Development an Anomaly Network Intrusion Detection System Using Neural Network

Prof. Dr. Kais Said Al-Sabbagh
Electric engineering dept.
University of Baghdad

Assist.prof. Hamid M. Ali
Computer engineering dept.
University of Baghdad

Elaf Sabah Abbas
M.Sc. student
Computer engineering dept.

Abstract

Most intrusion detection systems are signature based that work similar to anti-virus but they are unable to detect the zero-day attacks. The importance of the anomaly based IDS has raised because of its ability to deal with the unknown attacks. However smart attacks are appeared to compromise the detection ability of the anomaly based IDS. By considering these weak points the proposed system is developed to overcome them.

The proposed system is a development to the well-known payload anomaly detector (PAYL). By combining two stages with the PAYL detector, it gives good detection ability and acceptable ratio of false positive.

The proposed system improve the models recognition ability in the PAYL detector, for a filtered unencrypted HTTP subset traffic of DARPA 1999 data set, from 55.234% in the PAYL system alone to 99.94% in the proposed system; due to the existence of the neural network self-organizing map (SOM). In addition SOM decreases the ratio of false positive from 44.676% in the PAYL system alone to 5.176% in the proposed system.

The proposed system provides 80% detection ability of smart worms that are meant to invade the PAYL detector in the PAYL system alone, due to the existence of the randomization stage in the proposed system.

Keywords: Intrusion Detection Systems (IDS), PAYL, SOM, Randomization.

الخلاصه

معظم الـ(Intrusion Detection Systems) هي من نوع (Signature based) والتي تعمل بشكل مشابه الى مضادات الفايروسات ولكنها غير قادرة على التعرف على الهجمات التي تظهر لاول مره (الهجمات غير المدرجه في قاعده بياناتها) وقد ظهرت اهميه الـ(anomaly based IDS) نتيجه لقدرتها على اكتشاف مثل هذه الهجمات بالرغم من ذلك فانه ظهور الهجمات الذكيه اصبح يمثل تهديد الى الـ(Anomaly based IDS). تم تطوير النظام المقترح للتغلب على نقاط الضعف المذكوره سابقا.

النظام المقترح هو تطوير الى نظام الـ(PAYL) المعروف. بدمج مرحلتين مع كاشف الـ(PAYL) يتم الحصول على قدره كشف جيده و نسبه ايجابيه كاذبه (False positive) حسن النظام المقترح قابلية الـ(PAYL) للتعرف على الانماط، من 55.234% في الـ(PAYL system alone) الى 99.94% في النظام القترح، نتيجه لوجود الشبكه العصبيه.

وكذلك قلل وجود الـ(SOM) الـ(False positive) من 44.676% في الـ(PAYL system alone) الى 5.176% في النظام المقترح.

بسبب وجود مرحلة الـ(randomization) اظهر النظام المقترح قابليه على اكتشاف الـ(smart worms) والمصممه لغزو كاشف الـ(PAYL) في الـ(PAYL system alone) بنسبة 80%.

I. Introduction

Network Intrusion detection system (NIDS) is a type of security management system for computers and networks; it inspects all network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

The Internet and computer networks are exposed to an increasing number of security threats. With new types of attacks appearing continually, developing flexible and adaptive security oriented approaches is a severe challenge. In this context, anomaly-based intrusion detection techniques are a valuable technology to protect target systems and networks against malicious activities **[P. Garcı´a-Teodoro et. al., 2009]**. PAYL detector is statistical model that detects anomalies by capturing features of data payload in an efficient way -automatic and unsupervised-, it has been used before in the context of computer security in many researches and gave such a good results **[S. Zanero, 2005, KeWang, 2006, Ke Wang et. al., 2004, D. Bolzoni et. al., 2006]**.

The PAYL detector alone is considered a successful system to detect the abnormalities in the incoming traffic, of course after it has a proper training phase. In recent time, smart attacks **[C. Mao et.al., 2010]** have been dedicated to invade the PAYL detector, such as the mimicry attacks **[KeWang, 2006]**. To avoid such attacks, on the PAYL detector the randomized testing technique **[KeWang, 2006]** is used. This technique provides diversity on the packet payload that confuses the attacker about which byte positions to pad his exploit code in order to appear normal.

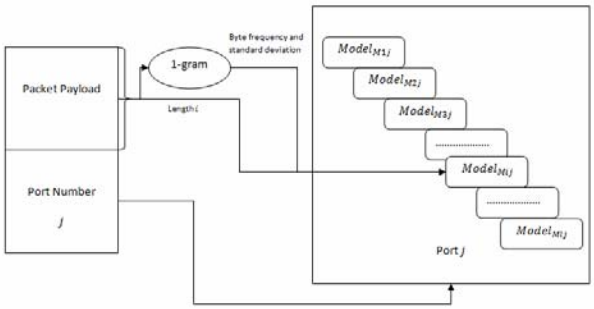
Instead of using the clustering technique that follows the PAYL detector (we call PAYL detector followed by clustering technique PAYL system alone), which is used to reduce the number of models that varies by one byte and the models that don't have enough training samples, a neural network Self- organizing map SOM is used. SOM is used for classification of the incoming traffic (data) into clusters. There are multiple reasons for choosing a SOM for classification purpose. The algorithm is robust with regard to the choice of the number of classes to divide the data into, and it is also resistant to the presence of outliers in the training data, which is a desirable property. In addition, the SOM have the best performance trade-off between speed and classification quality **[S. Zanero, 2005]**.

To build a highly performance anomaly based intrusion detection system, it is necessary that the proposed system consists of several stages where the combination of these stages is responsible for eliminating weak points that faces the PAYL system alone

II. Background

i. PAYL Detector: 1-gram Payload Modeling and Anomaly Detection

PAYL detector detects anomalies by combining an n-gram analysis algorithm with a classification method based on clustering of



packet payload**.** PAYL detector computes a set of models. One for each specific observed length i of each port j as shown in figure 1

Figure 1: Internal structure of PAYL detector

The distance between models is measured by using simplified Mahanalobis distance. The advantage of simplified Mahalanobis distance is that it takes into account not only the average value but also its variance and the covariance of the variables measured. Instead of simply computing the distance from the mean values, it weights each variable by its standard deviation and covariance, so the computed value gives a statistical measure of how well the new example matches (or is

consistent with) the training samples. And the whole computation time is linear in the length of the payload with a small constant to compute the measure. This produces an exceptionally fast detector **[KeWang, 2006]**.

ii. Self -organizing Map (SOM)

A Self-organizing Map (also called Self-Organizing Feature Map or Kohonen map), is a kind of artificial neural network. It works by emulating the cognitive classification process typical of the human brain and it is based on competitive learning **[S. Zanero, 2005]**. SOM learns to classify input vectors according to how they are grouped in the input space. They differ from competitive layers in that neighboring neurons in the self-organizing map learn to recognize neighboring sections of the input space. Thus, self-organizing maps learn both the distribution (as do competitive layers) and topology of the input vectors they are trained on **[Mat Lab Help]**.

The goal of learning in a SOM is accomplished by modifying the weight vector of neurons in a certain manner. To select which weight vectors must be updated and how, two parameters are used: the learning rate that controls the "influence" of the input on the neuron weight array components, and the update radius, that controls which neurons to modify. To accomplish the classification, a SOM goes through three phases: initialization, training, and classification **[S. Zanero, 2005, D. Bolzoni, 2009]**.

iii. Randomized Testing

Randomization is necessary to avoid mimicry attacks. Instead of modelling and testing the whole packet payload, the packets are randomly partitioned into several (possibly interleaved) substrings or sub sequences S1, S2, ..., SN, and model each of them separately against the same single normal model as shown in figure 2 (to make sure that partitions will produce different models from the one built using the whole payload) **[KeWang, 2006]**.

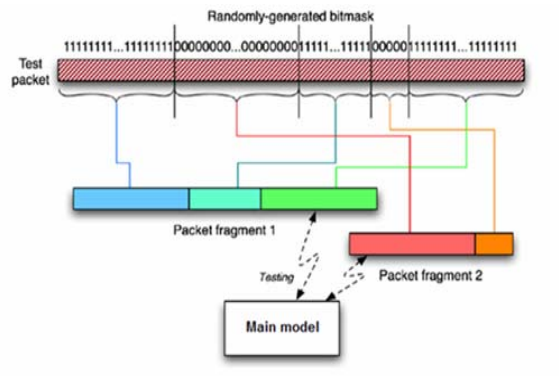


Figure 2: Randomized Testing.

To perform this randomization it is necessary to provide appropriated random binary bit mask which must be a purely random binary string, so that the following two criteria must be satisfied **[W. Stallings, 2005]**:

1- Uniform distribution.
2- Independence.

There are several advantages come from using this type of randomization, first all the calculations are liner that take a very short time to performed so it gives an ability to provide this system directly on line for training and testing. Second, because of using a pure binary random mask only two sub-models appeared and need to be compared with the model extracted from the entire packet payload, each of these models has distribution different from the original model. Third when the binary mask changed there is no need to retrain the entire system (because of the properties of the pure random binary mask). Still, there is the disadvantage of the false positive increment, but the false positive rate is usually much higher when the partitions have extremely unbalanced lengths **[KeWang, 2006]**.

III. The Proposed System Architecture

Each anomaly-based NIDS must consist of two phases, training phase and testing phase. Input data for both phases must pass through the same stages to get comparable information, which can be used for detecting any abnormalities.

The general structure of the proposed anomaly based NIDS is shown in figure 3, and the detailed structure are shown in figure 4.

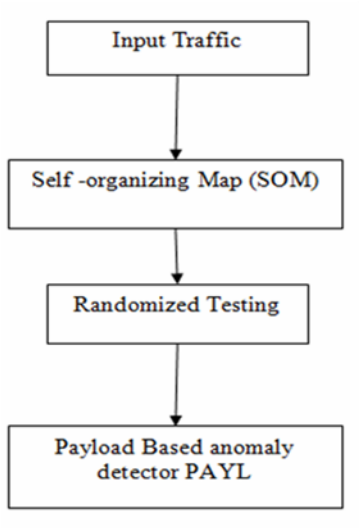


Figure 3: general structure of the proposed system

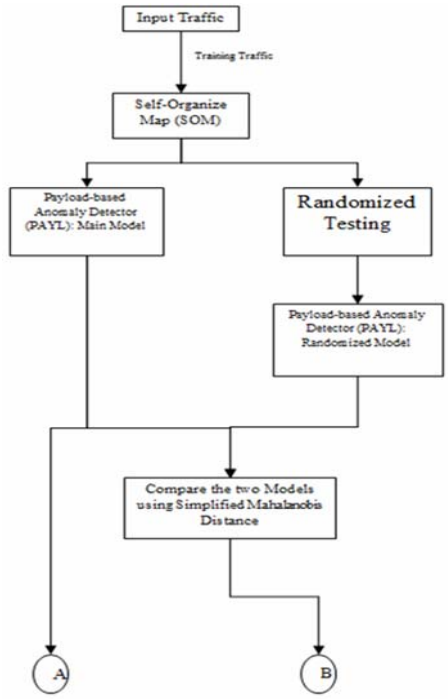


Figure 4 (a): Testing Phase.

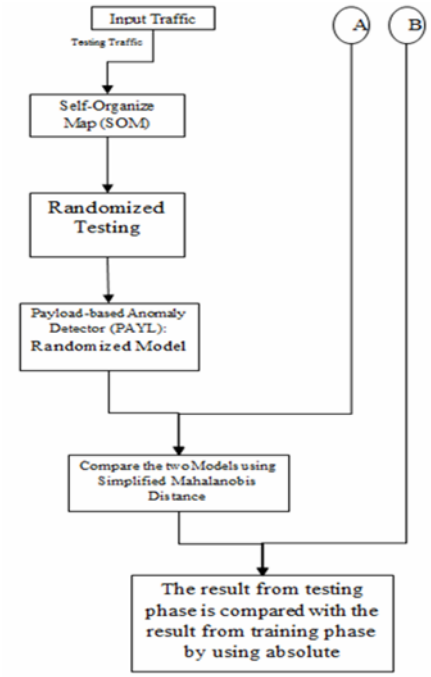


Figure 4 (b): Training Phase.

Figure 4: detailed structure of the proposed Anomaly NIDS.

Proposed Anomaly NIDS Implementation

A. Input Traffic

The input traffic that is used in this work is DARPA 1999 data set. The data set contains five weeks of collected data that are provided for IDS evaluation, three of them for training phase and two of them for testing phase. The first and third weeks of the training data do not contain any attacks. This data was provided to facilitate the training of anomaly detection systems. The second week of the training data contains a select subset of attacks from the 1998 evaluation in addition to several new attacks. The inside network traffic data is used which was captured between the router and the victims. Because most public applications on the Internet use TCP (web, email, telnet, and ftp), and to reduce the complexity of the work due to the huge amount of data, only the inbound TCP traffic at the port 80 of the hosts 172.016.xxx.xxx which contains

most of the victims at the first four weeks is examined. The traffic is filtered using Wireshark program, the filter is:
http&&(ip.dst>=172.16.0.0&&ip.dst<=172.16.255.255)&&data.

This filter provides all the conditions that are required:

- Hosts ip address in the range of 172.016.xxx.xxx
- The port number is 80 for HTTP
- The packets must contain data between 1 byte to 1460 bytes

All filtered packet payloads copied into a file. Then MATLAB program is used to transfer them from hexadecimal format into the decimal format of ASCII characters. After the extraction and transformation of all packets payloads, the training and testing phases can be started.

B. PAYL detector

The practical work of the PAYL detector is described in the following segments:

At first a program segment that describes the simplified Mahalanobis distance within the PAYL detector is:

p_{new}= is the incoming packet payload.

fv_{new}= is the feature vector of the incoming packet. It is a vector of length 256.

$meanij$= is the mean of the byte value distribution of the training model.

/* averaged feature vector of length 256 computed from the training examples */

σ_{ij} = is the standard deviation of the byte value distribution of the training model;

/* standard deviation of length 256 computed from the training examples */

α = is the smoothing factor determined previously.

Thr = is a previously defined threshold from training phase.

d= is the initial distance equal to zero;

/* Now to compute simplified Mahalanobis distance between the new incoming packet and the previously calculated model */

$$For\ k = 0\ to\ 255$$

$$d = d + \frac{fvp_{new}(k) - meanij(k)}{\sigma_{ij}(k) + \alpha}$$

$$End$$

$$If\ (d > Thr)$$

$$Then\ generate\ an\ alert;$$

$$End$$

Compute one model Mij for each observed length i of payloads sent to port j Such model might introduce several problems. First, the total size of the model can become very large. (The payload lengths are associated with media files that may be measured in gigabytes and many lengths may be defined causing a large number of centroids to be computed.) Further, the byte distribution for payloads of length i can be very similar to that of payloads of lengths $i-1$ and $i+1$; after all they vary by one byte. Another problem is that for some lengths, there may not be enough training samples. Sparseness implies the data will generate an empirical distribution that will be an inaccurate estimate of the true distribution leading to a faulty detector.

The solution for these problems is to "borrow" data from neighboring lengths to increase the number of samples. Two

neighboring models will be computed by using the simple Manhattan distance to measure the similarity of their average byte frequency distributions **[KeWang, 2006]**.

Then, a program segment that describes the work of the Clustering technique is:

FV= relative byte frequency values for a payload of length **i**.

/* feature vector is double vector between 0.0 and 255.0]*/

Thr=previously defined threshold

l = length of the longest packet payload

PAYLOAD = array [1 ..*l*] of [0 ..255]

$MODELS = [M1, M2, M3 \ldots ML]$

/* There are a model for each length */

For k = 1 to infinity

IF Manhattan distance between two

neighboring models < Thr

Then

Merge them

/* the operation repeated until no more models can be merged*/

We call The PAYL detector followed by clustering technique PAYL system alone. PAYL system alone is used as comparison system with the proposed Anomaly NIDS.

C. Self-organizing Map

A Self-organizing Map (also called Self-Organizing Feature Map or Kohonen map), is a kind of artificial neural network. It works by emulating the cognitive classification process typical of the human brain and it is based on competitive learning. To accomplish the classification, a SOM goes through three phases: initialization, training, and classification.

- **Initialization**: First of all, some parameters (number of neurons, learning rate, radius, and number of training samples) have to be fixed. The following segment shows the work of initialization phase:

α_0: is the initial learning rate.
$\alpha = \alpha_0$ is the current learning rate
Red_0: is the initial radius size.
$Red = Red_0$ is the current radius.
T: is the number of training samples (epochs).
D: is the input data, its range lies between $[d1 \ldots\ldots dn]$.
L: is the maximum length of the input data.
N: is the total number of neurons that arranged in two dimension array.
Wi: is the weight vector for i'th neuron
To initialize the weight vector for each neuron

For $i = 1$ *to* N
Wi = random values of length L in the range of D
End

- **Training**: The training phase consists of a number of iterations (also called epochs). Each training sample is used for one iteration only, thus the number of samples determines the number of iterations. The following segment shows the work of training phase:

For $i = 1$ *to* T
To find the wining neuron (BMU):
Win-distance=mathematical distance (Neuron1, x)
$Win - neuron = Neuron_1$
$for\ j = 1\ to\ N$
$Distance = mathematical\ distance\ (Neuron_j, x)$
$if\ (Distance \leq Win - distance)$
then
$Win - distance = Distance$
$Win - neuron = Neuron_j$
End
To find the neighboring neurons to the winner neuron:

$for\ j = 1\ to\ N$
If (trigonometric distance(win - neuron, neuronj) < Red)
then
Neuronj is a neighbor to the Win-neuron
End
To update the weight vectors of the neighboring neurons:
$for\ j = 1\ to\ number\ of\ neighboring\ neurons$
$for\ k = 1\ to\ L$
$W_j[k] = W_j[k] + \alpha * (x[k] - W_j[k])$
End
End
To update the learning rate and the radius values:
$Red = Red_0 * e^{-\frac{i}{ni}}$
$\alpha = \alpha_0 * e^{-\frac{i}{ni}}$
End

- **Classification**: To classify an input x, the SOM proceeds similarly to the training phase. The following segment shows the work of classification phase:

Win-distance=mathematical distance (Neuron1, x)
$Win - neuron = Neuron_1$
$for\ j = 1\ to\ N$
$Distance = mathematical\ distance\ (Neuron_j, x)$
$if\ (Distance \leq Win - distance)$
then
$Win - distance = Distance$
$Win - neuron = Neuron_j$
End

D. Randomized Testing

The following segment shows the randomization stage in the proposed anomaly NIDS:

Mask is the vectors [1…. l]
Mod1= is the first sub-model
Mod2= is the second sub-model
/* Mod1 and Mod2 are vectors [1…. l] */

For $i = 1$ *to* l

```
If (Mask (i) == 0)

then

Mod1 (i) = P (i)

else

Mod2 (i) = P (i)
/* to make the length of Mod1 and Mod2 ==L the rest of the packet will Stuffed by Null */
```

IV. The Proposed System versus the PAYL System Alone

The proposed system is applied and tested on subset of the DARPA 1999 dataset. This is because of the huge size of the data set which takes very long time and effort to extract payload from the packet. Accordingly the proposed system is compared with the well-known PAYL system alone. The same training and testing subsets of DARPA 1999, used in this work are applied to the PAYL system alone.

Each stage from the proposed system has its own strength that can eliminate a weakness faces the PAYL system alone:

1- Self-Organizing Map (SOM): this stage classifies the input traffic into several types. SOM classification reduces the calculation and time required to cluster the models resulted from the PAYL detector. SOM stage can perform better classification than the clustering in the PAYL system alone. Because SOM classification is based on the bytes distribution of the input traffic (not only the length of the incoming packets) and its ability to learn from that traffic. Using the same training traffic for both systems, the proposed system and the PAYL system alone, the clustering in the PAYL system alone shows less ability to recognize models compared to the SOM stage in the proposed system.

In the practical work of this dissertation the testing traffic consisting of 5081 packets. The number of packets that are appeared for the first time in the PAYL system alone are 2270 from 5081 packets (in the PAYL system alone the packet is considered appearing for first time if its payload length never appear during the training phase). There are 33 unique lengths within the 2270 packets that are appeared for the first time. By comparing these 33 unique lengths with the packets from the training phase, it is found that none of these 33 packets match any packet from the training data.

In the proposed system, the number of packets that appears for the first time are 3 from 5081 packets (in the proposed system the packet is considered appearing for the first time if it is classified by the SOM network into a neuron that doesn't have a model). In fact there is one packet in the testing data that is repeated three times and classified by the SOM network into neuron number 81. During the training phase no packet is classified into neuron number 81, which has no model. For this reason the packet in the testing phase which is classified into neuron 81 is considered appearing for the first time. By comparing these 3 packets with the packets from the training phase no match has been found, and this proves the accuracy and classification efficiency of the SOM stage in the proposed system.

From the above discussion, we conclude that the SOM stage decreases the ratio of false positive because of its efficiency in classification (the classification operation is an important stage in any network intrusion detection system). The clustering in the PAYL system alone, which is based on the length of the packet payload, is less accurate than the SOM stage in the proposed system. Where, the PAYL system alone is giving 2270 packets appearing for the first time from the testing traffic of 5081 packets. The false positive ratio of the PAYL system alone is (2270/5081)*100 equal 44.676%. While the SOM stage in the proposed system shows more accuracy in classification. The SOM stage gives

only 3 packets appearing for the first time from the testing traffic of 5081 packets. The false positive ratio of the proposed system is (3/5081)*100 equal 0.5988%.

Even if we compare the unique packets appearing for the first time for both systems as a measurement for the ratio of the false positive, the SOM stage in the proposed system gives a better performance (the number of unique packets is 1 from 3 first time appearing packets) than the clustering in the PAYL system alone (the number of unique packets is 33 from 2270 first time appearing packets).

2- Randomization: this stage is used to prevent smart worms from blinding the PAYL system alone and evade the protected systems. However the randomization stage will increase the ratio of false positive, but the existence of the SOM stage decreases the ratio of false positive; the resulting ratio of false positive for the proposed system become reasonable and acceptable.

3- Finally the PAYL detector stage of the proposed work is using the 1-gram and the same mathematical model of the PAYL system alone. The main difference is that the proposed system is working on already classified and randomized input data and there is no need to cluster anything after the PAYL detector computations.

For the training traffic extracted from weeks one and three from DARPA 1999 dataset, the number of training packets is 2148. The PAYL system alone required 1minutes to be trained, while the proposed system required 29minutes to be trained

For the testing traffic that are extracted from weeks two and four from DARPA 1999 data set, the number of testing packets are 5081. PAYL system alone required 3 minutes to end its testing phase, while the proposed system required 48 minutes to end its testing phase.

For the PAYL system alone, the total number of testing packets that generate an alert is 2452 from 5081 packets. 2270 out of 2452 packets are appeared for the first time and 182 out of 2452 packets are false positives and attacks. For the proposed system the total number of testing packets that generate alerts is 263 from 5081 packets. 3 out of 263 packets are appeared for the first time and 260 out of 263 packets are false positives and attacks.

Table 1 shows performance comparison between the proposed system and PAYL system alone.

Table 1: The performance comparison between the PAYL system alone and the proposed system.

	Number of models that appear for the first time	Total number of packets that generate alerts	False positive and the attacks ratio	Training time in minutes	Testing time in minutes
PAYL System alone	**2270**	**2452**	**44.676%**	**1**	**3**
The Proposed System	**3**	**263**	**5.176%**	**29**	**48**

V. Conclusions

During the study and development of the proposed system, several points were observed and noticed.

- There are two phases for each anomaly based IDS, training phase and testing phase, to make the data classified correctly during the testing phase the same stages must be used for both phases. The data in the proposed system is passing through three stages, first the SOM network stage; second the Randomized testing stage and finally the PAYL detector stage.
- Because the traffic payload of the same length can be classified into several categories, and the traffic is belong to a relatively small number of services and protocols. A good learning algorithm can classify the traffic into small number of clusters; a neural network the self-organizing map SOM network has been used. So instead of using the clustering technique that follows the work of the PAYL detector, SOM network is used by the proposed system to classify the input traffic (before it enters to any stage) into a certain number of types. SOM learn both the distribution (as competitive layers do) and the topology of the input traffic that is trained on. So for the proposed system the SOM network has been used to classify any input data to a specific

neuron. To accomplish the classification, the SOM network goes through three phases: initialization, training and classification.

- The PAYL system alone is considered a successful system to detect the abnormalities in the incoming traffic, of course after it has a proper training phase. In recent time, attacks have been dedicated to invade the PAYL system alone, such as the mimicry attacks. To avoid such attacks, on the PAYL system alone, we used the randomized testing technique that randomly partition packet payload into several subsets by using a randomly-generated bitmask. This technique provides diversity on the packet payload that confuses the attacker about which byte positions to pad his exploit code in order to appear normal.
- The proposed system is applied on subset of DARPA 1999 dataset for the training and testing phases, the same subset has been applied to the PAYL system alone. The proposed system is showing a better performance than the PAYL system alone. Where the clustering operation in the PAYL system alone is replaced by using the SOM network, the mimicry attacks are overcame by using the randomization method, and the same PAYL detector is used for the proposed system.
- The PAYL system alone require less time in training phase and testing phase than the proposed system. The time consumption in the proposed system, which is represent a drawback in it, is attributed to the existence of the SOM stage and the randomization stage that increase the amount of calculations required to train and test the proposed system.

IV. References

Ching-Hao Mao, En-Si Liu, Kuo-Ping Wu and Hahn-Ming Lee: "Web Mimicry Attack Detection Using HTTP Token Causal Correlation", International Journal of Innovative Computing Information and Control (IJICIC), Volume 7, 2010.

Damiano Bolzoni: "Revisiting Anomaly-based Network Intrusion Detection Systems", Ph.D. Thesis, University of Twente, 2009.

Damiano Bolzoni, Sandro Etalle, Pieter Hartel and Emmanuele Zambon: "POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System", Fourth IEEE International Workshop, 10 pp. - 156, 2006.

Ke Wang, Salvatore J. Stolfo: "Anomalous Payload-based Network Intrusion Detection", Recent advances in intrusion detection: 7th international symposium, Sophia-Antipolis, France, September 15-17, volume 3224, pp. 203-222, 2004.

KeWang: " Network Payload-based Anomaly Detection and
Content-based Alert Correlation", Ph.D. Thesis, School of Arts and Sciences, Columbia University, 2006.

"Self-Organizing Feature Map toolbox", Mat Lab Help.

Stefano Zanero: "Analyzing TCP Traffic Patterns Using Self Organizing Maps", 13th International Conference on Image Analysis and Processing (ICIAP), Cagliari (Italy), September 6-8, volume 3617 of LNCS, pp. 83-90, 2005.

P. Garcı´a-Teodoro, J. Dı´az-Verdejo, G. Macia´-Ferna´ndez and E. Va´zquez: "Anomaly-based network intrusion detection: Techniques, systems and challenges", Computers Security, Volume 28, pp. 18-28, 2009.

William Stallings: "Cryptography and Network Security Principles and Practices", Prentice Hall, 2005.