ x = 1, 2 \ldots K_i$. Each flow must belong to only one cluster of flows. Let $C^{\mathrm{k}}_{y,i}$ denote the y-th cluster of flows formed with flows of identical flow key value. As an example, if k $= 1$ the flow key values for clustering are the elements of $\mathbb{A}_i$ and $y = 1, \ldots, |\mathbb{A}_i|$. For this case, the first cluster of flows in time slot i for cluster key 1 is denoted as $C^1_{1,i}$ and the flow key value is the first element of $\mathbb{A}_i$. This first cluster of flows holds that $C^1_{1,i} \subset \mathbb{F}_i$. The second cluster of flows $C^1_{2,i}$ is formed by other subset of flows of $\mathbb{F}^1_i$ where flows have in common the next source IP address in the same alphabet $\mathbb{A}_i$. The subsequent clusters of flows are formed similarly. Thus, in slot i, there will be as many clusters of flows as source IP addresses in the alphabet $\mathbb{A}_i$ for cluster key CK-1, therefore, the last cluster of flows in time slot i for cluster key 1 is $C^1_{|\mathbb{A}_i|,i}$.

After clusters $C^{\mathrm{k}}_{y,i}$ have been obtained, entropy estimation is carried out for each cluster of flows. Similarly, if k=1, each cluster of flows $C^1_{y,i}$, is composed of flows $f^1_{x,i}$ that share the same flow key value of source IP address, there is no uncertainty with respect to this flow key. However, there is uncertainty with respect to the other three flow keys, i.e., source port, destination port, and destination IP address, which are denoted as the *free dimensions*. Thus, entropy is calculated for each of these three free dimensions producing a triplet of values that can be visualized inside a three-dimensional space called the *entropy space*. To represent the three-dimensional mapping, we present these free dimensions as coordinates in the space for cluster key CK-1 as follows

$$\mathrm{CK}-1 \rightarrow \left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt},}\ \widehat{H}_{\mathrm{dstIP}}\right) \tag{2a}$$

For the remaining cluster keys we have the coordinates defined as

$$\mathrm{CK}-2 \rightarrow \left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt},}\ \widehat{H}_{\mathrm{srcIP}}\right) \tag{2b}$$

$$\mathrm{CK}-3 \rightarrow \left(\widehat{H}_{\mathrm{srcIP}},\ \widehat{H}_{\mathrm{dstIP},}\ \widehat{H}_{\mathrm{dstPrt}}\right) \tag{2c}$$

$$\mathrm{CK}-4 \rightarrow \left(\widehat{H}_{\mathrm{srcIP}},\ \widehat{H}_{\mathrm{dstIP},}\ \widehat{H}_{\mathrm{srcPrt}}\right) \tag{2d}$$

Coordinates $\left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt},}\ \widehat{H}_{\mathrm{dstIP}}\right)$ are represented in the entropy space, and each point is called a *data-point*. Figure 3 summarizes this procedure. This is repeated for each value of x to get a point cloud data with a total of $|\mathbb{A}_i|$ points. This point cloud is the spatiotemporal representation of the traffic trace being analyzed. The raw entropy space is formed by the set of these three-dimensional data points represented as

$$\left\{\hat{\mathbf{H}}_i^{|\mathbb{A}_i|}\right\}^{\mathrm{RAW}} = \Big[\left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt},}\ \widehat{H}_{\mathrm{dstIP}}\right)^{(i)}_1, \\ \ldots \left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt}},\ \widehat{H}_{\mathrm{dstIP}}\right)^{(i)}_{|\mathbb{A}_i|}\Big] \tag{3}$$

where $\left\{\hat{\mathbf{H}}_i^{|\mathbb{A}_i|}\right\}^{\mathrm{RAW}}$ is composed of l triplets and $l = \sum_{i=1}^{m} |\mathbb{A}_i|$. As it can be seen, each triplet $\left(\widehat{H}_{\mathrm{srcPrt}},\ \widehat{H}_{\mathrm{dstPrt},}\ \widehat{H}_{\mathrm{dstIP}}\right)^{(\#\mathrm{slot})}_{\#\mathrm{cluster}}$ is indexed by time slot and cluster number. In

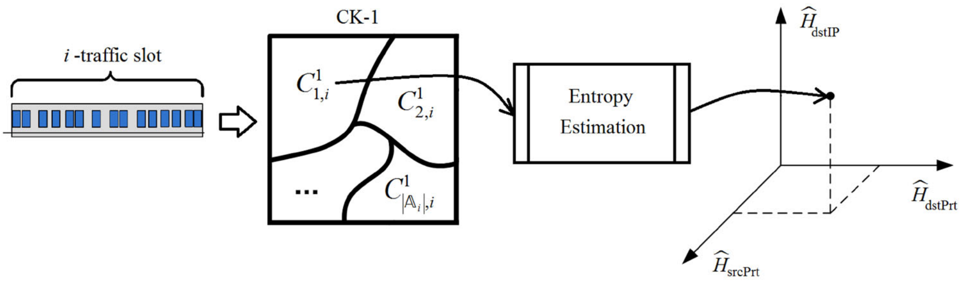

i -traffic slot
CK-1
$C^1_{1,i}$
$C^1_{2,i}$
...
$C^1_{|\mathbb{A}_i|,i}$
Entropy Estimation
$\widehat{H}_{\mathrm{dstIP}}$
$\widehat{H}_{\mathrm{dstPrt}}$
$\widehat{H}_{\mathrm{srcPrt}}$

Fig. 3 Data-point generation from a cluster of flows in CK-1

this way, for atypical values, it is possible to identify the time slot and the free dimensions associated. This is important to track for forensic analyses.

With the spatiotemporal description given by the entropy spaces for different time slots, we carry out experimental analysis of the point clouds and determine the patterns that they form which are related to the behavior of the internet traffic where monitoring is taking place. We captured traffic traces daily in two academic networks under attack-free conditions, and the corresponding entropy spaces of each traffic trace have similarities in the cloud shapes. Afterwards, we performed three controlled attacks consisting on worms (code red, blaster and welchia), and the patterns of the point clouds change in shape significantly while the attack is taking place. Recall that our method is providing a point cloud for each time slot, allowing to visualize the effects on the shape of the clouds as time progresses.

While attacks take place, portions of malicious traffic behave in entropy spaces as outliers. Data-points with high or low entropy values indicate changes in the random behavior of some traffic features of a network, e.g., IP addresses and ports can be extremely diverse. This change in behavior is key to develop an unsupervised method because the classification of anomalous traffic may be done based on outlier detection techniques, using statistical distances in the entropy space.

3.4 Clustering-based traffic sanitization

After the creation of the clusters of flows and the corresponding entropy spaces, information is ready to be processed by the *Sanitization* block, which is responsible for the analysis of those data-points that are the *outliers*.

In general, a data-point represents a relationship of one element of an alphabet set of CK-k for given traffic slot i with the rest of the flow keys or free dimensions in a cluster of flows in that particular traffic slot. Since this relationship is described in terms of entropy values, the abrupt changes or variability of the free dimensions generate significant changes to the position of the data-points in the entropy space. The position of the anomalous data-point or outlier, is characterized by a separation from the cloud of data points. Anomalous traffic features have high or low diversity, and as a consequence, the entropy space helps to distinguish the anomalous portions included in the network traffic. Data-points representing these anomalies get separated from the main point cloud forming their own cluster which is made up of a small number of points. If we define a number of clusters k for the entropy space, at least one of them will capture the anomalous traffic data point cloud, and its location within the space will be given by its centroid. This cluster will contain a small number of elements and is a low density cluster.

The processing of the data point cloud starts with the elimination of those data points that lie on any of the axes of the entropy space, i.e., data points with nulls on two of the free dimensions. These data points provide small amount of information regarding the remaining set of data points in the cloud, and interfere with the final position of the centroids. Afterwards, and with the purpose of obtaining significant results, the new data point cloud must be analyzed to detect and reject outliers. This detection of the outliers can be carried out by the use of statistical distances, e.g., Mahalanobis distance. For the n multivariable samples of dimension p, $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$, the Mahalanobis distance of the i-th observation, $\boldsymbol{x}_i$, with respect to a point $\boldsymbol{y}$ is defined as

$$MD_i(x, y) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{y})}, \tag{4}$$

where Σ^{-1} is the inverse of the covariance matrix. A constant Mahalanobis distance $MD_i = d$ between a vector $\boldsymbol{x}$ and the mean value $\boldsymbol{\mu}$ defines a prediction ellipse, where all the points on the ellipse are equally likely and hence have the same distance from the ellipse's centroid. For Gaussian multivariable data with mean $\boldsymbol{\mu}$ and covariance matrix Σ, the square of the Mahalanobis distance is approximated by a chi-squared distribution with p degrees of freedom (df) denoted as χ_p^2, i.e.,

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \sim \chi_p^2. \tag{5}$$

Defining a specific hyper-ellipse, where MD_i^2 takes a critical value of χ_p^2 evaluated in α, we get the probability that random sample $\boldsymbol{x}$ is within the prediction ellipse as

$$\Pr\left\{(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha)\right\} = 1 - \alpha, \tag{6}$$

which corresponds to the $(1 - \alpha)$ percentile of $\chi_p^2(\alpha)$. The detection of outliers is fundamental for traffic sanitization. The entropy space helps to show outliers originated by anomalous traffic connections. In this paper, we consider typical behavior those data points with Mahalanobis distance to the centroids that belong to the 98th percentile, the ones left out were considered outliers. By the elimination of the outliers and those points that lie on the axes, we create a new and sanitized entropy space, which needs to be characterized. This new space is a different version of $\left\{\hat{\mathbf{H}}_i^{|\mathbb{A}_i|}\right\}^{\mathrm{RAW}}$, and as a consequence, its data points can be grouped in a set of clusters. The clustering is an unsupervised classification where there is no a-priori knowledge. In this paper, we use k-means algorithm to form clusters and analyze this new entropy space [28]. Two of the disadvantages of the k-means algorithm are that is not capable of handling noisy data and outliers, and that it needs to have a specific number of clusters k defined before being applied. We resolve the first disadvantage by removing the outliers from the raw data point cloud, and the second disadvantage by the use of Mojena's stopping rule.

The Mojena's stopping rule [31], also known as upper tail rule, [29], allows to specify the appropriate number of groups in a hierarchical clustering. This method identifies the first stage in the dendrogram at which there is a large change in the distance between clusters. The idea is to stop the fusion process and therefore, select the number of groups already found, when the following condition is satisfied:

$$\alpha_{k+1} > \bar{\alpha} + cs_{\alpha}, \tag{7}$$

where α_{k} is the fusion level at cluster stage $k = 0,\ 1,\ \ldots,$ $n-1$. $\bar{\alpha}$ and s_{α} are the mean and unbiased standard deviation of the fusion levels and c is a constant suggested by Mojena to be in the range 2.75–3.50 for optimal results, which will be discussed in Sect. 5.1.

When traffic is within acceptable behavior, similar patterns of entropy data-points are formed. Now, when typical and anomalous traffic are combined into the same capture, all known entropy data-points patterns are altered and rearranged. This change in traffic conditions significantly displaces the coordinates of the reference centroids. In summary, the rationale behind our approach is the assumption that typical and anomalous traffic form different clusters in the entropy space.

3.5 Validation and verification of sanitized datasets

Sanitization consists in filtering the packets of the traffic trace that correspond to those entropy data-points that form the outliers. To do this, several filters are used in *tcpdump*-for the IP addresses and ports involved. The result of this operation is a new traffic trace sanitized. Consistency of this procedure is verified, with homogeneity at two levels.

The Verification layer is responsible for the confirmation that the properties of the filtered traffic and its respective sanitized entropy space $\left\{\hat{\mathbf{H}}_i^{|\mathbb{A}_i|}\right\}^{\mathrm{SAN}}$ are similar or within certain criteria based on two *behaviors*:

(a) *Cluster centroid homogeneity k* cluster centroids will be generated in the entropy spaces when clustering is applied. The technique of k-means is used to obtain the centroids from the entropy data-points in the 3D entropy spaces, and when such coordinates are similar for the three centroids, then the trace has been sanitized.

(b) *Distribution homogeneity* In the PCA-transformed entropy space, the probability distribution of each of the free dimensions is similar to each other when the data-points correspond to a sanitized traffic trace.

The two behaviors just described will not be as such in the original traffic trace due to the anomalous traffic that is present when captured. This anomalous traffic generates different behaviors from those that will be obtained once the trace is sanitized.

3.6 Implementation considerations

The Exporter block is implemented by a Perl script called flowanalyzer.pl script, where each cluster key or pivot is managed and computed by using typical data structures called hashes (more details about the deduction of the theoretical complexity O(1) can be found in Donald Knut's book [24]). Here, the advantage of using hashes is to maintain a direct reference among the pivot and their corresponding data-fields for further purposes, like a dictionary. Flow clustering and Entropy space blocks are implemented by flowentropy-naive.pl using same strategy of hashes. After choosing the corresponding pivot, each flow key is assigned. Each flow key element (an element that belongs to srcIP, dstIP, srcPrt, and dstPrt set, depending of the chosen pivot) and its corresponding frequency are updated each time that the flow key element is found. Instead of using loops to locate the flow key element, the hash structure maintains a straightforward reference between the flow key element and its frequency, thus reducing the computation time by using the key element like a memory address. The same feature is also used when the entropy is computed. The clustering procedure in the Sanitization block is implemented by Elkan's fast k-means algorithm, which uses various geometric inequalities to reduce considerably the number of distance computations required, [16]. Its overall complexity is about O(nke) where n is the number of data-points, and e is the number of iterations. Another important issue about the traffic trace sanitization method proposed is that it consists of the analysis of previously captured, recorded and stored traffic traces, thus the methodology is an offline process that generates as output new and sanitized traffic traces. Hence, there are no special considerations for real-time processing.

4 Datasets

The evaluation of the proposed approach was carried out over real-world datasets collected from UAN (Universidad Autonoma de Nayarit). The corpus of data used in this paper consists of 63 traffic traces, organized in four datasets UAN-01 to UAN-04. The size of the set is 169.0387 GB with 324.2 million packets. Traces were collected on weekdays at the same time of the day to minimize errors from diurnal effects in network usage. Table 1 shows the summary of six traces that are part of the data set UAN-01.

5 Experimental results and analysis

In this section, we introduce the network architecture where tests were conducted, together with the architecture used for the traffic trace sanitization that we propose, the entropy spaces generated by such traces, the analysis of the entropy

Table 1 Traffic traces corresponding to dataset UAN-01

Dataset UAN-01			
Trace name (Dataset-Yr/M/DD)	Time period	Number of packets	Size (GBytes)
UAN-01-110801	07:00:00-14:59:59 2011	2,982,275	1.23573
UAN-01-110802	07:00:00-14:59:59 2011	3,284,151	1.20658
UAN-01-110803	07:00:00-14:59:59 2011	4,053,405	1.60624
UAN-01-110805	07:00:00-14:59:59 2011	3,428,552	1.16633
UAN-01-110806	07:00:00-14:59:59 2011	2,670,853	0.75337
UAN-01-110807	07:00:00-14:59:59 2011	2,213,258	0.72636
	Total number of packets	18,632,494	6.69461

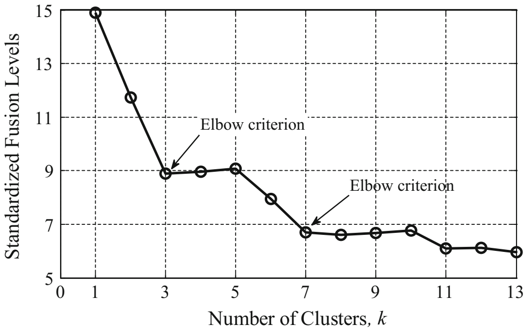


Fig. 4 Mojena's plot for trace UAN-01-110802

spaces and the further transformation of the entropy data-points to analyze the status of the traffic in the network. With these, we establish criteria for the detection of anomalies in the network segment.

We exemplify the sanitizing procedure using data set UAN-01 consisting of six traffic traces captured. However, the procedure is applied to other traces as well.

5.1 Determining the number of clusters

Figure 4 shows a Mojena's plot employing the standardized fusion levels for the entropy space of trace UAN-01-110802. The first elbow indicates that three clusters is an adequate number. However, there are other elbows at $k = 7$ and $k = 11$ that could be considered. We select the smallest number because it represents the largest change in distance for the centroids to be identified, otherwise more clusters would be formed in the same space with their corresponding centroids being closer to one another.

The k-means analysis is used to partition each of the entropy data-point clouds in three clusters, $k = 3$ which is in agreement to the information obtained from Mojena's stopping rule. Verification process will be carried out once sanitization has been completed. We use an unsupervised method to identify the existence of natural patterns and outliers that form the clouds of entropy data-points. In order to do this analysis, k-means clustering was chosen. Figure 5 shows the entropy spaces generated by using the cluster key of *srcIP* for each of the traffic traces of dataset UAN-01. Entropy space analysis can be performed using Table 2 which contains information about the centroid identification, the centroid coordinates, the norm of the centroid, and the population of each centroid. Each entropy space contains three centroids denoted as C_1, C_2 and C_3, respectively. The properties of each centroid are determined by the behavior of the traffic. For example, entropy spaces of traces UAN-01-110801, UAN-01-110803, UAN-01-110805 y UAN-01-110806 were processed to generate three clusters. However, we can notice that there is a cluster with very low data-points for each entropy space. According to the aforementioned criterion, these four traces contain anomalous traffic. On the other hand, traces UAN-01-110802 and UAN-01-110807 differ from that behavior. Regarding the content of Table 2, we observe a similarity on the position and norm of the centroids.

5.2 Sanitizing anomalous traces with clustering

In this section, the previous four traffic traces that were identified as anomalous by using cluster density are analyzed for detection and anomalous traffic filtering. In each subsection, the procedure for sanitizing the trace is discussed for each traffic trace.

5.2.1 Trace UAN-01-110801

Figure 6a shows the entropy space based on the CK-1 pivot of this trace. On the results presented in Table 2, there is a cluster of low density with a centroid C_2. Data-points that belong to the low density cluster are generated by an anomalous traffic behavior. The filtering of such anomalous data-points is done by applying a prediction ellipse on dimensions $\hat{H}_{\text{srcPrt}}$ and $\hat{H}_{\text{dstPrt}}$. For this sample of data-points we obtain the mean value and covariance matrix as

$$\boldsymbol{\mu} = \begin{bmatrix} 0.2647 \\ 3.1272 \end{bmatrix}, \quad \Sigma^{-1} = \begin{bmatrix} 6.3429 & 0.8146 \\ 0.8146 & 0.5384 \end{bmatrix}$$

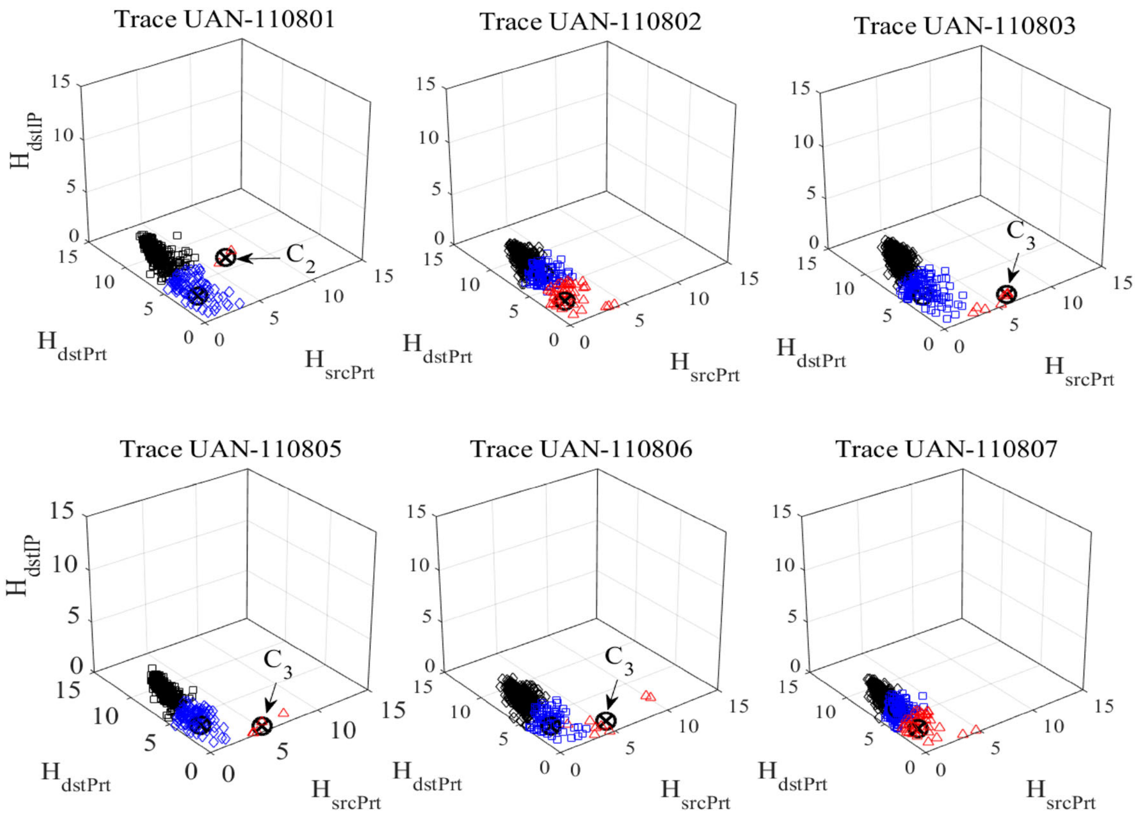


Fig. 5 Entropy spaces for dataset UAN-01, clustering with k-means is used, $k = 3$

Now, from tables of the chi-squared distribution, the critical value of χ_p^2 for the 98th percentile with two degrees of freedom, $p = 2$ is obtained for $\alpha = 0.02$, which gives $\chi_p^2(\alpha) = 7.82$, hence the prediction ellipse has a Mahalanobis distance of

$$MD = \sqrt{7.82} = 2.7964$$

The data points outside the prediction ellipse of the 98th percentile have a Mahalanobis distance of $MD_i > 2.7964$. In Fig 6b, we can notice that there are 9 data-points outside of the ellipse. However, the forensic analysis of the traffic related to those 9 data-points indicates that only 7 data-points were originated by anomalous traffic. These anomalous data-points are labelled as DP_i, $i = 1, \ldots, 7$. Table 3 was elaborated using the mean value and the covariance matrix just obtained, and shows the Mahalanobis distance of such data points taking values over 2.7964.

Through forensic analysis we identified the flow clusters related to those entropy data-points generating the anomaly. The anomalous data-points belong to traffic slot $i = 427$, and corresponds to two flow clusters, one with duration of six seconds and the other one with duration of 0.012 seconds. The total number of packets is 389, and the victim of such anomaly is the local IP address 192.100.162.21 assigned to an institutional email server. On the other hand, there exists an external IP address, 85.156.207.209 that belongs to the attacker with geographical origin in Finland. The attack consists in a port scanning of the email server with the attempt to find security vulnerabilities of some of its ports.

Traffic filtering is not only specific to removal of traffic related to the two data-points that belong to the low density cluster, i.e., all traffic that involves the attacker's IP address was removed from the trace.

5.2.2 Trace UAN-01-110803

The cluster of low density on this trace is identified with centroid C_3. The forensic analysis of such traffic provides the IP address of the attacker to be 82.200.168.68 located in Kazakhstan. It was found that the target of the attack was the email and web servers of the institutional network with IP addresses 192.100.162.21 and 192.100.162.50, respectively. Forensic analysis revealed a Brute force SSH attack, which was comprised of 262,478 packets. In this anomalous traffic we identified a repeating pattern of Brute force SSH attack consisting of 24 packets in which the attacker establishes a session with a random source port to destination port 22, which is assigned for Secure SHell protocol (SSH) of one of the aforementioned servers. On average, each session had a duration of 2.5 s. Because the session is encrypted, we cannot determine all details of this anomalous traffic.

Table 2 Dataset UAN-01analysis summary

C_i	Coordinates			Norm	Cluster's population
	H_{scrPrt}	H_{dstPrt}	H_{dstIP}		
Trace UAN-01-110801					
C_1	0.1594	5.9946	3.3066	6.8479	439
C_2	6.4566	6.0005	0.5534	8.8318	2
C_3	0.7367	1.7978	1.2694	2.3208	241
Trace UAN-01-110802					
C_1	0.1422	6.0527	3.2823	6.8869	365
C_2	0.4843	3.9056	2.5553	4.6924	123
C_3	0.6624	1.4684	1.2031	2.0106	265
Trace UAN-01-110803					
C_1	0.5479	6.2495	3.2072	7.0458	381
C_2	0.5244	3.3513	1.1669	3.5870	202
C_3	5.7298	0.0091	0.9806	5.8131	6
Trace UAN-01-110805					
C_1	0.1149	5.6403	2.9148	6.3500	428
C_2	0.5151	1.9214	1.3786	2.4203	354
C_3	4.8267	0.1798	0.3970	4.8464	5
Trace UAN-01-110806					
C_1	0.1716	5.0231	2.5297	5.6268	437
C_2	0.5170	1.7964	1.3761	2.3213	333
C_3	5.2347	1.4249	0.0617	5.4255	11
Trace UAN-01-110807					
C_1	0.1031	5.4985	2.6092	6.0871	241
C_2	0.2186	3.6167	2.1033	4.1897	193
C_3	0.5048	1.5784	1.2519	2.0769	261

Table 3 Mahalanobis distance for anomalous traffic data-points

Data-point	MD_i
DP1	11.824
DP2	10.096
DP3	5.352
DP4	3.267
DP5	3.054
DP6	4.393
DP7	3.800

5.2.3 Trace UAN-01-110805

By the forensic analysis of the traffic slots corresponding to cluster with centroid C_3, we could get the IP address of the attacker which was 222.189.238.114. The attack performed was a brute force attempt, and was trying to establish "abnormal" amount of connection to the MySQL Database Server (192.100.162.50) and tried to login in as "root". It is kind of a brute force attempt, which caused MySQL to respond with "Response Error 1045". Using offensive IP Database Query service at www.bizimbal.com, we discover that the source IP address of the attacker is blacklisted because it is associated with 55 offensive actions and its location is in China. Additionally to the attack already found, another anomaly was identified in the trace. The external IP source address was identified to be 213.186.118.39, and with the forensic analysis we could determine that it is located in Kiev, Ukraine. Using Wireshark we identified the victim to be the institutional web server (192.100.162.50). The packets related to the attack from the Ukraine show that its intention was to use the HTTP HEAD method by using different URLs to access as web server administrator and obtain access to the internal information. This attack was conducted in 50 seconds.

5.2.4 Trace UAN-01-110806

Our forensic analysis of the trace allowed us to determine the IP address of the attacker which was 201.138.218.64 and

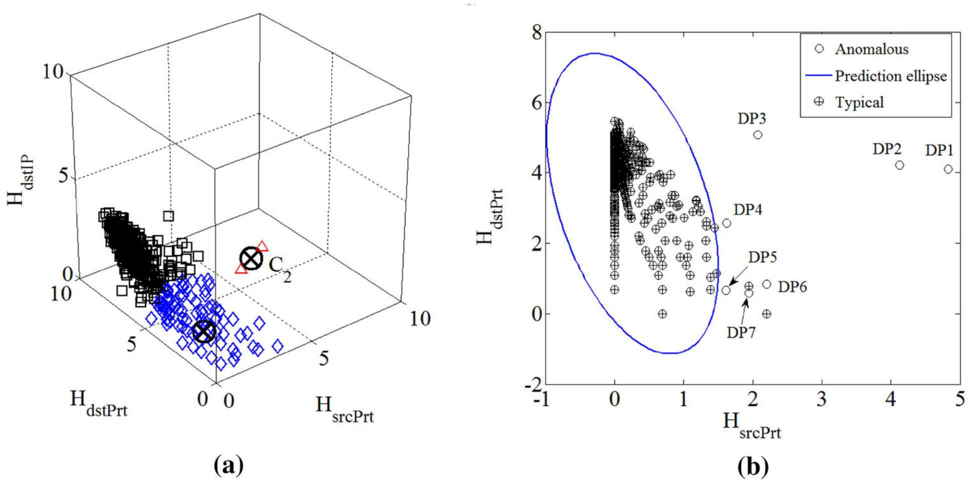


Fig. 6 Traffic trace UAN-01-110801 **a** entropy space with suspicious behavior given by centroid C_2. **b** Classification of entropy data-points and their relation to prediction ellipse

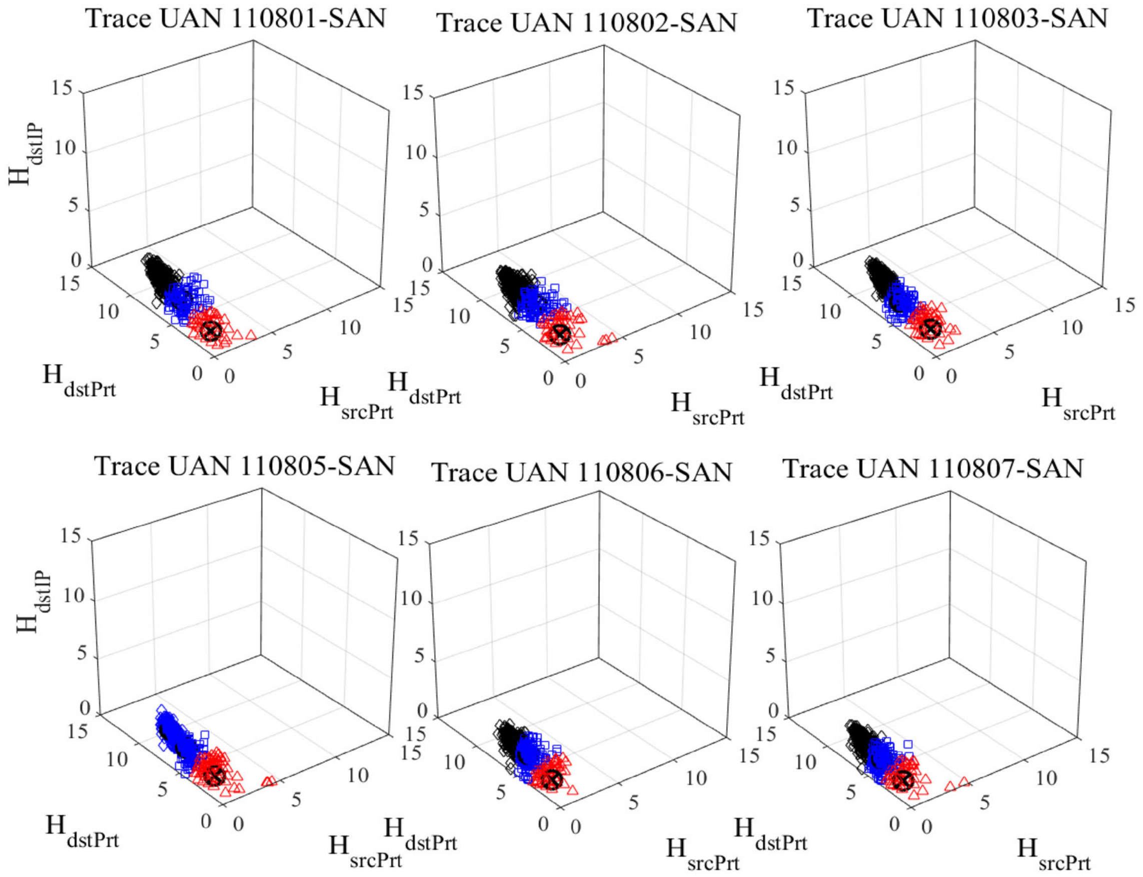


Fig. 7 Entropy spaces for sanitized dataset UAN-01-SAN

the TCP ports and IP address that he attempted to access. The attacker tried to access, without success, the etc/passwd directory. Subsequently, using the technique of SYN scan, a search for open ports on the server was made.

Once the anomalies in the traffic traces have been detected, they are sanitized by filtering out all those packets related to the anomalies just identified. The four sanitized traces and two traces with no low density centroids (UAN-01-110802 and UAN-01-110807) form a new data set, identified as UAN-01-SAN. Entropy spaces of this dataset are analyzed by k-means and are shown in Fig. 7. The similarity among these traffic traces is observed. Table 4 shows a summary of the sanitized dataset UAN-01-SAN. We notice the closeness of the centroid location either by coordinates or by norm. It can also be seen that there are no low density clusters. The last column includes comments about the results obtained with the applied forensic analysis.

The verification process shows that the sanitized traces do not have low density clusters, and the sanitized dataset presents homogeneity on the centroid positions, which can be noted on the coordinates or the norm of the centroid norm. With this procedure, any network administrator or researcher can construct sanitized traffic traces that characterize normal traffic behavior in the network and that can be used for intrusion detection purposes

5.3 Verification by an entropy space transformation

In this subsection, we introduce the other verification method of the traffic sanitization process which is based on distributions of transformed entropy space. The transformation of the entropy spaces basically consists of the use of statistical tools to generate a different set of parameters that can be of help in the detection of anomalies. We use the correlation coefficients and the PCA techniques [14].

The PCA technique consists of transforming the set of variables that are correlated into an uncorrelated set of variables denoted as principal components, trying to reduce dimensionality of the set of variables. The principal components result in a system with data in such way that the first dimension corresponds to the maximum variability of the original data; the second largest variability will be in the second dimension, and so on. We take the original three-dimensional entropy spaces for dataset UAN-01 and we apply the PCA technique. The result is a two-dimensional dataset with axes PCA1 and PCA2. Now, we process the

Table 4 Dataset UAN-01-SAN analysis summary

C_i	Coordinates			Norm	Cluster population	Comment
	H_{scrPrt}	H_{dstPrt}	H_{dstIP}			
Trace UAN-01-110801						
C_1	0.1060	6.2668	3.4234	7.1417	367	Sanitized trace of SYN scan attack
C_2	0.4119	4.1754	2.4559	4.8616	115	
C_3	0.7503	1.4152	1.1116	1.9498	192	
Trace UAN-01-110802						
C_1	0.1422	6.0527	3.2823	6.8869	365	No evidence of suspicious traffic
C_2	0.4843	3.9056	2.5553	4.6924	123	
C_3	0.6624	1.4684	1.2031	2.0106	265	
Trace UAN-01-110803						
C_1	0.0602	6.1619	3.3322	7.0055	330	Sanitized trace of Brute force SSH attack (details not identified due to the use of encryption)
C_2	0.1417	4.2603	2.3992	4.8915	139	
C_3	0.6537	1.5139	1.2951	2.0968	257	
Trace UAN-01-110805						
C_1	0.0834	6.1175	3.1159	6.8658	283	Sanitized trace of Brute force attack and Malicious scripts attack
C_2	0.1626	4.4327	2.4320	5.0586	181	
C_3	0.5933	1.7170	1.2904	2.2283	307	
Trace UAN-01-110806						
C_1	0.0990	5.5030	2.6631	6.1143	268	Sanitized trace of SYN scan attack
C_2	0.2136	3.8976	2.3576	4.5602	202	
C_3	0.4967	1.6746	1.2661	2.1573	289	
Trace UAN-01-110807						
C_1	0.1031	5.4985	2.6092	6.0871	241	No evidence of suspicious traffic
C_2	0.2186	3.6167	2.1033	4.1897	193	
C_3	0.5048	1.5784	1.2519	2.0769	261	

two-dimensional information by characterizing the behavior of components PCA1 and PCA2. This is achieved by estimating their corresponding probability density function (pdf), f. The Kernel Density Estimator (KDE) is a nonparametric technique used to construct estimates for f of observed random variables. KDE is suitable for problems involving multi-modal densities and when the underlying densities are unknown. The value of the density at a given point is estimated as the sum of the smoothed values of kernel functions $K_h(x)$. Each kernel function is associated with a positive number h which determines the level of smoothing created by the function; this smoothing parameter is called the bandwidth of the kernel [41]. A kernel function satisfies the condition $\int K_h(x) = 1$. Usually, but no always, $K(x)$ is a unimodal probability density symmetric about zero. Mathematically, KDE is defined by letting $X = \{X_1, X_2, \ldots, X_n\}$, denote a vector of n observations from a random variable with density f, then the kernel estimator of f at point x can be obtained by

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \tag{8}$$

In this paper, we use the well-known Gaussian kernel which is defined as

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{9}$$

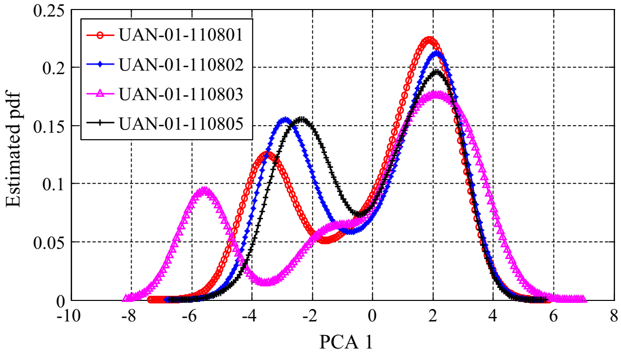


Fig. 8 KDE for PCA 1 from unsanitized dataset UAN-01

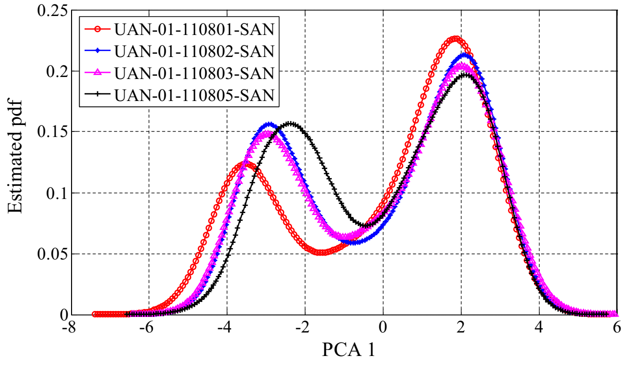


Fig. 10 KDE for PCA 1 of sanitized dataset UAN-01-SAN

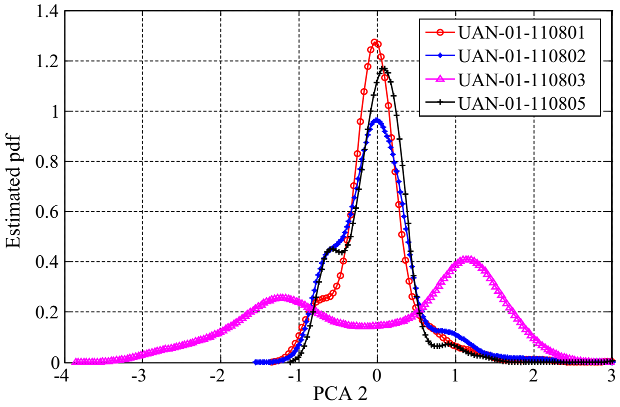


Fig. 9 KDE for PCA 2 from unsanitized dataset UAN-01

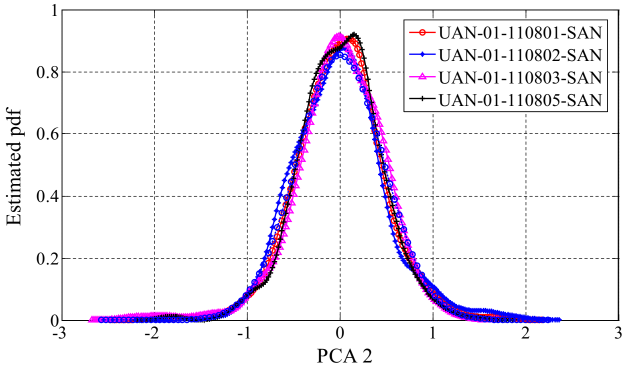


Fig. 11 KDE for PCA 2 of sanitized dataset UAN-01-SAN

The selection of the bandwidth is an important aspect in KDE. In this paper we calculate the bandwidth as suggested by [40] and [44] using

$$h = 1.06\hat{\sigma} n^{\frac{-1}{5}}. \tag{10}$$

where $\hat{\sigma}$ is the sample standard deviation of vector X. Figure 8 shows the result of the estimation of the pdf corresponding to PCA1 for the unsanitized dataset UAN-01. Recall that four of the six original traffic traces had anomalies, then for those four unsanitized traces, we apply this methodology and obtain an estimation of the pdf of the corresponding PCA1 for each of the traces. It can be seen in the figure that the estimated pdfs have clear differences, however, most of them can be considered to have bimodality, and in other words, their shapes have two important lobes. Notice that, traffic trace UAN-01-110803 has three modes. Similarly, the technique is applied to the next dimension, denoted as PCA2. This is shown in Fig. 9, where there are clear differences among the estimated pdfs, i.e., PCA2 has more sensitivity to the anomalies in the traffic traces.

Results in Figs. 8 and 9 show that the use of unsanitized datasets to create network traffic profiles is not adequate since it will not represent the typical traffic behavior of the network due to the significant content of anomalous traffic. This is a problem that affects anomaly-based NIDS since it decreases the sensibility for detection of outliers. Therefore, it is important to reduce the noise level of intrusions before any modeling is carried out for the characterization of the normal or typical traffic behavior. To proceed with the characterization of the typical traffic of the network, first follow the sanitization procedure of the traffic traces that was described previously in Sect. 3, and with the anomalies filtered out proceed to carry out the PCA and the KDE techniques. Figure 10 shows the pdfs estimation using the KDE technique for the sanitized traces of the dataset UAN-01-SAN for the variable PCA1. It can be seen that the behavior of the pdfs is different from that shown in Fig. 8 and that now all the estimated pdfs consists of two modes. Figure 11 contains the estimated pdfs for the variable PCA2 of the sanitized traces, it can also be seen the difference compared to those in Fig. 9. The spurious modes on unsanitized traffic traces disappear once the traces are sanitized. This provides a homogeneous behavior with a single mode for all the pdfs in the figure.

The activity level of users and servers affects the behavior of sanitized traces. For instance, datasets UAN-01 and UAN-04 are formed by traces obtained under low activity periods in the network. Sanitized traces from dataset UAN-04 presented similar results to those found in dataset UAN-01 with respect to the pdf behavior of the principal components. On the other hand, we observed that in high activity periods (datasets UAN-02 and UAN-03), sanitized traces presented a bimodal behavior with respect to pdf of PCA2 which is not present in low activity periods.

6 Conclusions

In this paper we propose a flow-based and unsupervised method to obtain traffic traces that represent typical traffic behavior in a network segment. The method consists of sanitization of the traffic traces and verification by position of cluster centroids. The characterization of the behavior is obtained by means of the entropy spaces and the further application of the PCA methodology in order to obtain traces that have the same statistical behavior of typical traffic. The verification of such traces is carried out by using the KDE method. Given the shortage of labeled datasets, our results can be extended to generate evaluation datasets for Intrusion Detection Systems by replaying our sanitized traffic using tools like tcpreplay and the controlled injection of malicious traffic.

With a proper sanitization process, it is possible to remove packets that could alter the accuracy of a NIDS. Less excessive data means more accuracy in detecting anomalies. This evidence is shown when few data-points generated by anomaly traffic move away from the rest of the data-points. On the other hand, clustering techniques are highly enhanced with the use of information theory and additional strategies to reduce the amount of analyzed information (flow processing), because it helps to NIDS to determine if a network traffic is anomalous with certain degree of accuracy by using relevant data. The use of the techniques described here, could be adapted to networks of different sizes and different traffic loads without reconfiguration of the NIDS as it could occur with supervised techniques.

Acknowledgments We would like to thank CONACyT for its support through its "SEP-CONACyT Ciencia Básica CB-2011" Research funding for Project number 167859. We also would like to thank the Telecommunications Focus Group at Tecnológico de Monterrey, Campus Monterrey.

References

1. Ahmad, I., Abdullah, A., Alghamdi, A., & Hussain, M. (2013). Optimized intrusion detection mechanism using soft computing techniques. *Telecommunication Systems, 52*(4), 2187–2195.
2. Anthes, G. (2010). Security in the cloud. *Communications of the ACM*, *53*(11), 16–18.
3. Aydın, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers & Electrical Engineering*, *35*(3), 517–526.
4. Bace, R. G. (2000). *Intrusion detection*. Indianapolis, IN: Macmillan Technical Publishing.
5. Bermúdez-Edo, M., Salazar-Hernández, R., Díaz-Verdejo, J., & García-Teodoro, P. (2006). Proposals on assessment environments for anomaly-based network intrusion detection systems (Vol. 4347, pp. 210–221). Berlin: Springer.
6. Brown, C., Cowperthwaite, A., Hijazi, A. & Somayaji, A. (2009). Analysis of the 1999 DARPA/Lincoln Laboratory IDS evaluation data with NetADHICT. In *Proceedings of the 2th IEEE international conference on computational intelligence for security and defense applications*, Piscataway, NJ (pp. 67–73).
7. Brugger, S. T. & Chow, J. (2007). An assessment of the DARPA IDS evaluation dataset using snort. Technical Report CSE-2007-1.
8. Burkhart, M., Schatzmann, D., Trammell, B., Boschi, E., & Plattner, B. (2010). The role of network trace anonymization under attack. *SIGCOMM Computer Communication Review*, *40*(1), 5–11.
9. Chen, L. M., Chen, M. C., Liao, W., & Sun, Y. S. (2013). A scalable network forensics mechanism for stealthy self-propagating attacks. *Computer Communications*, *36*(13), 1471–1484.
10. Cisco Systems. (2011). Data sheets and literature. http://www.cisco.com/en/US/products/ps6601/prod_literature.html.
11. Claise, B. (2008). Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. In RFC 5101.
12. Cretu, G. F., Stavrou, A., Locasto, M. E., Stolfo, S. J., & Keromytis, A. D. (2008). Casting out demons: Sanitizing training data for anomaly sensors. In *IEEE symposium on security and privacy, SP 2008* (pp. 81–95).
13. Cretu, G., Stavrou, A., Stolfo, S. J., Keromytis, A. D., & Locasto, M. E. (2013). U.S. Patent No. 8,407,160. Washington, DC: U.S. Patent and Trademark Office.
14. Cureton, E. E., & D'Agostino, R. B. (1983). *Factor analysis, an applied approach*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
15. Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering*, *13*(2), 222–232.
16. Elkan, C. (2003). Using the triangle inequality to qccelerate *k*-Means. In *Proceedings of ICML* (pp. 147–153).
17. Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., et al. (2003). Packet-level traffic measurements from the sprint IP backbone. *IEEE Network*, *17*(6), 6–16.
18. Gang, L., Hongli, Z., Yu, Z., Qassrawi, M. T., Xiangzhan, Y., & Lizhi, P. (2013). Automatically mining application signatures for lightweight deep packet inspection. *Communications, China*, *10*(6), 86–99.
19. He, W., Hu, G., & Zhou, Y. (2012). Large-scale IP network behavior anomaly detection and identification using substructure-based approach and multivariate time series mining. *Telecommunication Systems, 50*(1), 1–13.
20. He, D., Kumar, N., & Khan, M. K. (2014). *Robust anonymous authentication protocol for healthcare applications using wireless medical sensor networks*., Multimedia systems Berlin: Springer.
21. Huang, C., & Janies, J. (2009). An adaptive approach to granular real-time anomaly detection. *EURASIP Journal on Advances in Signal Processing*, *7*, 893.
22. Jordan, E. H., Kelly, E. J., & Jordan, K. B. (2013). U.S. Patent Application 13/828,510.
23. Juniper Networks. (2010). http://www.juniper.net.
24. Knuth, D. (1997). *The art of computer programming* (3rd ed.). Reading, MA: Addison-Wesley.
25. Lakhina, A., Crovella, M., & Diot, C. (2005). Mining anomalies using traffic feature distributions. *SIGCOMM '05: Proceedings of the 2005 conference on applications, technologies, architectures, and protocols for computer communications* (pp. 217–228). New York, NY: ACM.

26. Langin, C., & Rahimi, S. (2010). Soft computing in intrusion detection: the state of the art. *Journal of Ambient Intelligence and Humanized Computing*, *1*(2), 133–145.
27. Laskov, P., et al. (2005). *Learning intrusion detection: Supervised or unsupervised?.*, Image analysis and processing-ICIAP Berlin: Springer.
28. Macqueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Procedings of the fifth Berkeley symposium on math, statistics, and probability* (Vol. 1, pp. 281–297). University of California Press.
29. Martinez, W. L., Martinez, A. R., & Solka, J. L. (2011). *Exploratory data analysis with MATLAB* (2nd ed.). Boca Raton: CRC.
30. McMahon, D. (2014). Beyond perimeter defense: Defense-in-depth leveraging upstream security. *Best Practices in Computer Network Defense: Incident Detection and Response*, *35*, 43–53.
31. Mojena, R. (1977). Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal*, *20*(4), 359–363.
32. Nadeem, A., & Howarth, M. (2013). Protection of MANETs from a range of attacks using an intrusion detection and prevention system. *Telecommunication Systems, 52*(4), 2047–2058.
33. Narang, P., Ray, S., Hota, C., & Venkatakrishnan, V. (2014). PeerShark: Detecting peer-to-peer botnets by tracking conversations. In *IEEE security and privacy workshops* (pp. 108–115). IEEE.
34. Narang, P., Hota, C., & Venkatakrishnan, V. N. (2014). PeerShark: Flow-clustering and conversation-generation for malicious peer-to-peer traffic identification. *EURASIP Journal on Information Security*, *2014*(1), 1–12.
35. Nikolova, E., & Jecheva, V. (2012). Some similarity coefficients and application of data mining techniques to the anomaly-based IDS. *Telecommunication Systems, 50*(2), 127–136.
36. Nychis, G., Sekar, V., Andersen, D. G., Kim, H. & Zhang H. (2008). An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the internet measurement conference*, Vouliagmeni (pp. 151–156).
37. Paxson, V. (2004). Strategies for sound internet measurement. In *IMC '04: Proceedings of the 4th ACM. SIGCOMM conference on Internet measurement* (pp. 263–271).
38. Robinson, A., Chan, Y., & Dietz, D. (2006). Detecting a security disturbance in multi commodity stochastic networks. *Telecommunication Systems*, *31*(1), 11–27.
39. RSA 2012 Cybercrime Trends Report. http://www.rsa.com.
40. Scott, D. W. (2001). *Multivariate density estimation: Theory, practice, and visualization*. Chichester: Wiley-Interscience.
41. Scott, D. W., & Rain, S. R. (2004). Multi-dimensional density estimation. In C. R. Rao & E. J. Wegman (Eds.), *Handbook of statistics data mining and computational statistics*. New York: Elsevier.
42. Shafi, K., Abbass, H. A. & Zhu, W. (2009). A methodology to evaluate supervised learning algorithms for intrusion detection, Technical Report.
43. Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, *31*(3), 357–374.
44. Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman and Hall.
45. Silviera, F., Diot, C., Taft, N. & Goviandan, R. (2010, June). Detecting traffic anomalies using an equilibrium property. In *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, New York (pp. 377–378).
46. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An overview of IP flow-based intrusion detection. *IEEE Communications Surveys & Tutorials*, *12*(3), 343–356.
47. Tolle, J., Jahnke, M., Felde N. G., & Martini, P. (2006). Impact of sanitized message flows in a cooperative intrusion warning system. In *IEEE MILCOM'06*.
48. Velarde-Alvarado, P., Vargas-Rosales, C., Toral-Cruz, H., Ramirez-Pacheco, J., & Hernandez-Aquino, R. (2013). *Characterizing flow-level traffic behavior with entropy spaces for anomaly detection, Building next-generation converged networks: Theory and practice*. Baca Raton, FL: CRC.
49. Wressnegger, C., Schwenk, G., Arp, D., & Rieck, K. (2013, November). A close look on n-grams in intrusion detection: anomaly detection vs. classification. In *Proceedings of the 2013 ACM workshop on artificial intelligence and security* (pp. 67–76). New Yok, NY: ACM.
50. Xu, K., Zhang, Z., & Bhattacharyya, S. (2008). Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Transactions on Networking*, *16*(6), 1241–1252.

Pablo Velarde-Alvarado currently, he is a Research-Professor at the Area of Basic Sciences and Engineering of the Autonomous University of Nayarit. He received the B.Tech. degree in electronics engineering from the Autonomous University of Guadalajara (UAG), in 1993, and the M.Sc. and Ph.D. degrees in electrical engineering from the Center for Research and Advanced Studies (CINVESTAV-IPN) in Guadalajara City, in 2001 and 2009, respectively. He is a member of the National System of Researchers (SNI). His research interests include IP-Traffic Modeling and design of concise behavior models for Entropy-based Intrusion Detection Systems.

Cesar Vargas-Rosales received a Ph.D. in electrical engineering from Louisiana State University in 1996. Thereafter, he joined the Center for Electronics and Telecommunications at Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, Mexico. He is currently the Telecommunications and Microelectronics program Director at ITESM. Dr. Vargas is a member of the National System of Researchers (SNI) since 1997, and is the coauthor of the book Position Location Techniques and Applications. He has carried out research in the area of personal communication systems on CDMA, smart antennas, adaptive resource sharing, location information processing, and multimedia services. His research interests are personal communications networks, position location, mobility and traffic modeling, intrusion detection, and routing in reconfigurable networks. Dr. Vargas is the IEEE Communications Society Monterrey Chapter Head and has been a Senior Member of the IEEE since 2001.

Rafael Martinez-Pelaez received his Ph.D. from the Technical University of Catalonia (Spain) in 2010. He is an associate professor in the Department of Information Technology at Autonomous University of Ciudad Juarez, Mexico. He is a member of the National System of Researchers (SNI) of the National Mexican Science Council (CONACYT). His research interests include authentication technologies, smart cards, and security issues on electronic services.

Homero Toral-Cruz received Ph.D. and M.S. degrees in Electrical Engineering, Telecommunication option from Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Jalisco, Mexico, in 2006 and 2010, respectively. He received the B.Sc. degree in Electronic Engineering from "Instituto Tecnológico de la Laguna", Coahuila, Mexico in 2002. He is currently an Assistant Professor at Sciences and Engineering department in University of Quintana Roo, Mexico. Prior to holding this position, he served as an Assistant Research at Electrical Engineering department, Telecommunication section in CINVESTAV, Jalisco, Mexico. His research interest includes VoIP technologies, QoS and network measurements, Internet technologies, IP traffic modeling, network performance evaluation, security networks and WSN.

Alberto F. Martinez-Herrera finished his B.Sc. studies in Electronics and Telecommunications at Universidad Autónoma del Estado de Hidalgo in 2004. From 2005 to 2009, he worked at Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Monterrey Campus, Mexico in several projects related to information security such as Intrusion Detection Systems and Applied Cryptography. Currently he is finishing a Ph.D. in Information Technologies and Communications in the same institute under the ITESM Scholarship Program and the CONACyT-Mexico Scholarship Program. He received the ISOC Fellowship Program grant to attend to the 63th IETF, held in Prague, during the spring of 2007. His research interests have been focused on areas related to applied cryptography, network security systems (secure protocols and intrusion detection systems), network topologies and bio-inspired computer techniques. Now he works on efficient hardware design techniques applied to cryptographic primitives and their resistance against side channel attacks.