An Architecture for Intrusion Detection Based on an Extension of the Method of Remaining Elements

P. Velarde-Alvarado*[1], C. Vargas-Rosales [2], D. Torres-Roman [3], A. Martinez-Herrera [2]

[1]Universidad Autónoma de Nayarit,
Ciudad de la Cultura "Amado Nervo" Tepic, Nayarit. Mexico
*pvelarde@nayar.uan.mx
[2]Department of Electrical and Computer Engineering,
ITESM-Monterrey Eugenio Garza Sada 2501 Sur,
Monterrey, Nuevo Leon. Mexico
[3]Center for Investigation and Advanced Studies (CINVESTAV-IPN),
Av. Cientifica 1145, C.P. 44019,
Zapopan, Jalisco, Mexico

ABSTRACT
This paper introduces an Anomaly-based Intrusion Detection architecture based on behavioral traffic profiles created by using our enhanced version of the Method of Remaining Elements (MRE). This enhanced version includes: a redefinition of the exposure threshold through the entropy and cardinality of residual sequences, a dual characterization for two types of traffic slots, the introduction of the Anomaly Level Exposure (ALE) that gives a better quantification of anomalies for a given traffic slot and r-feature, an alternative support that extends its detection capabilities, and a new procedure to obtain the exposure threshold through an analysis of outliers on the training dataset. Regarding the original MRE, we incorporate the refinements outlined resulting in a reliable method, which gives an improved sensitivity to the detection of a broader range of attacks. The experiments were conducted on the MIT-DARPA dataset and also on an academic LAN by implementing real attacks. The results show that the proposed architecture is effective in early detection of intrusions, as well as some kind of attacks designed to bypass detection measures.

Keywords: Anomaly-based Intrusion Detection, Method of Remaining Elements (MRE), traffic profiling, entropy.

RESUMEN
Este artículo presenta una arquitectura para la detección de intrusiones basado en anomalías cuya base referencial son perfiles de comportamiento del tráfico creados con nuestra versión mejorada del Método de los Elementos Remanentes (MRE). Esta versión de MRE incluye lo siguiente: una redefinición del umbral de exposición a través de la entropía y remanencia de las secuencias residuales, una caracterización simultanea para dos tipos de ranura de tráfico, la introducción del nivel de exposición de anomalías (ALE) brinda una mejor cuantificación de las anomalías para un rasgo y ranura de tráfico determinado, un soporte alternativo que extiende las capacidades de detección, y un nuevo procedimiento para obtener el umbral de exposición a través de un análisis de valores atípicos del conjunto de datos de entrenamiento. La incorporación de las mejoras señaladas proporciona un método confiable con mayor sensibilidad en la detección de un rango más amplio de ataques. Los experimentos se realizaron empleando la traza de red MIT-DARPA y en una LAN académica usando ataques reales. Los resultados muestran que la arquitectura propuesta es efectiva en la detección temprana de intrusiones, así como de algunos ataques diseñados para evadir la detección.

1. Introduction

Network infrastructures have become critical because enterprise organizations and individuals depend on the Internet for their daily activities. However, such dependence has its risks. For instance, an interruption of the network services can cause severe problems such as financial losses, damage or theft of confidential data, damage reputation of an organization, etc., [1]. “Targeted attacks”, [2], are events that can lead to disruption of network services. In particular, targeted attacks are one of the biggest threats to security and operation of an organization. They can affect critical infrastructure and have the potential of putting the general public at risk.

Targeted attacks have a high prevalence in the network. The 2008 Computer Crime and Security Survey, conducted by the Computer Security Institute and the FBI shows that 27% of the companies surveyed were able to detect targeted attacks on their computer systems, [3].

Perimeter defenses, such as firewalls and Network-based Intrusion Detection Systems (NIDS), provide an important security measure against network attacks. However, they do not provide a sufficient level of protection for server-based applications. Since many applications communicate with each other and with end users over the Internet, application-level attacks will often penetrate a perimeter via a legitimate access point. Moreover, firewalls and NIDS are unable to inspect encrypted (SSL) traffic, which is not decrypted until it reaches the host [4]. While firewalls, router-based packet filtering, and NIDS are necessary components of a multi-layer security system, they are insufficient on their own [5,6].

To overcome these drawbacks, one promising approach makes use of entropy to obtain knowledge of the structure and composition of traffic, summarized by behavioral traffic profiles, [7-13]. This approach is being proposed as a good candidate in traffic analysis for the development of a new generation of NIDS. A recent proposal in this context is the Method of Remaining Elements (MRE), an entropy-based method that profiles the behavior of traffic slots, [14]. This method is applied to the early detection of worm attacks, port scans and DDoS attacks. MRE highlights the traffic slots where attacks occur by means of a parameter called the exposure threshold, β_r. However, an in-depth analysis of the method revealed that certain attack patterns cannot be detected. Hence, there are some drawbacks associated with MRE which are mentioned as follows:

1) The exposure threshold, β_r, characterizes the normal behavior of traffic slots for a given r-traffic feature (e.g., origin or destination address, origin or destination port) and a maximum slot duration, t_d. MRE learns this value through empirical observation of the *"sets of remaining elements"* in training datasets. Specifically, β_r depends only on the *maximum cardinality of the set of the remaining elements*, denoted as $\tilde{M}_r$. Traffic slots where the cardinality of the set of remaining elements is higher than $\tilde{M}_r$ are considered anomalous. However, attacks such as *ipsweep* present cardinalities even much lower than M_r hence, they cannot be detected by a threshold based on M_r.

2) The characterization with a fixed maximum slot duration, i.e., t_d, is unable to detect attacks whose Inter-Packet Time (IPT) is several times greater than the period of characterization. *Sasser.Worm* exhibits this type of behavior, and is considered as a strategy to avoid detection.

3) MRE uses an index to quantify abnormalities called PDT, which is the percentage of difference between actual cardinality of the set of the remaining elements and $\tilde{M}_r$. However, this index is insufficient to detect attacks whose behavior does not generate enough diversity in the traffic features (it implies specifically that entropy cannot be used). An example that fits this pattern is the PoD attack (*ping of death*).

4) We identified attacks where the cardinality of the set of remaining elements and even the entropy of the residual sequence do not provide the means for detection. These attacks generate high volumes of packets with identical features. Such features do not have an effect on the behavior of residual sequences. Therefore, MRE is unable to establish a status based on β_r. An example is the *back attack*.

5) With respect to the first drawback, it is also necessary to provide a procedure to obtain the exposure threshold β_r through modeling the normal behavior on the training dataset.

The main contribution of this paper is to propose a solution to the aforementioned disadvantages in order to design an extended version of MRE and integrate it to an anomaly-based intrusion detection architecture. In this sense, we propose to define β_r as a function of the entropy of residual

sequences and the maximum cardinality of the set of remaining elements, in particular, $\beta_r = H(\tilde{S}) / \log(\tilde{M}_r)$. The incorporation of entropy enables an additional support for the identification of anomalous behavior on traffic slots. In other words, β_r defines a locus that separates the abnormal behavior from the benign one based on the entropies and cardinalities of the residual sequences. We also propose a dual characterization based on two types of slots: short-time traffic slots (STTS) and long-time traffic slots (LTTS). STTS duration does not exceed 0.5 seconds and its purpose is to detect attacks with high volume of traffic in short and continuous periods. The size of this slot is sufficient to process and respond quickly to the existence of malicious traffic. In contrast, LTTS duration does not exceed 60 seconds. This larger slot has the ability to detect attacks with malicious traffic that contains very limited number of packets in each slot and is temporarily dispersed.

In addition, we propose an indicator index for a given i-traffic slot and r-feature, called the Anomaly Level Exposure (ALE_i^r). This index is employed by our algorithm to determine the type of approach required to set the status (i.e., abnormal or benign) of an i-traffic slot under analysis. Three approaches that can be used are: a) entropy of residual sequences and the cardinality of the set of remaining elements, b) cardinality of the set of significant elements, and c) length of the sequence of unitary cardinality. The last two approaches are integrated as a support for MRE that extends the detection capabilities by identifying patterns of behavior of the attacks mentioned above. Finally, we propose a method to obtain the exposure threshold based on a fixed point-like iterative process, and analysis of outliers on the training dataset.

Application of this procedure improves the accuracy and efficiency of traffic characterization compared with the approach used in, [14]. Therefore, the resulting behavioral profiles produce a better anomaly detection capability. Also, it is important to mention that the detection system that we propose is independent of the network architecture since it only needs to capture and process the packet features necessary to detect intrusion.

With respect to detection performance of our architecture, we considered two scenarios. In the first one, we implemented true worm attacks within an academic LAN that varied in their propagation rates as well as in their scanning techniques. In the second one, the scenario consisted in studying the performance detection by using the MIT-DARPA traces that were organized in two main groups, the Denial of Service (DoS) attacks such as PoD, neptune, back and smurf; and the Probes with attacks such as satan, portsweep and ipsweep. Our architecture detected all of them except the satan attack. This approach used by considering the MIT-DARPA traces allows us to set up a performance study under extremely valuable intrusion detection public domain datasets, which provide a performance benchmark for detection and prevention systems as mentioned in [15].

2. Network-based Intrusion Detection Systems

There are two main approaches to design Network Intrusion Detection Systems (NIDS): misuse (signature-based) detection and anomaly (behavior-based) detection [6, 16]. Signature-based (S-NIDS) e.g., Snort, [17], employ pattern recognition techniques, i.e., they have a database with the known attack signatures and match these signatures with the analyzed data, when one similarity is found an alarm is activated. Nevertheless, S-NIDSs are incapable of detecting attacks that are not represented in its knowledge base. Anomaly-based (A-NIDS) e.g., PAYL, [18], first builds the statistical model describing the network's normal behavior (i.e., a behavioral profile from training data), and then warns any behavior that deviates from the model. A-NIDS has the advantage over S-NIDS to detect new types of attacks (zero-day attacks) as soon as they appear, another advantage is the response time which is in the order of seconds contrasting with hours or even days that may require an S-NIDS. Anomaly detection has long been suggested as a promising approach to detect previously unknown attacks. However, it faces several challenges e.g., evasive attacks might try to confuse IDS with fragmented, encrypted, tunneled, or junk packets. Hence, a recent proposal is to use the principles of entropy to obtain knowledge about behavior of traffic

features, such as the source and destination IP packet addresses, the source and destination port numbers, the type of protocol, the number of bytes per packet, the time elapsed between packets, etc., to build a profile of the network's normal behavior. This profile serves as a baseline in detecting anomalies.

3. Measures of Entropy

MRE, proposed in [14], is a traffic profiling method that uses an entropy estimator called the Balanced Estimator-II and from this, the *proportional uncertainty*, (PU) is defined, which is a measure of uncertainty. β_r is a value of proportional uncertainty that characterizes the traffic slot's normal behavior for a given r-feature and a maximum duration slot, t_d. The measurements of uncertainty of the traffic slots are processed with respect to this threshold and subsidiary measurements to determine its status. The mathematical descriptions of these measures are presented in the following subsections.

3.1 Balanced Estimator-II

Consider a discrete dataset *X* of size *N* where a finite number *M* of elements form the dataset's alphabet, $A = \{x_1, x_2, ..., x_M\}$. Let n_k be the number of times the value x_k appears in the dataset, thus, we have $\sum_{k=1}^{M} n_k = N$. An estimate of Shannon's entropy can be obtained by the low-bias balanced estimator, [19], which is defined as

$$\hat{H}^{bal}(X) = \frac{1}{N+2}\sum_{k=1}^{M}\left[\left(n_k+1\right)\sum_{j=n_k+2}^{N+2}\frac{1}{j}\right]. \qquad (1)$$

The second summation in (1) can be represented as a partial harmonic series or equivalently, as a harmonic number. A harmonic number is a number of the form

$$\mathbf{H}_n = \sum_{k=1}^{n}\frac{1}{k}\ . \qquad (2)$$

A harmonic number of the form (2) can be expressed analytically as

$$\mathbf{H}_n = \gamma + \Psi(n+1). \qquad (3)$$

where $\gamma = 0.5772156649$ is the Euler-Mascheroni constant, [20], and $\Psi(n+1)$ is the digamma function with corresponding asymptotic expansion that gives

$$\mathbf{H}_n \sim \log(n) + \gamma + (1/2)n^{-1} - (1/12)n^{-2} + (1/120)n^{-4} - (1/250)n^{-6} + \cdots \qquad (4)$$

By rearranging the second summation in (1), and using the definition of a harmonic number in (2), we may express it in terms of two harmonic numbers,

$$\sum_{j=n_k+2}^{N+2}\frac{1}{j} = \sum_{j=1}^{N+2}\frac{1}{j} - \sum_{j=1}^{n_k+1}\frac{1}{j} = \mathbf{H}_{N+2} - \mathbf{H}_{n_k+1}, \qquad (5)$$

using (4), this difference in harmonic numbers can be represented as

$$\mathbf{H}_{N+2} - \mathbf{H}_{n_k+1} = \log\left(\frac{N+2}{n_k+1}\right) + \rho_{N+2} - \rho_{n_k+1}, \qquad (6)$$

where ρ_{N+2} and ρ_{n_k+1} approach the same value *C* when *N* and n_k increase indefinitely, hence the difference $\rho_{N+2} - \rho_{n_k+1}$ approaches zero. A more computationally efficient expression for (1) can be obtained by replacing the second summation in (1) by (6) to get the Balanced Estimator-II,

$$\hat{H}^{bal-II}(X) = \frac{1}{N+2}\sum_{k=1}^{M}\left(n_k+1\right)\log\left(\frac{N+2}{n_k+1}\right). \qquad (7)$$

The condition that maximizes (7) occurs when x_k's frequencies are minimal, i.e., $n_k = 1$ where $k = 1, 2, \cdots, M$; under this condition, the size of the alphabet is equal to the size of the dataset, i.e., $M = N$. Therefore, the maximum value of (7) is given by

$$\hat{H}_{MAX}^{bal-II}(X) = \frac{2M}{M+2}\log\left(\frac{M+2}{2}\right). \qquad (8)$$

3.2 Proportional Uncertainty

The proportional uncertainty, *PU*, [14], is an index of uncertainty regarding the maximum value of Shannon´s entropy in a dataset. For a discrete dataset *X*, *PU* is defined as,

$$PU(X) = \frac{\hat{H}^{bal-II}(X)}{\log(M)} \leq \lim_{M\to\infty} \frac{\frac{2M}{M+2}\log\left(\frac{M+2}{2}\right)}{\log(M)} = 2, \text{ for } M > 1 \qquad (9)$$

$PU_{\text{Max-rel}}$ is defined as the maximum PU for a given alphabet size M, its value is given by

$$1 \leq PU_{\text{Max-rel}}(X) = \hat{H}_{MAX}^{bal-II}(X)/\log(M) \leq 2. \qquad (10)$$

Values of $PU_{\text{Max-rel}}$ for different alphabet sizes are shown in Figure 1. $PU_{\text{Max-rel}}$ determines the upper limit of the exposure threshold β_r for a given alphabet size *M*.

4. The Enhanced Method of Remaining Elements

Let the analyzed *i*-th traffic slot be described by the $\mathbf{S}_i^1$, $\mathbf{S}_i^2$, $\mathbf{S}_i^3$ and $\mathbf{S}_i^4$ sequences, which represents the source IP address, destination IP address, source port, and destination port, respectively, of the W_i packets arriving during the *i*-th traffic slot. Traffic slots do not overlap each other and have maximum slot duration of t_d seconds. $\mathbf{S}_i^r$ sequences, where $r = 1, 2, 3, 4$, constitute the *input sequences* to the MRE algorithm. MRE applies to them an iterative process of removal of its *significant elements* (significant elements are those with the higher frequency). While the

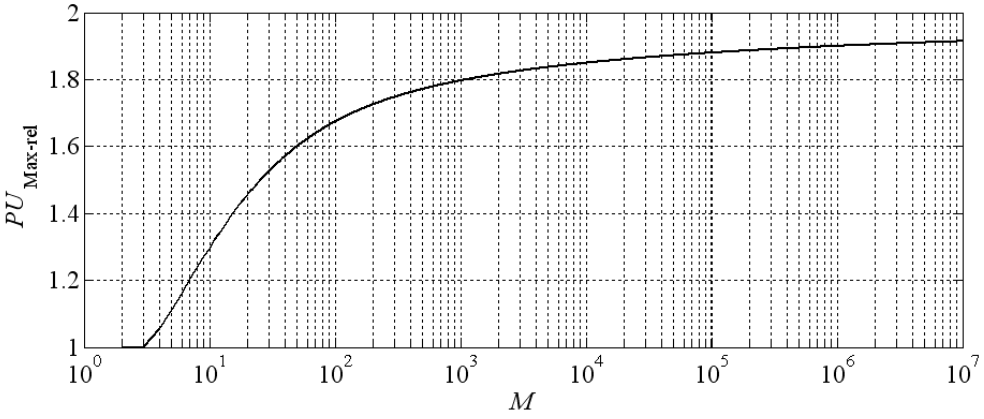

Figure 1. PU's maximum value for different alphabet sizes.

iterative removal of significant elements is carried out, the input sequence is identified as *sequence in progress*, and is denoted by $\mathbf{S}_{i,j}^r$, where *j* is the iteration number. The resulting sequence, when the algorithm stops is called the *residual sequence* and is denoted by $\tilde{\mathbf{S}}_i^r$. The cardinality of the set of the remaining elements in the residual sequence is denoted by R_i^r, where $R_i^r \geq 1$.On the other hand, the cardinality of the set of the significant elements, i.e., those not belonging to $\tilde{\mathbf{S}}_i^r$ is denoted by I_i^r, where $I_i^r \geq 0$.The *j*-iterative process of removing significant elements is performed while two conditions are met: 1) the proportional uncertainty of the sequence in process is less than an exposure threshold β_r, i.e., $PU(\mathbf{S}_{i,j}^r) \leq \beta_r$, and 2) the alphabet of the sequence in process is greater than two, this is $|\mathbf{S}_{i,j}^r| > 2$.Figure 2 shows Algorithm 1 to determine $\tilde{\mathbf{S}}_i^r$, R_i^r, and I_i^r, given an input sequence $\mathbf{S}_i^r$ and an exposure threshold β_r. We can say that the exposure threshold β_r is the level of proportional uncertainty *that might reach* a sequence in process during the removal process of significant elements.

There are two cases where a sequence in process does not reach this threshold; the first case occurs on sequences with alphabet size $|\mathbf{S}_i^r| = 1$; for these sequences $R_i^r = 1$. The second case occurs when the input sequences have low diversity in such a way that the entropy of the sequence in process, i.e., $\hat{H}^{bal-II}(\mathbf{S}_{i,j}^r)$, is not increased as its

```
1:  Parameters: S_i^r , β_r
2:  Items = |S_i^r |
3:  if (Items = = 1)
4:     I_i^r = 0 , R_i^r = Items
5:  else
6:     compute PU( S_i^r , β_r )
7:     if ( PU ≥ β_r )
8:        I_i^r = 0, R_i^r = Items
9:     else
10:       build table T
11:       sort T // decreasing order
12:       PU = 0, j = 1, S_{i,j}^r = S_i^r
13:       while (PU <= βr && |S_{i,j}^r | > 2 ) do
14:          S_{i,j}^r = S_{i,j}^r \ T( b ( j ) ) // remove j element
15:          compute PU ( S_{i,j}^r , βr )
16:          j++
17:       end while
18:       I_i^r = j − 1
19:       R_i^r = Items − I_i^r
20:       S̃_i^r = S_{i,j}^r
21:    end if
22: end if
```

Figure 2. Algorithm 1 to obtain $\tilde{\mathbf{S}}_i^r$, R_i^r, and I_i^r given a input sequence $\mathbf{S}_i^r$ and a β_r. Table T, consists of *(a, b)* value pairs, *a* means frequency and *b* is a particular *r*-instance (i.e., a IP address or port number value).

significant elements are removed, i.e., the relation $\hat{H}^{bal-II}(\mathbf{S}_{i,j}^r)/\log(R_{i,j}^r) \geq \beta_r$ is never fulfilled. Under this condition R_i^r inexorably becomes $R_i^r = 2$. Both the maximum traffic slot duration and the exposure threshold determine the values of $\tilde{\mathbf{S}}_i^r$, R_i^r, and I_i^r, i.e., changing β_r or t_d can lead to different values of them, this is because β_r determinates the separation condition of the significant elements of $\mathbf{S}_{i,j}^r$ and moreover, the value of β_r was obtained for a given t_d in the training phase.

4.1 Exposure of Anomalies and MRE Support

Exposure of anomalies aims to highlight the traffic slots whose residual sequences, $\tilde{\mathbf{S}}$ exceed a certain level of entropy $H(\tilde{\mathbf{S}})$ for a given residual alphabet size $\tilde{M}_r$. The boundary separating the normal from the anomalous behavior is given by the exposure threshold, β_r. This threshold is a behavioral traffic profile that is learned during the training phase by processing traffic slots with a maximum duration of t_d seconds. β_r is described by a two variable discrete function, where the proportion of entropy and alphabet size of the residual sequences satisfy

$$H(\tilde{\mathbf{S}})/\log(\tilde{M}_r) = \beta_r . \qquad (11)$$

For a given β_r, there are pairs of values $H(\tilde{\mathbf{S}})$ and $\tilde{M}_r$ associated with residual sequences that satisfy Equation (11), such values define a curve or function. Figure 3 presents in a semilog plot several functions of exposure thresholds. In general, each function defines the conditions of both alphabet size and entropy of residual sequences to determine the analyzed sequence's status (i.e., benign or anomalous).

The Anomaly Level Exposure, *ALE* in a traffic slot *i* and *r*-feature is defined as

$$ALE_i^r = R_i^r - 2 . \qquad (12)$$

This indicator index summarizes the behavior of the sequences in terms of exposure of anomalies. More concretely, in typical traffic (i.e., free of attacks) and typical working hours, the residual sequences satisfy $PU(\tilde{\mathbf{S}}_i^r) < \beta_r$ with residual cardinalities $R_i^r = |\tilde{\mathbf{S}}_i^r| = 2$ and $ALE_i^r = 0$, hence they are considered anomaly free (under reserve).

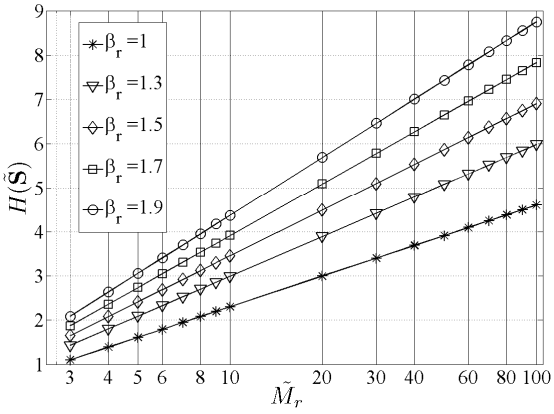

Figure 3. The $\tilde{M}_r$ and $H(\tilde{\mathbf{S}})$ relation with different exposure thresholds.

In anomalous conditions, the residual sequences have a proportional uncertainty such as $H(\tilde{\mathbf{S}}_i^r)/\log(R_i^r) \geq \beta_r$, with residual cardinalities $R_i^r = |\tilde{\mathbf{S}}_i^r| > 2$ and $ALE_i^r > 0$, this "highlights" or exhibits its anomalous condition. Finally, when the residual sequences have cardinalities $R_i^r = |\tilde{\mathbf{S}}_i^r| = 1$, and hence $ALE_i^r = -1$, it is not possible to define the status condition by means of ALE_i^r.

MRE support addresses the two issues outlined above: the "under reserve" and unitary cardinalities. The first relates to detecting attacks whose behavior does not allow detection by the

ALE index. Specifically, this occurs when malign traffic does not affect the residual sequence's properties but its counterpart (the one formed by significant elements). Therefore, we define a behavioral profile for I_i^r by a threshold for the maximum value of I_i^r which is denoted as I_r. For the other case, similarly we defined a behavioral profile for the length of the unitary cardinality sequences, U_i^r. The profile is the threshold of its maximum length which is denoted as U_r.

4.2 Exposure Threshold Algorithm

The exposure threshold is obtained through knowledge of the network using training datasets comprised of typical traffic traces. The minimum recommended size of the dataset is a week of typical traffic. The traffic traces analyzed belong to many open traffic sessions of different origin-destination pairs. Such traces can contain contiguous or non-contiguous messages due to the aggregation suffered through the network nodes.

Figure 4 shows the algorithms 2 and 3 to obtain β_r through fixed-point iterations using these datasets. In algorithm 2, the input sequence $\mathbf{S}_i^r$ given a maximum slot duration t_d, is introduced to MRE with feedback using an initial cutoff threshold $\beta_B = \beta_{initial}$ to return the residual sequence's *PU* and then feed back again if $\beta_A > \beta_B$. The initial cutoff threshold is defined as $\beta_{initial} = 1$. The objective of the algorithm is to determine the maximum cutoff threshold (i.e., the exposure threshold) for the sequence $\mathbf{S}_i^r$. Algorithm 3 delivers the value of proportional uncertainty required by algorithm 2, and seeks to raise the entropy of the sequence in process by extracting significant elements to reach the cutoff threshold β_B. Finally, the maximum cutoff threshold for the sequence $\mathbf{S}_i^r$ is reported as β_i.

Algorithm 2

```
1:  % Input data
2:  βB = 1.0;   % initial β
3:  X = S_i^r ;  % input sequence
4:  t_d ; % maximum slot duration
5:  βA = fnbeta ( X, βB );
6:  while ( βA > βB )
7:          βB = βA ;
8:          βA = fnbeta ( X, βA );
9:  end
10: βi = max( βA, βB );
```

Algorithm 3

```
1:  function B = fnbeta ( S_i^r , β )
2:  if  |S_i^r| = = 1  ||  |S_i^r| = = 2
3:      B = 0;
4:  else
5:      PU = H( S_i^r ) / log ( |S_i^r| );
6:      j = 0;  S_{i,j}^r = S_i^r ;
7:      while ( PU ≤ β  && |S_{i,j}^r|>2 )
8:          j++ ;
9:          S_{i,j}^r = S_{i,j-1}^r \ T(b(j)) ;
                        %removing
10:     PU(S_{i,j}^r) = H(S_{i,j}^r) / log(| S_{i,j}^r |) ;
11:     end
12:     B= PU;
13: end
```

Figure 4. Algorithms used to obtain the exposure thresholds.

The processing of the sequences $\mathbf{S}_i^r$ belonging to a given training traffic trace generates a vector of β_i -thresholds denoted as $\boldsymbol{\beta}_r^{\text{trace}} = [\beta_1\ \beta_2\ \cdots\ \beta_m\ ]$, where *m* is the number of traffic slots that form the trace. Vector $\boldsymbol{\beta}_r^{\text{trace}}$ is statistically analyzed to

determine the exposure threshold of the whole traffic trace. This process is summarized in Figure 5. The first stage decimates the input vector $\boldsymbol{\beta}_r^{\text{trace}}$ to remove the values that have a threshold $\beta_i = 1$, which is the minimum value of *PU* in the context of exposure of anomalies; the new vector is denoted as $\boldsymbol{\beta}_r^{D1}$. Stage two explores the existence of outliers in $\boldsymbol{\beta}_r^{D1}$; if there are no outliers, $\boldsymbol{\beta}_r^{D1}$ is transmitted to stage five, otherwise, an analysis of outliers (stage three and four) in $\boldsymbol{\beta}_r^{D1}$ is performed. Stage three sets the threshold for outliers, represented by the upper limit Ls. In this way, the thresholds $\beta_i \in \boldsymbol{\beta}_r^{\text{trace}}$ such that $\beta_i > Ls$ are regarded as outliers. The upper limit is calculated as

$$Ls = Q_3 + w(Q_3 - Q_1), \tag{13}$$

where Q_1 and Q_3 are the first and third quartile, respectively, w is maximum whisker length, the default 1.5 corresponds to approximately $\pm 2.7\sigma$, [21], where σ is the standard deviation of $\boldsymbol{\beta}_r^{D1}$. Stage four decimates the vector $\boldsymbol{\beta}_r^{D1}$ to remove all values below Ls, thus a new vector is formed, denoted as $\boldsymbol{\beta}_r^{D2}$ which contains the outliers of $\boldsymbol{\beta}_r^{D1}$. The fifth stage processes the decimated vectors either $\boldsymbol{\beta}_r^{D1}$ or $\boldsymbol{\beta}_r^{D2}$ to be characterized by a percentile (e.g., 95%). The β_r^{trace} values generated by each training trace, finally are averaged to obtain the exposure threshold for the respective intrinsic r-feature, i.e.,

$$\beta_r = \overline{\beta}_r^{\text{trace}}. \tag{14}$$

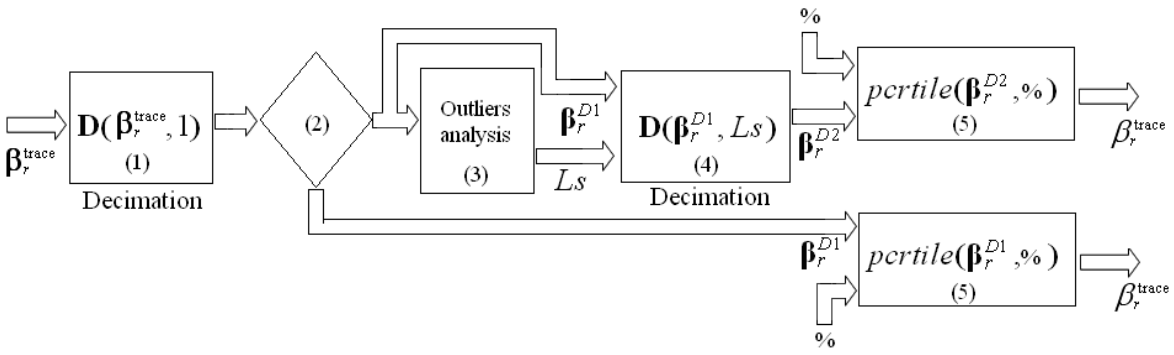

Figure 5. Stages to obtain β_r^{trace}.

4.3 MRE-based Intrusion Detection System Architecture

The architecture for an A-NIDS based on MRE is summarized in Figure 6. The training phase is responsible for building the behavioral profiles for the two types of defined traffic slots: STTS and LTTS. The profiles for a given r-feature are represented by the following parameters: 1) exposure threshold, β_r, 2) threshold for the maximum value of I_i^r, denoted as I_r, and 3) threshold for the maximum value of U_i^r, denoted as U_r. Experimentally, we found that the best performance detection was achieved with a maximum slot duration, $t_d = 0.5$ seconds for STTS, and $t_d = 60$ seconds for LTTS. Having defined the profiles the next step is to obtain current traffic measurements in order to determine the status of traffic slot i. The measurements are: 1) Cardinality of the set of the remaining elements, R_i^r. 2) Cardinality of the set of the significant elements, I_i^r. 3) Length of the unitary cardinality sequence, U_i^r. The diagnosis defines the status of analyzed i-th traffic slot based on any of the following approaches for detection: a) R_i^r and $H(\tilde{\mathbf{S}}_i^r)$ versus $\tilde{M}_r$ and $H(\tilde{\mathbf{S}})$. b) I_i^r versus I_r. c) U_i^r versus U_r. ALE_i^r defines the selection of the approach to use. A deviation of the measurement with respect to the profile is regarded as an anomaly.

One of the most valuable features of the presented architecture is that it does not need to compare chunks involving a fixed number of contiguous messages in order to detect anomalies; also it does not need to analyze chunks with the same number of messages in order to detect anomalies.

5. Experimental Platform, Dataset, and Tools

The evaluation of MRE was conducted in two different scenarios: the first scenario (labeled as SC1) is an academic LAN which is subdivided into four subnets (192.168.1.0, 192.168.2.0, 192.168.4.0, and 10.253.253.0). There are 100 hosts running Windows XP SP2 mainly. One router (192.168.1.1) connects the subnets with 10 Ethernet switches and 18 IEEE 802.11b/g wireless access points. The data rate of the core network is 100Mbps. A sector of the network is left vulnerable for worm propagation experiments, with ten not patched Windows XP stations (192.168.1.104 – 113). In the experiments Blaster, Sasser, and Welchia worms were released in the vulnerable sector. The scanning port attack was observed on the proxy server (192.168.4.253). The dataset was collected by a network sniffer tool based on libpcap library used by *tcpdump*, [22]. The dataset contains traces corresponding to 30 days of standard traffic in user's typical working hours. These traces were arranged into five datasets (SCx-D1 to SCx-D5) comprised of six traces each (SCx-Dy-01 to SCx-Dy-06), to be used for training purposes. The trace files in this collection contain TCP traffic and a total of 32.6 million packets. In addition, the dataset SC1-D6 is comprised of four traces, three traces correspond to worm attacks, and the last one to a portscan

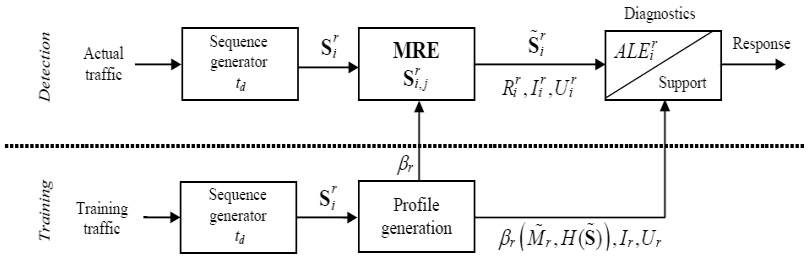

Figure 6. Architecture for an A-NIDS based on MRE.

attack. All traces were cleaned to remove spurious data using *plab*, a platform for packet capture and analysis, [23]. Traces were split into segments using *tracesplit* which is a tool that belongs to *Libtrace*, [24]. The traffic files in ASCII format suitable for MATLAB® processing were created with *ipsumdump*, [25]. The second scenario (SC2), is based on a sub-set of the 1998 MIT-DARPA data, [26], public benchmark for testing NIDS, adds six more attacks to our experiments. Table 1 gives an overview of the attacks in their respective scenarios that were evaluated in this paper.

Attack	Description	Trace
Portscan	Reconnaissance from forged (spoofed) addresses used to discover open ports	SC1-D6-01
Blaster	Computer worm that propagates by exploiting the Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability	SC1-D6-02
sasser	Computer worm that attempts to exploit the vulnerability in LSASS. It spreads by scanning the randomly selected IP addresses for vulnerable systems.	SC1-D6-03
welchia	Computer worm that exploits multiple vulnerabilities, including: DCOM RPC vulnerability, and the WebDav vulnerability	SC1-D6-04
pod	Denial of service ping of death	SC2-D1-01
smurf	Denial of service ICMP echo reply flood	SC2-D1-02
neptune	Syn flood denial of service on one or more ports	SC2-D1-03
portsweep	Surveillance sweep through many ports to determine which services are supported on a single host	SC2-D1-04
ipsweep	Surveillance sweep performing either a port sweep or ping on multiple host addresses	SC2-D1-05
back	Denial of service attack against apache webserver where a client requests a URL containing many backslashes	SC2-D1-06

Table 1. Description of the attacks.

Trace	β_r^{trace} (*srcIP*)	β_r^{trace} (*dstIP*)	β_r^{trace} (*srcPrt*)	β_r^{trace} (*dstPrt*)
SC1-D5-01	1.276	1.344	1.465	1.448
SC1-D5-02	1.379	1.409	1.497	1.482
SC1-D5-03	1.342	1.421	1.539	1.49
SC1-D5-04	1.354	1.379	1.517	1.577
SC1-D5-05	1.372	1.398	1.509	1.455
SC1-D5-06	1.349	1.379	1.53	1.49
$\beta_r = \overline{\beta}_r^{\text{trace}}$	**1.345**	**1.388**	**1.510**	**1.490**

Table 2. Traffic traces processed of dataset SC1-D5 to obtain β_r, for $t_d = 0.5$ s.

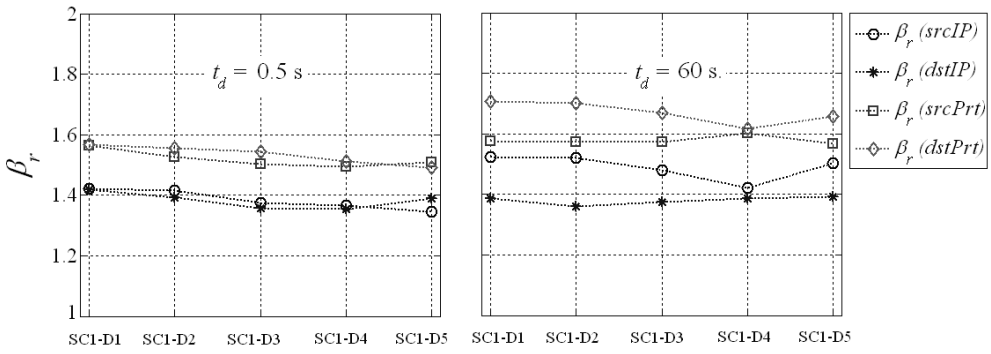

Figure 7. β_r for the training datasets in SC1, where (a) $t_d = 0.5$ s, and (b) $t_d = 60$ s.

6. Obtaining the Exposure Threshold

Table 2 shows the results of the dataset SC1-D5 processing using STTS. The exposure threshold β_r is obtained by means of the average β_r^{trace} in every r-feature. Figures 7a and 7b summarize the evolution (five weeks) of the exposure thresholds for the five training datasets that comprise the scenario one (SC1), obtained for STTS and LTTS i.e. $t_d = 0.5$. s and $t_d = 60$ s.

7. Detection of Attacks

Datasets SC1-D6 and SC2-D1 belonging to scenarios 1 and 2, respectively, were used to evaluate the performance detection of our proposed architecture. Table 3 summarizes the anomalies caused by the activity of the attacks and those that were detected by our architecture; the first and second columns are the number of slot and *r*-feature compromised by the attack. The third column indicates the type of traffic slot in which the detection was achieved. The fourth column gives the value or range of *ALE* corresponding to the compromised traffic slots. The fifth and sixth columns are support parameters. The seventh column indicates the detection approach used. The eighth column indicates type of traffic by protocol where the attack was present. The ninth column indicates the scenario where the attack took place. Finally, the tenth column gives the name of the attack.

Compromised		Slot type	*ALE range*	I_i^r	U_i^r	Approach detection	Protocol	scenario	Attack
i	*r*								
95 to 162	1, 4	STTS	13 to 44	-	-	1	TCP	SC1	portscan
195 to 4521	2, 4	STTS	22 to 336	-	-	1	TCP	SC1	blaster
1 to 33	1, 2	LTTS	125 to 2334	-	-	1	TCP	SC1	sasser
42 to 3875	2	STTS	49 to 452	-	-	1	ICMP	SC1	welchia
2 to 71	1	STTS	326 to 503	-	-	1	ICMP	SC2	smurf
5550	3	STTS	12	-	-	1	TCP	SC2	neptune
7743	3, 4	STTS	65	-	-	1	TCP	SC2	portsweep
607 to 698	3, 4	LTTS	18 to 27	-	-	1	TCP	SC2	ipsweep
10 and 11	1, 2	STTS	-1	-	406, 44	3	ICMP	SC2	pod
328 to 338	3, 4	LTTS	NA	89 to 103	-	2	TCP	SC2	back

Table 3. Summary of the analyzed attacks.

For the detection of the portscan attack, it was sufficient to use STTS. In particular, 25 traffic slots located between slots 95 and 162 had anomalous behavior with $ALE_i^r > 0$. These anomalies were caused by a portscan attack by using spoofed source IP addresses; it was a targeted attack against the proxy server. Figure 8a shows a R_i^r plot for the feature $r = 1$ (i.e., source IP address), we can see that the attack is exposed during the compromised traffic slots and subsequently traffic returns to its normal behavior,because $ALE_i^r = 0$.

Figure 8b shows a R_i^r plot for the same attack, but now for the compromised feature $r = 4$, similarly the attack is exposed with $ALE_i^r > 0$. However, we can appreciate in Figure 8b four anomalous slots, but a forensic analysis showed that it was benign traffic scan. In this case, the activity of attack as soon as it appears, it produces changes in the behavior of the traffic slot, consequently it generates the anomaly and, hence an alert can be issued.

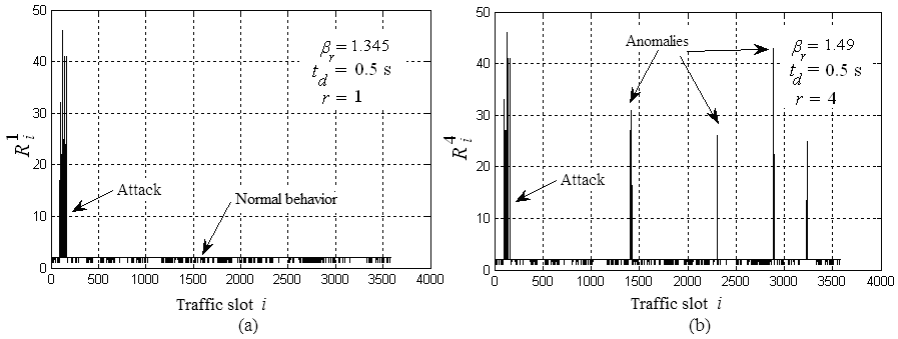

Figure 8. Portscan attack detected in trace SC1-D6-01 in features: (a) srcIP, and (b) dstPrt.

Another attack in SC1 is shown in Figure 9, that corresponds to the Blaster worm attack, the affected features are $r = 2$ and $r = 4$, and their exposure levels are higher than the port scan attack, this attack lasted 38 minutes and similarly its detection was handled with STTS. The following attacks: *sasser, welchia, smurf, neptune, portsweep,* and *ipsweep* were also timely detectable by the level of exposure as shown in Table 3.

On the other hand, in scenario 2 some attacks cannot be directly detected by *ALE*, in particular, *pod* and *back* attacks, under such situations the support to MRE is used. The anomaly detection is based on the maximum length of the unitary cardinality sequences, denoted by U_r. It is also supported by the typical level of significance, denoted by I_r. Our empirical observations on the training datasets allowed us to define the thresholds for the two above parameters as $U_r = 5$ and $I_r = 42$.

The anomalies caused by Ping of death *(pod)* attack generated two very long sequences of unitary cardinality and consequently an $ALE = -1$. In particular, $U_{i=10}^{r=1,2} = 406$, and $U_{i=11}^{r=1,2} = 44$, the

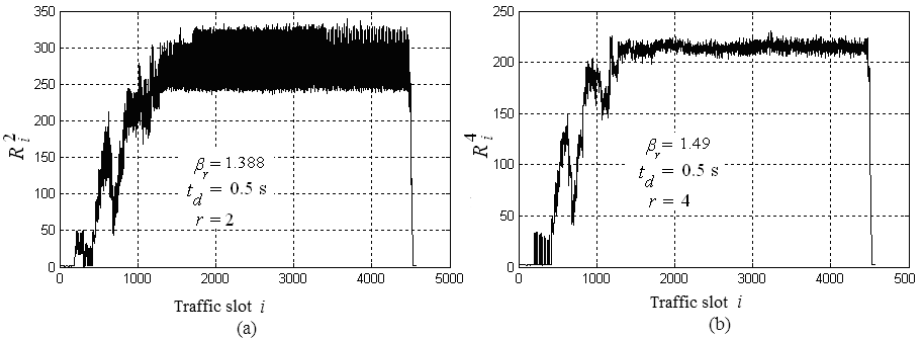

Figure 9. Blaster worm attack detected in features: (a) srcIP, and (b) dstPrt.

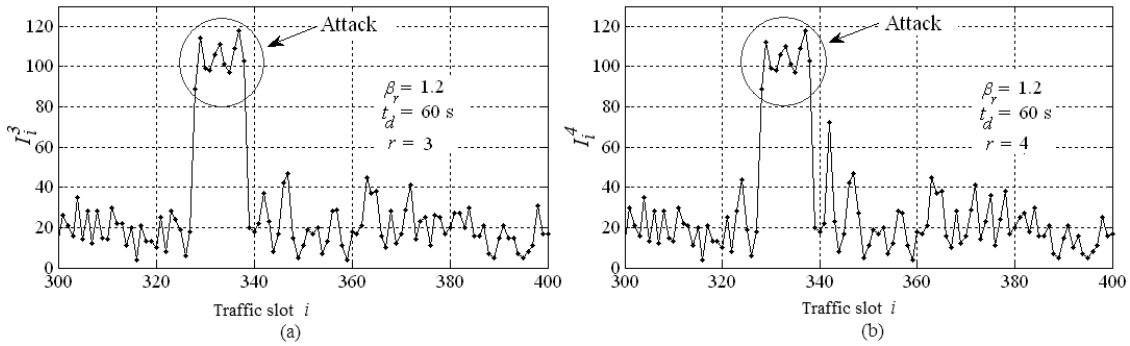

Figure 10. Back attack detected by a significance analysis in features: (a) srcIP, and (b) dstPrt.

compromised features are $r = 1$ and $r = 2$, respectively. U_i^r in both traffic slots had a deviation from the value defined by $U_r = 5$, this indicates the anomaly and the consequent detected attack.

The *back* attack presents characteristics that also require the support of MRE, back is an attack of DoS, where there is an attacker (135.8.60.182) and a target (172.16.114.50). Each attack session lasts 0.162 seconds and consists of 61 packets. The approximate total duration of the attack is 11 minutes. Features $r = 3$ and $r = 4$ are affected, however, because the attack generates a large volume of packets related to these features, the anomalous attacks do not appear in any residual sequence, and therefore the anomalies are absorbed by the significant component I_i^r. An analysis of significance elements allowed us to discover the deviation from the threshold $I_r = 42$ in the compromised slots, as shown in Figure 10 and Table 3.

8. Conclusions

We have presented an architecture developed to perform traffic profiling and intrusion detection based on our enhancement of the Method of Remaining Elements (MRE). Our experimental results indicate that our approach is very promising, and applicable to anomaly intrusion detection systems. In particular, the Anomaly-based NIDS proposed in this paper uses training datasets to characterize the behavior for two types of traffic slots in terms of an exposure threshold; a threshold for the cardinality of significant elements, and a threshold for length for unitary cardinality sequences. Thus, changes in the properties of sequences with malicious traffic can be detected as anomalies by using one of the three proposed approaches for detection. The experimental results carried out in two scenarios (an academic LAN and the MIT-DARPA dataset) showed that the traffic characterization by means of the exposure threshold (defined through an algorithm based on fixed-point iterations and a fixed point-like iterative process) provides a better sensitivity to detect intrusions with respect to the previous proposal. Furthermore, we achieved a significant improvement by incorporating an MRE support for the detection of other types of attacks which are not sensitive to the Anomaly Level Exposure *(ALE).* Future work aims to develop a hardware implementation for MRE.

References

[1] FLUKE NETWORKS. Security ROI – A financial view of network security. Application Note. 2005

[2] CounterStorm. White paper. Targeted Attack Technical Brief http://www.counterstorm.com/

[3] 2008 CSI Computer Crime and Security Survey; http://www.gocsi.com/

[4] Sana Security. http://www.sanasecurity.com/support/enterprise/pr/faq.php

[5] Nucci A., and Bannerman S., Controlled Chaos. IEEE Spectrum. Vol.44. No.12. Dec. 2007, pp. 42-48.

[6] Vacca J. R., Computer and Information Security Handbook. The Morgan Kaufmann Series in Computer Security by Elsevier Inc., 2009, pp. 41-42, 64

[7] Xu K., Zhang Z., and Bhattacharyya S., Internet Traffic Behavior Profiling for Network Security Monitoring, Transactions on Networking, IEEE/ACM . Vol. 16, No. 6, Dec. 2008, pp. 1241 – 1252.

[8] Ziviani A., Gomes A., and Monsores M., Network Anomaly Detection Using Nonextensive Entropy, IEEE Communications Letters, IEEE. Vol. 11, No.12, Dec. 2007, pp. 1034-1036.

[9] Wagner A., and Plattner B., Entropy Based Worm and Anomaly Detection in Fast IP Networks, Proc. of the 14th IEEE International WorksShop on Enabling Tech.: Infrastructure for Collaborative Enterprise, 2005, pp. 172 – 177, Linköping, Sweden, June.

[10] Lee W., and Xiang D, Information-theoretic Measures for Anomaly Detection, In Proc. of IEEE Symposium on Security and Privacy, 2001, pp. 130-143, Oakland, CA, USA, May.

[11] Velarde-Alvarado P., Vargas-Rosales C., Torres-Román D., and Muñoz-Rodríguez D., Entropy Based Analysis of Worm Attacks in a Local Network, Research in Computing Science, Vol. 34. May 2008, pp. 225-235.

[12] Velarde-Alvarado P., Vargas-Rosales C., Torres-Román D., and Martinez-Herrera A, Entropy-Based Profiles for Intrusion Detection in LAN Traffic, Advances in Artificial Intelligence: Algorithms and Applications, Research in Computing Science, Vol. 40, 2008, pp. 119-130.

[13] Nychis G., Sekas V., Andersen D., Kim H., and Zhhang H., An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. Internet Measurement Conference, ACM-SIGCOMM, 2008, pp. 151-156, Vouliagmeni, Greece, October.

[14] Velarde-Alvarado P., Vargas-Rosales C., Torres-Roman D., and Martinez-Herrera A, Detecting Anomalies in Network Traffic Using the Method of Remaining Elements. IEEE Communications Letters, Vol.13, No.6, June 2009, pp. 462-464

[15] Wang Y. Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection .Igi Global. 2009, pp. 70

[16]] Jajodia S., Intrusion Detection Systems, Advances in Information Security. Springer Science+Business Media, LLC., 2008.

[17] Roesch M., Snort - Lightweight Intrusion Detection for Networks. In: LISA '99: Proc., 13th USENIX Conference on System Administration, 1999, pp. 229-238

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

[19] Bonachela J. A., Hinrichsen H., and Muñoz M. A., Entropy Estimates of Small Data Sets, Journal of Physics A: Mathematical and Theoretical. No. 41. April, 2008.

[20] Conway J. H., and Guy R. K., The Book of Numbers, New York: Springer-Verlag, 1996, pp. 143 and 258-262.

[21] Tukey J.W., Exploratory Data Analysis, Addison-Wesley Series in Behavioral Science, 1977.

[22] Jacobson V., Leres C., and McCanne S., Tcpdump/libpcap. http://www.tcpdump.org/

[23] Peppo A., plab. Tool for traffic traces. http://www.grid.unina.it/software/Plab/

[24] Trac Project. Libtrace. http://www.wand.net.nz/trac/libtrace

[25] Kohler E., ipsumdump. Traffic tool. http://www.cs.ucla.edu/~kohler/ipsumdump

[26] Lincoln Laboratory, MIT. DARPA Intrusion Detection Data.

Authors´ Biographies

Pablo VELARDE-ALVARADO

Currently, he is researcher and full-time professor at the Area of Basic Sciences and Engineering of the Universidad Nacional Autónoma de Nayarit. He received the B. Tech. degree in electronics engineering from the Universidad Autónoma de Guadalajara (UAG), in 1993, and the M. of 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. His research interests include IP-Traffic Modeling and design of concise behavior models for Entropy-based Intrusion Detection Systems.

Cesar VARGAS-ROSALES

Dr. Cesar Vargas Rosales received a Ph.D. in electrical engineering from the 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.

Deni TORRES-ROMAN

He received a Ph.D. degree in telecommunication from the Technical University Dresden, Germany in 1986. He was professor at the University of Oriente, Cuba. Co-author of a book about data transmission. He was awarded the Telecommunication Research Prize in 1993 from AHCIET Association and was recipient of the 1995 best Paper Award from AHCIET Review, Spain. Since 1996, he has been an associate professor at the Center for Investigation and Advanced Studies (CINVESTAV-IPN). His research interests include hardware and software designs for applications in the telecommunication area. He is a member of the IEEE.

Alberto MARTINEZ-HERRERA

Currently he is pursuing a Ph.D in information technologies and communications at the Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, Mexico. He received the bachelor's degree in telecommunications from the Universidad Autónoma del Estado de Hidalgo. His main research interests are focused on areas related to cryptography and network security systems, mainly related to the DNS protocol.