

# 3D graph Visualization prototype system for Intrusion Detection: A surveillance aid to security analysts

I. Xydas<sup>1</sup>, G. Miaoulis<sup>2</sup>, P.-F. Bonnefoi<sup>3</sup>, D. Plemenos<sup>3</sup>, D. Ghazanfarpour<sup>3</sup>

Technological Educational Institute of Athens, Ag.Spiridona St.,12210 Athens Greece

<sup>1</sup>Network Operation Center ([yxydas@teiath.gr](mailto:yxydas@teiath.gr))

<sup>2</sup>Department of Computer Science ([gmioul@teiath.gr](mailto:gmioul@teiath.gr))

<sup>3</sup>Université de Limoges, Laboratoire XLIM, CNRS, UMR 6172

83, rue d'Isle, 87000 Limoges France

[bonnefoi@unilim.fr](mailto:bonnefoi@unilim.fr), [plemenos@unilim.fr](mailto:plemenos@unilim.fr), [djamchid.ghazanfarpour@unilim.fr](mailto:djamchid.ghazanfarpour@unilim.fr)

## ABSTRACT

Intrusion Detection (ID) and Intrusion Prevention (IP) systems are an indispensable part of the information security infrastructure of every networking company or organization. Intrusion Detection Systems (IDS) have problems, such as false positives, operational issues in high-speed environments and the difficulty of detecting unknown threats. Intrusion Prevention Systems (IPS) are still in their infancy. There is a misconception in the market that intrusion detection and intrusion prevention are basically the same technology or that IDS systems are on the way out and IPS and firewalls are the wave of the future. In fact, IDS are far from becoming obsolete and both systems complement each other. Much of ID research has focused on improving the accuracy and operation of IDSs but surprisingly there has been very little research into supporting the security analysts' intrusion detection tasks. In this paper we will describe an ongoing surveillance prototype system which offers a visual aid to the security analyst by monitoring and exploring 3D graphs. The system offers a visual surveillance of the network activity on a web server for both normal and anomalous or malicious activity. Colors are used on the 3D graphics to indicate different categories of web attacks and the analyst has the ability to navigate into the web requests, of either normal or malicious traffic.

## Keywords

Visualization, information security, intrusion detection, web attacks, expert systems, neural networks, anomaly detection, network security, surveillance aid.

## 1. INTRODUCTION

Principally, there are two major categories of intrusion detection: the *Rule-based detection*, also referred to as *signature detection* and *misuse detection* and the *Anomaly detection*, also referred to as *profile-based detection* [2]. In the first approach pattern matching is used to detect known attack patterns and in the second

profiles with 'normal' behavior are created and everything that deviates sufficiently from the normal causes an alert. IDS systems based on the former category detect attacks accurately, only for the known signatures and are ineffective against previously unseen attacks. On the other hand, IDS systems based on profiles are capable of detecting novel attacks but their effectiveness is affected greatly by

what “features” of the system behavior have been learnt. They are also characterized by a high rate of false alarms and the task of selecting an appropriate set of features has proved to be a hard problem. There are also various hybrid approaches whose effectiveness is out of the scope of this paper, but most of the commercial IDS systems are ruled based.

The work of an ID analyst is a complex task that requires experience and knowledge. Analysts must continually monitor IDSs for malicious activity. The number of alerts generated by most IDS can quickly become overwhelming and thus the analyst is overloaded with information which is difficult to monitor and analyze. Attacks are likely to generate multiple related alerts. Current IDS do not make it easy for operators to logically group related alerts. This forces the analyst to look only at aggregated summaries of alerts or to reduce the IDS signature set in order to reduce the number of alerts. There are more than 3000 rules in Snort, an open source IDS available to the general public ([26],[4]). By reducing the signature set the analyst knows that although it reduces the false alarms it is also likely to increase the number of false negatives, meaning that he will not be able to detect actual attacks.

According to a recent survey, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI) [17], in the intrusion detection area visualization tools are needed to offload the monitoring tasks, so that anomalies can be easily flagged for analysis and immediate response by the security analyst. Information presented in a visual format is learned and remembered better than information presented textually or verbally. The human brain is structured so that visual processing occurs rapidly and in parallel. Given a complicated visual scene humans can immediately pick important features in a matter of milliseconds. Humans are limited in terms of attention and memory but they excel at

the processing of visual information. The strength of Visualization to amplify cognition and the lack of visualization tools for ID analysts have led us to design such a visual tool and create a prototype system which is described in this paper. It is a surveillance aid for the security analyst, offering him a user friendly visual tool to detect anomalies in web requests by exploring 3D graphs, to understand quickly the kind of underlying attack by means of colors and the possibility to navigate into the payload of the web request for further analysis and adequate response. The system looks into web requests to detect “fingerprints” which are special characters or chains of characters. These fingerprints are then passed to an expert system to decide if they constitute a malicious request or attack. The output of the expert system is then transformed to a 3D graph for visual interpretation and in parallel is kept for statistical analysis. The role of the expert system is to eliminate the false alarms by consulting its knowledge data base; a task which is absent in rule-based IDS systems. Web attacks can be either rejected by the server or can be successful due to security weaknesses. If penetration occurs action must be taken by the security analyst as the prototype system does not deal with resolving the damage caused by an attack. It is solely a surveillance device.

The rest of this paper is organized as follows: section 2 presents related work, section 3 describes the visualization prototype system section 4 describes our experimental data, section 5 describes the system performance evaluation. Finally, concluding remarks and ideas for future work appear in section 6.

## **2. RELATED WORK**

There is an ongoing research on IDS systems especially on anomaly detection and profile or specification-based detection. This includes various statistical methods,

artificial neural networks and data mining methods ([10],[11],[12]). Interesting recent work on the detection of web-based attacks are presented in [13] (a new taxonomy of Web attacks suitable for efficient encoding), [14] (Anomaly detection of web-based attacks) and [15] (a multi-model approach to the detection of web-based attacks). However, there has been little research into understanding the human component of ID work. Recent surveys in security area have shown the need for novel monitoring tools and vulnerability scanners for the ongoing surveillance of systems and network activity [16]. The framework of an Information Visualization (IV) for Intrusion Detection is presented in [17] as well as a three-phase process model (monitoring-analysis/diagnosis-response) that frames corresponding requirements for IV tools, based on a survey of ID security analysts' needs.

Visualization has been used in various areas such as the VISUAL system in [18], which is a home-centric Visualization tool of Network Traffic for security administration. Visualization has been also used in [19] for a passive visual fingerprinting of network attack tools, such as nmap, superscan, nessus, nikto and others. Visual tools have also been used to visualize logs of IDS systems, such as the SnortView in [20], a 2D visualization system of Snort logs and a Web-based system for Intrusion Detection in [21]. The latter captures the network traffic from the Snort IDS and using a data mining system displays with a web browser the traffic, filtered by source/destination host, protocol or alert, using bar graphs or pie-charts. Interesting work has been presented by Axelsson in [22] where 3D visualization has been used to detect web malicious traffic. He processed the logs of a web server and used a log reduction system based on frequencies in order to select the traffic for the visualization of the web requests and the detection of unauthorized traffic. 3D visualization is done on preselected traffic, including both normal

and malicious traffic and the operator should navigate into the subgraphs and the graph tails in order to detect malicious or suspect traffic. In recent work [23] Axelsson presented an IDS system based on a bayesian classifier combined with a 2D visual tool called Bayesvis. Finally, it is worth referring to the research on the Information Visualization systems protection such as presented in [24], (Attacking Information Visualization system usability; overloading and deceiving the human, 2005).

Our work focused on creating an ongoing surveillance tool offering the security analyst a novel visual tool for monitoring and diagnostic needs. We would like to offer an online tool which is capable of dealing with real network traffic in addition to processing stored web logs. Visualization has been designed in such a way that the operator is not overwhelmed with uninteresting normal traffic. The security analyst immediately detects the malicious traffic by spotting the colored information on the screen and he has the option of displaying only the unauthorized colored traffic, by removing the normal black traffic. This allows him to navigate quickly into the web request data for a precise diagnosis and quick response. Malicious traffic has been detected by the artificial neural network's knowledge base and a self-organizing network has been used for the web attacks classification. Our approach differs to that of Axelsson's in that we are dealing with real time data in addition to web logs processing, we are using coloring in the 3D visualization for quick interpretation and diagnosis and furthermore, we have implemented two artificial neural networks, both an unsupervised and a supervised one, for the web class and attack classification. In addition we were expanding the signature method for ID to detect backdoor intrusions and code execution attempts by high level applications such as HTML, Java, SQL, Perl, Php and Access db. Finally, we must

emphasize that the whole system is developed in Linux and all system modules are written in standard C language, offering speed and portability to any operating system and platform, even on small portable computers.

### 3. VISUALIZATION PROTOTYPE SYSTEM

Our visualization prototype system consists of the following modules: The data capture module, the preprocessor module, the neural knowledge base module, the statistical analysis module and the graph generator module.

The data capture module selects data either online from the Internet traffic or offline from the web server logs. The preprocessor module examines the web requests to detect malicious traffic and its output is then forwarded to the neural knowledge base module to predict the type of unauthorized traffic. Then, both normal and malicious traffic are processed by the graph generator module for visualization. Additionally, all traffic is kept for statistical analysis. Figure1. shows the architecture of the visualization prototype system. Each module is described in detail below.

In 2004, an increase of 36% of web attacks was registered for a total of about 400.000 web servers worldwide. The two most popular web servers are Microsoft Internet Information Services (IIS) and the open source Apache web server. The IIS web server of the Library of the Technological Educational Institute of Athens was used in order to study the various types of attacks and to create the knowledge data base of the system.

Modern web servers offer optional features which improve convenience and functionality at the cost of increased security tasks. These optional features are taken in consideration in our design in addition to traditional types of web attacks

(Unicode, directory traversal, buffer overflow, Server-Side Includes-SSI, Cross Site Scripting-CSS, mail and CGI attacks). Different kinds of application insertion attempts are detected such as HTML, Javascript, SQL, Perl, Access and Php. In addition IIS indexing vulnerabilities, IIS highlight, illegal postfixes, IIS file insertion (.stm), IIS proxy attempts and IIS data access vulnerabilities (msadc) are detected as well. All .asa, .asp and Java requests are tested for URI (Uniform Resource Identifier) legal syntax according to standards, meaning that a corresponding query not in the form `<?key=value>` is illegal. Trojan/backdoor upload requests are detected as well. These backdoors are left by worms such as Code Red, Sadmin/IIS and Nimda. Backdoor attempts for apache and IIS servers are detected when web requests ask for the corresponding password files (.sam and .htpasswd). Finally, command execution attempts are detected for both Windows (.exe, .bat, .sys, .com., .ini, .sh, .dll and other) and Unix (cat, tftp, wget, ls and other) environments.

A total of 30 fingerprints (dimension 30) was used in the model to group all the different types of web known attacks ([8], [25]).

To classify the above web attacks types a self-organizing neural network system has been used. The system was based on the famous Grossberg and Carpenter's Adaptive Resonance Theory (ART1). ART1 algorithm is an unsupervised learning algorithm with biological motivations. Clustering algorithms are motivated by biology in that they offer the ability for learning through classification. Based on the Grossberg's *stability-plasticity dilemma* we cluster new concepts with analogous old ones and when we encounter new knowledge we create new clusters without destroying what has already been learned ([1], [5], [7]).

The input vector (feature vector) to the ART1 system was prepared by the preprocessor which detected the various fingerprints in a web request. The presence of a specific fingerprint is indicated in the feature vector as 1 (true) and its absence as 0 (false or unknown). The ART1 neural network created 15 clusters or classes. These 15 classes were finally grouped to 9 as there was more than one class for command execution (Windows, Unix) and IIS type of attacks. It is interesting to notice that ART1 did not create a separate class for Directory traversal and Unicode attacks because almost all of the web requests containing Unicode or traversal fingerprints (..\ or ..) always included another type of attack (e.g buffer overflow, command execution attempt or other). So, directory traversal and Unicode attempts are not classified as separate attack classes. For historical reasons we included Unicode attempts into the Miscellaneous class.

Therefore, the final 9 web attack classes used are the following:

- CMD (Unix or Windows command execution attempt)
- INS (code insertions of type HTML, Perl, SQL Javascript, SQL, Perl, Access)
- TBA (Trojan backdoor attempt)
- MAI (different mails such as sendmail, formail, email etc. )
- BOV (Buffer overflow)
- CGI (cgi scripts)
- IIS (IIS server attacks)
- CSS (Cross Site Scripting or Server Side Include)
- MISC (miscellaneous, coldfusion, Unicode, and malicious web request options such as PROPFIND, CONNECT, OPTIONS, SEARCH, DEBUG, PUT and TRACE ).

If the preprocessor detects even one fingerprint its output is forwarded to a neural network for classification. Neural networks (NN) represent a class of very powerful, general-purpose tools that have

been successfully applied to prediction, classification and clustering problems. The popularity of neural networks is based on their remarkable versatility and ability to handle both binary and continuous data and to produce good results in complex domains [1]. When the output is continuous the network can address prediction problems, but when the output is binary, as in our case, the network works as a classifier.

The NN used is a multilayer network with one hidden layer, using the *generalized delta rule (backpropagation algorithm)* for learning and the sigmoid function as activation function [3], [9]. For applying the gradient descent method to the training of the network we used the *continuous updating* approach, which requires that the weights will be updated after each training pattern is presented. Let the training set be  $\{\mathbf{x}(k), \mathbf{d}(k)\}$   $k=1, N$  where  $\mathbf{x}(k)$  is the input pattern vector to the network,  $\mathbf{y}(k)$  the actual output and  $\mathbf{d}(k)$  the desired output vector for the input pattern  $\mathbf{x}(k)$ . Then, the weights are updated according to the equation:

$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \eta \partial E(k)/\partial \mathbf{w}$  , where  $\mathbf{w}$  presents all the weights arranged as a vector,  $\mathbf{w}^{\text{new}}$  and  $\mathbf{w}^{\text{old}}$  may be viewed as weight vectors at time indices  $k+1$  and  $k$  respectively,

$$E(k) = \frac{1}{2} \sum_{j=1}^m [y_j(k) - d_j(k)]^2$$

is the sum of squares of the error over all output units for the  $k^{\text{th}}$  exemplar,  $m$  the number of output units and  $\eta$  the learning rate (we used  $\eta=0.2$ ). The total classification error over the set of  $N$  exemplars is defined as

$$E_T = \sum_{k=1}^N E(k).$$

To accelerate the training and increase the speed of convergence while minimizing the possibility of oscillation around local minimums a momentum term (we used

$\beta=0.95$ ) has been added to the basic gradient descent formulation. In this case, the weight vector at time index  $(k+1)$  is related to the weight vectors at time indices  $k$  and  $(k-1)$  by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - [\eta \partial E / \partial \mathbf{w} + \beta \Delta \mathbf{w}(k-1)].$$

Initially, for the prediction of the network output the “winner-takes-all” method was used, that is the output with the biggest value (rated between 0 and 1) determined the class of the web attack. Later, during the evaluation of the network performance a threshold was used instead of the “winner-takes-all” mechanism.

The predicted by the neural network type of attack is then used to create a colored directed graph in **dot** form of the well known GraphViz [28] package, using the corresponding *DOT* language. This language describes three kinds of objects: graphs, nodes and edges. A graph  $G$  is a tuple  $(V,E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges (subset of the Cartesian product  $V \times V$ ). The language has a large number of attributes that affect the graph drawing.

The payload of a web request is cut in nodes and the directed edges are the links between these nodes from left to right. Therefore, a web request from an IP source 217.229.196.17 with payload GET /hact/graphics/blackwell.jpg, has as nodes the words “217.229.196.17”, “GET”, “hact”, “graphics”, “blackwell.jpg” and as “directed edges” the links between these nodes from left to right:

217.229.196.17  $\rightarrow$  GET  $\rightarrow$  hact  $\rightarrow$  graphics  
 $\rightarrow$  blackwell.jpg.

When each web request with its IP source address and the requested data is visualized

in a 3D graph the security analyst can navigate into the graph for a quick interpretation and evaluation in case of a malicious attempt. Timestamps were not added to the graph as graphs are displayed in real time and the objective here is to keep the display as simple as possible.

There are two graphs generated with the GraphViz package. One graph contains real time traffic, e.g. both normal and possible malicious traffic and the other does not contain normal but only the possible malicious traffic. Normal traffic is visualized in black and malicious traffic in 9 different colors, one for each attack class. This visual separation was necessary because normal traffic overloads the display and the security analyst cannot interpret quickly the malicious attempts. When visualizing both normal and malicious traffic the security analyst spends more time navigating through the graph trying to eliminate normal traffic by zooming into the colored part of the display, than he would if he had only a colored graph to contend with.

These two *dot* colored graphs are then visualized with Tulip [27], a 3D graph visualization tool, supporting various graph algorithms and extensive features for interactive viewing and graph manipulation.

Figure 2 visualizes a mixed traffic and Figure 3 displays only the malicious traffic.

The cyan graph indicates a buffer overflow attempt from IP 195.249.40.234, the dark orchid graph a Perl insertion attempt from IP 62.195.136.174, the deep pink graphs show backdoor attempts with PhpMyAdmin and IIS from IP 66.194.6.80 etc.

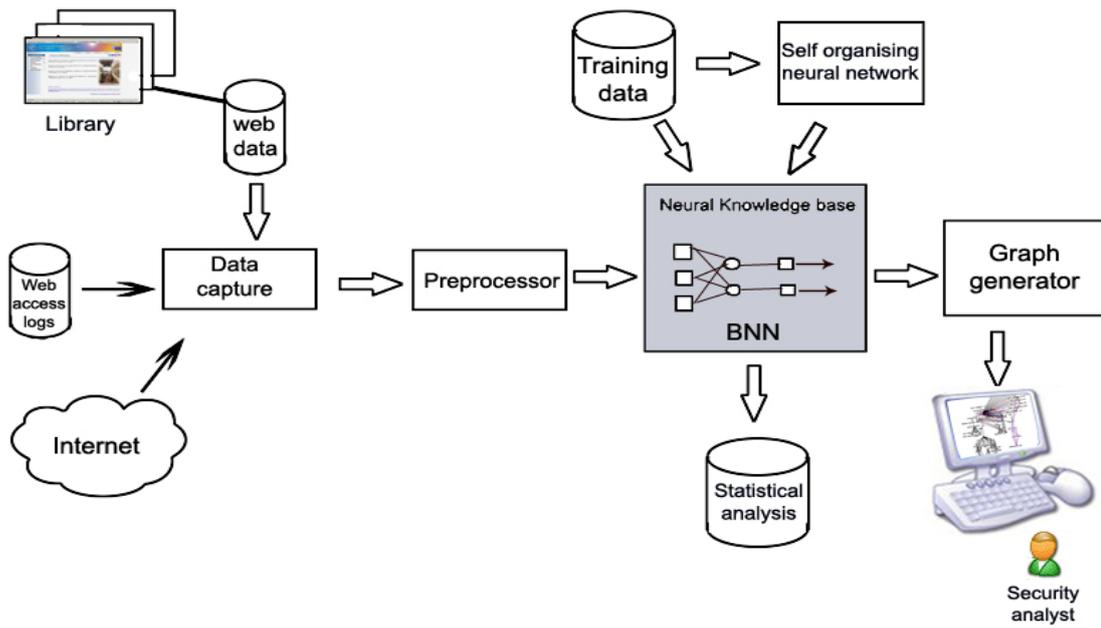


Figure 1. Visualization prototype system architecture

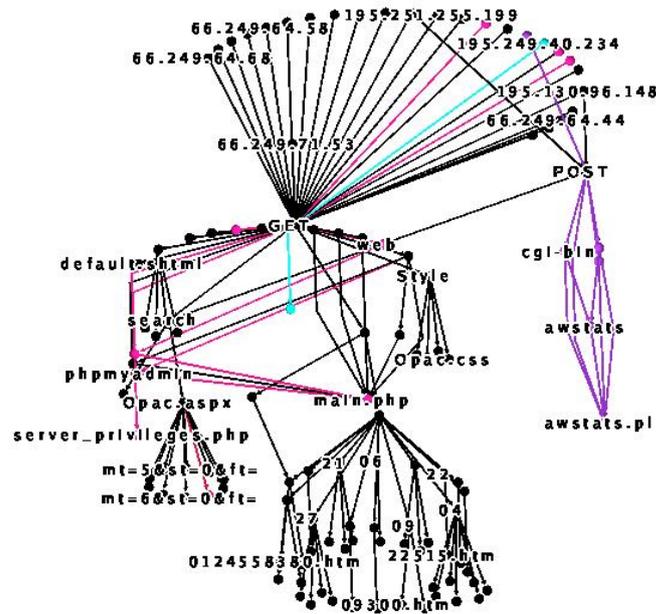


Figure 2. Visualization of normal (black) and malicious (colored) traffic

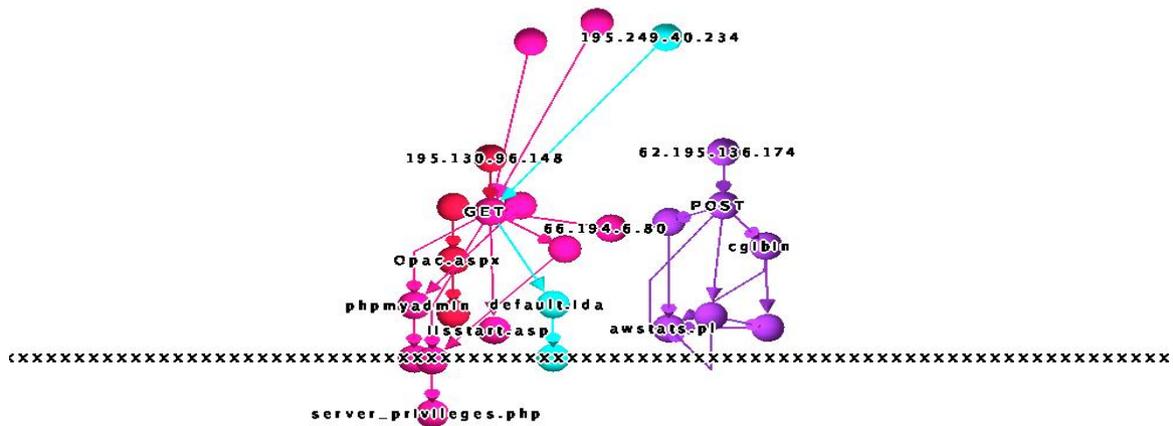


Figure 3. Visualization of malicious (colored) traffic

#### 4. TRAINING DATA QUALITY

To determine how well the neural network was able to learn from the given set of data (training set) we measured the *information* which is shared between the proposed input to the network and its target output. The entropy measure (or uncertainty) of the input layer and the output layer, independent of one another, should be as close to their maximum value as possible. Low entropy at the input or output causes a bias in the network's learning.

To measure the information which exists between the input data and the output data we had to calculate the *mutual information* between the two data sets. We want the network to take the input and remove all uncertainty about what the corresponding output should be. The amount of the original uncertainty we can remove depends on the mutual information present in the data. With an ideal training set, once we know the input value, there should be no doubt as to the correct output value: it should be the one value with a conditional probability, given the current input, of one; all other output values should have a probability of zero. As we cannot have an ideal training set, we need a measure of the

average spread of conditional probabilities over the whole training set [6].

Let  $H$  denote the entropy of a set of events,  $X$  and  $Y$  the data sets of input and output respectively,  $H(X|Y)$  the conditional entropy of inputs given the outputs and  $I(X;Y)$  the mutual information between the input and the output data of the training set. We measured the  $H(X)$ ,  $H(Y)$  and  $I(X;Y)$  using a program to calculate the equations:

$$H = \sum_{i=1}^n P_i \log (1/P_i),$$

$$H(X|Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log( P(y_j) / P(x_i, y_j) ),$$

$$I(X;Y) = H(X) - H(X|Y) \text{ or}$$

$$I(X;Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log(P(x_i, y_j) / (P(x_i)P(y_j)))$$

where  $n$  is the number of possible distinct input events,  $m$  the number of possible distinct output events and  $P_i$  is the probability of event  $i$  occurring out of the possible  $n$  events.

Table 1 shows the results with the used training set:

<b>n</b>	<b>log(n)</b>	<b>m</b>	<b>log(m)</b>	<b>H(X)</b>	<b>H(Y)</b>	<b>H(X/Y)</b>	<b>H(Y/H)</b>	<b>I(X;Y)</b>
49	3.891	9	2.197	3.512	2.160	1.777	0.420	1.74

**Table 1. Data sets entropy and mutual information results**

As we can see:  $H(\text{inputs}) \approx \log(n)$  and  $H(\text{outputs}) \approx \log(m)$ , so the used training set is a well balanced training set. The ratio  $I(\text{input}; \text{output}):H(\text{output})$  ranges from 0 to 1 and will be high if a data set is learnable. This ratio for our data set is equal to 0.80, which means that the data set used is learnable. However, it could be improved in the future.

## 5. SYSTEM PERFORMANCE EVALUATION

We tested the system performance using real data, captured with tcpdump utility in October and November 2005. Using only real data we could not have a complete set of various attacks, so we have completed the tests with web logs data of 2003 and 2004. In the statistical analysis module of the system (Fig.1) a confusion matrix is calculated to display the classification results of a network. The confusion matrix is defined by labeling the desired classification on the rows and the predicted classifications on the columns. For each exemplar, a 1 is added to the cell entry defined by (desired classification, predicted classification). Since we want the predicted

classification to be the same as the desired classification, the ideal situation is to have all the exemplars end up on the diagonal cells of the matrix. Tables 2a and Table 2b show such a confusion matrix for test1 (web logs 2003), with thresholds of 0.5 and 0.8 respectively. In addition, for each test a 2x2 table is calculated containing, on the first row the Hits (attacks present or True Positives) and the False Alarms (or False Positives) and on the second row the Misses (attacks present but not detected or False Negatives) and the Correct Rejections (normal traffic or True Negatives). Results are presented in Table 3 in this form. All tests have been run for various values of a detection threshold to show how changing the detection threshold affects detections versus false alarms. If the threshold is set too high then the system will miss too many detections and conversely, if the threshold is set too low there will be too many false alarms. For the tests we have used threshold values rating from 0.3 to 1.0 with a step of 0.1. The best results were obtained with a threshold value of 0.8 giving maximum detections of 98% and a minimum of false alarms. Table 3 summarizes the results with various testing data sets.

	<b>CMD</b>	<b>INS</b>	<b>TBA</b>	<b>MAI</b>	<b>BOV</b>	<b>CGI</b>	<b>IIS</b>	<b>CSS</b>	<b>MIS</b>	<b>NRM</b>
<b>CMD</b>	21593	16	0	0	0	0	0	0	0	0
<b>INS</b>	0	2	0	0	0	0	0	0	0	0
<b>TBA</b>	0	1	511	0	0	0	0	0	0	0
<b>MAI</b>	0	0	0	0	0	0	0	0	0	0
<b>BOV</b>	0	0	0	0	0	0	0	0	0	0
<b>CGI</b>	0	0	0	0	0	1	0	0	0	0
<b>IIS</b>	0	0	0	0	0	0	197	0	0	0
<b>CSS</b>	0	0	0	0	0	0	0	0	0	0
<b>MIS</b>	0	0	0	0	0	0	0	0	40	0
<b>NRM</b>	0	0	0	0	0	0	0	0	0	127182

**Hits: 22344**      **False Alarms: 17**  
**Missed: 141**      **Normal traffic: 127182**      **Total events: 149684**

**Table 2a. Confusion matrix for test1 (threshold 0.5)**

	<b>CMD</b>	<b>INS</b>	<b>TBA</b>	<b>MAI</b>	<b>BOV</b>	<b>CGI</b>	<b>IIS</b>	<b>CSS</b>	<b>MIS</b>	<b>NRM</b>
<b>CMD</b>	<b>21584</b>	<b>16</b>	0	0	0	0	0	0	0	0
<b>INS</b>	0	<b>2</b>	0	0	0	0	0	0	0	0
<b>TBA</b>	0	0	<b>508</b>	0	0	0	0	0	0	0
<b>MAI</b>	0	0	0	0	0	0	0	0	0	0
<b>BOV</b>	0	0	0	0	<b>140</b>	0	0	0	0	0
<b>CGI</b>	0	0	0	0	0	<b>1</b>	0	0	0	0
<b>IIS</b>	0	0	0	0	0	0	<b>197</b>	0	0	0
<b>CSS</b>	0	0	0	0	0	0	0	0	0	0
<b>MIS</b>	0	0	0	0	0	0	0	0	<b>40</b>	0
<b>NRM</b>	0	0	0	0	0	0	0	0	0	<b>127182</b>

**Hits: 22472      False Alarms: 16**  
**Missed: 14      Normal traffic: 127182      Total events: 149684**

**Table 2b. Confusion matrix for test1 (threshold 0.8)**

<b>Threshold</b>	<b>0.3</b>		<b>0.4</b>		<b>0.5</b>		<b>0.6</b>		<b>0.7</b>		<b>0.8</b>		<b>0.9</b>		<b>1.0</b>	
<b>Positives</b>	TP	FP														
<b>Negatives</b>	FN	TN														
<b>Logs 2003</b> <b>149684 events</b>	22304	28	22332	26	22344	17	22344	16	22442	25	22472	16	22334	0	0	0
	170	127182	144	127182	141	127182	142	127182	35	127182	14	127182	49	127182	22502	127182
<b>Logs 2003</b> <b>149545 events</b>	18079	241	18324	13	18323	5	18329	0	18501	13	18703	0	18688	0	0	0
	445	130780	428	130780	437	130780	436	130780	251	130780	62	130780	77	130780	18765	130780
<b>Logs 2003</b> <b>149656 events</b>	9555	12	9550	26	9551	0	9551	2	9559	0	9568	0	9532	0	0	0
	163	139726	154	139726	179	139726	177	139726	171	139726	162	139726	198	139726	9730	139726
<b>Logs 2003</b> <b>78688 events</b>	6579	29	6580	10	6583	27	6583	7	6637	0	6644	7	6639	0	0	0
	82	72002	96	72002	76	72002	96	72002	49	72002	35	72002	47	72002	6666	72002
<b>Logs 2004</b> <b>149450 events</b>	7517	1	7572	6	7577	0	7577	0	7575	0	7579	0	7553	0	0	0
	115	141817	55	141817	56	141817	56	141817	58	141817	54	141817	54	141817	7633	141817
<b>Logs 2004</b> <b>149503 events</b>	10128	8	10164	0	10176	8	10176	0	10183	0	10191	0	10168	0	0	0
	102	139265	74	139265	54	139265	62	139265	55	139265	47	139265	70	139265	10238	139265
<b>Logs 2004</b> <b>149749 events</b>	3536	6	3536	0	3614	0	3614	0	3631	0	3631	0	3605	0	0	0
	152	146055	158	146055	80	146055	80	146055	63	146055	63	146055	63	146055	3694	146055
<b>tcpdump Oct 05</b> <b>49372 events</b>	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9	0
	0	49363	0	49363	0	49363	0	49363	0	49363	0	49363	0	49363	0	49363
<b>tcpdump Nov 05</b> <b>22022 events</b>	34	22	34	0	34	0	34	0	56	0	56	0	34	0	0	0
	13	21953	35	21953	35	21953	35	21953	13	21953	13	21953	35	21953	69	21953

**Table 3. Performance evaluation tests**

## 6. CONCLUSIONS AND FUTURE WORK

Visualization offers a powerful means of analysis that can help the security analyst uncover hacker trends or strategies that are likely to be missed with other nonvisual methods. Visualization allows him to audit the analytical process, since the operator is examining the web traffic directly and online and is making iterative decisions about what is being presented. With our work we have contributed to network security and web visualization the following:

- A new visualization of web traffic that enables rapid perception and detection of unauthorized traffic
- On going visualization of network traffic
- Capability to isolate malicious traffic for immediate analysis and response
- Use of neural knowledge base for rapid classification of attacks
- A surveillance aid for the security analyst
- A visualization prototype system ideal for educational purposes and untrained users to understand web server security state.

Network data analysis is a very important but time consuming task for any administrator. A significant amount of time is devoted to sifting through text-only log files and messages generated by networks tools in order to secure networks. This project has demonstrated that visualization considerably reduces the time required for data analysis and at the same time provides insights which might otherwise be missed during textual analysis.

The web traffic surveillance could be expanded to other basic but popular internet services, such as mailing or DNS. Combining traditional or novel analytical

methods with visual presentation techniques can generate a very robust approach to network security. Visualization and artificial intelligence can be incorporated in ID systems to produce more powerful security systems capable of dealing with the new attack challenges.

The neural networks used in our work could be combined with ruled based systems or fuzzy logic systems to produce hybrid systems with approximate reasoning such as neural expert systems or neuro-fuzzy systems capable to deal with noisy and incomplete data. This is undoubtedly the future in the ID area.

## ACKNOWLEDGEMENTS

I would like to thank Mr. Fabien Guion for his help on the preprocessor programming and Mr. G. Patestos the library network administrator for providing web logs test data and relevant information.

## 7. REFERENCES

- [1] M.Tim Jones. AI Application Programming, p.35-113, Charles River Media, 2003.
- [2] C.Endorf, E. Schultz and J. Mellander. Intrusion Detection & Prevention, p.1-247, McGraw-Hill/Osborne, 2004.
- [3] M.Negnevitsky. Artificial Intelligence, A Guide to Intelligent Systems, p.25-297, Pearson, Addison Wesley, 2002.
- [4] R.U.Rehman. Intrusion Detection with Snort, Advanced IDS Techniques Using Snort, Apache, MySQL, PHP and ACID, p.1-207, Prentice Hall PTR, 2003.
- [5] K.Gurney. An Introduction to Neural Networks, p.7-166, UCL Press, London, 1997.

- [6] K.Swinger. Applying Neural Networks, A Practical Guide, p.105-163, Academic Press Ltd., 1996.
- [7] I.H.Witten,E.Frank. Data Mining, p.170-227, Morgan Kaufmann Publishers, 2000.
- [8] J.Chirillo. Hack Attacks Revealed, p.485-544, Wiley Publishing, 2002.
- [9] S.Haykin. Neural Networks, A Comprehensive Foundation, 2<sup>nd</sup> edition, p.156-208, Prentice Hall PTR, 1999.
- [10] R.Sekar, A.Gupta, J.Frullo, T.Shanbhag, A.Tiwari, H.Yang and S.Zhou. Specification-based Anomaly Detection: A new Approach for Detecting Network Intrusions, *Proceedings of the 9<sup>th</sup> ACM conference on computer and communications security*, p.265-274, ACM Press, Nov. 2002.
- [11] W-H.Chen, S-H.Hsu, H-P.Shen. Application of SVM and ANN for intrusion detection, *Computers and Operations Research*, Vol.32 Issue 10, Elsevier, Oct. 2005.
- [12] W.Lee, S.Stolfo, K. Mok. Adaptive Intrusion Detection: A Data Mining Approach, *Artificial Intelligence Review*, Vol.14 Issue 6, p.533-567, Kluwer Academic Publishers, Dec. 2000.
- [13] G.Alvarez, S.Petrovic. A new taxonomy of Web attacks suitable for efficient encoding, *Computers & Security*, Vol. 22 Issue 5, p.435-449, Elsevier, Jul. 2003.
- [14] C.Kruegel, G.Vigna. Anomaly Detection of Web-based Attacks, *Proceedings of the 10<sup>th</sup> ACM conference on Computer and communications security*, p.251-261, ACM Press, Oct. 2003.
- [15] C.Kruegel, G.Vigna, W.Robertson. A multi-model approach to the detection of web-based attacks, *Computer Networks*, Vol. 48 Issue 5, p.717-738, Elsevier, Aug. 2005.
- [16] J.R.Goodall, W.G.Lutters and A.Komlodi. The Work of Intrusion Detection: Rethinking the Role of Security Analysts. *Proceedings of the Tenth Americas Conference on Information Systems*, p.1421-1427, New York, Aug. 2004.
- [17] A.Komlodi, J.R.Goodall and W.G.Lutters. An Information Visualization Framework for Intrusion Detection. *CHI '04 extended abstracts on Human factors in computing systems*, p.1743-1746 , ACM press, Apr. 2004.
- [18] R.Ball, G.A.Fink, C.North. Home-Centric Visualization of Network Traffic for Security Administration, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, p.55-64, ACM Press, Oct. 2004.
- [19] G.Conti, K.Abdullah. Passive Visual Fingerprinting of Network Attack Tools, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, p.45-54, ACM Press, Oct. 2004.
- [20] H.Koike, K.Ohno. SnortView: Visualization System of Snort Logs, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, p.143-147, ACM Press, Oct. 2004.
- [21] A.Nalluri and D.C.Kar. A web-based system for Intrusion Detection, *Journal of Computing Sciences in Colleges*, Vol. 20 Issue 4, p.274-281, Consortium for Computing Sciences in Colleges (CCSC), USA, Apr. 2005.

- [22] S.Axelsson. Visualising Intrusions: Watching the Webserver. *Security and Protection in Information Processing Systems, IFIP 18<sup>th</sup> World Computer Congress, TC11 19<sup>th</sup> International Information Security Conference (SEC 2004)*, p.259-274, Toulouse, France, Kluwer, Aug. 2004.
- [23] S.Axelsson. Combining a Bayesian Classifier with Visualisation: Understanding the IDS, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, p.99-108, ACM Press, Oct. 2004.
- [24] G.Conti, M.Ahamad, J.Stasko. Attacking Information Visualization System Usability, Overloading and Deceiving the Human, *Proceedings of the 2005 Symposium on Usable Privacy and Security (SOUPS '05)*, p.89-100, Pittsburgh PA, USA, ACM Press, July 2005.
- [25] Fingerprinting Port 80 Attacks, A look into web server and web application attack signatures, [admin@cgisecurity.com](mailto:admin@cgisecurity.com), 2002.
- [26] Snort software, <http://www.snort.org>
- [27] Tulip software, <http://www.tulip-software.org>
- [28] Graph Visualization software, <http://www.graphviz.org>