Visualisation of Network Traffic using Dynamic Co-occurrence Matrices

Thorsten Kisner, Alex Essoh and Firoz Kaderali

Department of Communication Systems
Faculty of Mathematics and Computer Science
FernUniversität in Hagen, Germany
{thorsten.kisner, alex.essoh, firoz.kaderali}@fernuni-hagen.de
February 19, 2007

***Abstract*—Traffic visualisation is important in several areas e.g. network planning and monitoring, network traffic analysis and intrusion detection. A novelty in the work we present in this paper is the use of texture analysis methods from the domain of digital image processing for network traffic visualisation. We use strategies based on co-occurrence matrices to derive statistical properties for network traffic visualisation and anomalous traffic detection. Based on the fact that some of the statistical properties are related to a certain kind of traffic, which is also reflected in the allocation of the dynamic co-occurrence matrix, we are able to display the global status of our network and show periods, where the traffic behaviour is unusual. Further, we introduce a new parameter, *Network Traffic Homogeneity* (*NTH*) as a measure of the local roughness of the network traffic.**

I. Introduction

The topic Intrusion Detection has been of great interest over the past years. Not surprisingly when the dramatic increase of incidents and vulnerabilities over the past ten years is taken into consideration. The growing number of incidents and vulnerabilities resulting from the complexity of communication protocols and related applications together with the fact that many tools are currently available to randomly scan the internet for security holes, cause a lot of problems to security officers responsible for the network administration. Therefore many security officers use intrusion prevention, detection and response systems in order to anticipate, detect and react to attacks by outsiders or misuse as well as abuse by insiders.

With regard to detecting intrusions on computer systems and networks many approaches have been proposed in the past, ranging from rule based attacks detection e.g. using *Snort*, neural networks [1], data mining [2] or decision trees to support vector machines [3]. The process of detecting intrusions generally consists of analysing log files and displaying relevant information resulting from the analysis to the security officer. Studies carried out in the past confirmed that text based tools were widely used to estimate the network status. In [4] most of the interviewed system administrators answered that they only use text based tools but most of them realise the necessity of improved visualisation methods, including interactivity. It is nearly impossible for a human to gather all relevant information to directly understand the security status of the network, especially for heavily loaded networks with thousands of packets per minute.

Although humans cannot understand huge amounts of text data at once, they have magnificent capacities in image processing, thus a suitable visualization of this data is needed and has been the focus of a lot of research in the recent past. Hereby different approaches (geometric approaches [5], [6], [7], [8], [9], [10], icons and glyphs based methods [4] [11], pixels based methods [12], [13], hierarchy and graph based [14] or hybrid approaches [15]) have been used to effectively visualise the traffic. Since, in most cases, multidimensional data is being dealt with, a lot of effort has also been put into reducing the dimension of the multivariate data in order to be able to render the data on a display. Our approach is different from many other visualisation methods. We use the same basic data i.e. network traffic. We represent network traffic as a time series and transfer it to the domain of digital image processing by mapping traffic samples to an image. With this representation we can make use of statistical texture analysis methods to analyse and visualize the resulting data set. Further, we can process detailed network packet data, e.g. TCP flows, or can use aggregated data where only the packet arrival times and packet sizes are available.

In this paper we introduce a method to visualise network traffic using statistical texture analysis. To do this the co-occurrence matrix [16] [17] and related statistical metrics are plotted versus time. Since some of the statistical traffic parameters differ significantly according to the type of traffic, we are able to visualise network traffic and display scan periods. There has not been an application of statistical texture analysis methods, specifically co-occurrence matrices and related parameters to network traffic visualisation to date. Indeed, an application of co-occurrence matrices to computer security can be found in [18], where Mizuki et al. propose a method for the detection of masqueraders using the so called *Eigen Co-occurrence Matrices*, but the detection method works on Unix commands and not on network packets. Furthermore the visualisation of the traffic is not an issue in their paper.

The rest of this paper is organised as follows: in section II we give an overview of work carried out in the area of

traffic visualisation. Section III introduces texture analysis methods with co-occurrence matrices. In section IV we present the results of our proposal and we summarise and provide direction for future work in section V.

II. Related Work

Several tools have been proposed in the past to visualise network traffic. Characteristic for all of them is the fact that header related information, mostly IP address, port number or time are used to create a grid for the representation of the network status. Each position within the grid then represents a specific network activity.

Lakkaraju et al. [7] use *NVisionIP* to analyse and monitor the whole IP address space of a class B network in three different views. The whole network can be monitored by a two dimensional representation of subnets and related hosts with hosts being represented in the y-axis and subnets in the x-axis in the global view. Each host is coloured depending on its traffic characteristics. Network regions presenting suspicious behaviour can be considered further by selecting a rectangle in the global view. This opens the Small Multiple View (SMV), where traffic on open ports is colour encoded and displayed using port related two bar charts[1]. A further visualisation is offered by the machine view, where diverse traffic parameters such as byte and flow counts can be analysed further. Ball et al. [4] also use the IP address to display the state of a network on the screen. Hereby the first two bytes of an IP address are used as x coordinate and the last two as y coordinate. In order to avert overlapping, adaptation techniques are used [19]. The home network is visualised as a large square grid, wherein each single host is shown as a small square. External hosts are visualised by markers, whose size depends on the intensity of communication. The communication is represented by lines and the related direction is colour encoded. Goodall et al. [10] find it important to provide analysts with the so called ”big picture” of the network state by putting individual packet details into a larger context. This is achieved through the use of the *Time-based Network traffic Visualizer* (TNV). TNV provides different levels of aggregation. At the highest level the entire network can be shown. A further level is the so called matrix display, here hosts IP addresses are displayed in the y-axis and the time in the x-axis. Time intervals are represented by columns, hosts by rows, and the number of packets for a specific time interval is colour encoded and displayed in the resulting box. The network administrator or security officer is able to localise areas of higher activity through the colour of the box. The colour of the link indicates the protocol and port activity can also be displayed. Hideshima et al. [9] argue that it is very important to know where intrusions come from and introduce the geographical dimension to the logical and temporal ones. The logical visualisation is e.g. a 2-D representation of the relationship between IP addresses. In the temporal visualisation, an event of interest e.g. the number of packets or the number of attacks is visualised versus time. This visualisation provides better understanding of the time transition of attacks. The IP address block and geographical location are connected with a line. Statistical information e.g. number of attacks for each location are shown as bar charts. Arcs on the map indicate source and destination addresses. Itoh et al [14] also use the IP address as the starting point of their approach. They propose an algorithm for the hierarchical representation of network data. They formed four levels of hierarchical data using the IP address space and then visualise it using black square icons and rectangles. Hereby branch nodes are displayed as rectangular borders whereas leaf nodes are represented as squares inside the rectangular border. When a user clicks on a leaf node, the corresponding hosts are displayed together with the corresponding number of incidents.

The work of Girardin et al. [20] considerably differs from those we have seen so far. Girardin et al. use a neural network (self organizing maps, introduced by Kohonen [21]) to perform the mapping with regard to the traffic to be visualised. Hereby a multidimensional vector consisting of a number of attributes such as packet length, host and port is built and used as input for a neural network. Based on similarities between the data points, using the Euclidian distance as metric, the neural network then maps the multidimensional input vector into a 2-D grid. Within the grid, the units are visualised as squares, hereby the size, foreground colour and background colour are used to display the different states of the network. Problems with regard to text based tools, were the motivation for the work of Koike et al. [11], who proposed *Snortview* to visualise false positives from *Snort* log files. *Snortview* displays the information within three frames: the source address frame, the alert frame and the destination matrix frame. In the source address frame the source address of hosts detected by the Network Intrusion Detection System (NIDS) are displayed in the vertical axis. In the alert frame, source IP addresses are displayed versus time. Hereby alerts are encoded as coloured icons, the shape of the icon defining the type of attacks and the colour the related priority level. By clicking on a symbol in the source and destination matrix frame, the communication path is highlighted.

Further work can be found in [15] where a 2-D matrix of IP addresses is used to visualise cyber threats, in [8] where the source IP address, the destination IP address and the destination port were used to detect scans and SYN flooding attacks by plotting the number of unsucessfull connections versus time as a 2-D plot (histograph) and in [12] where Kim et al. detect and visualise anomalies based on the distribution of the IP header information.

Our approach is totally different from those presented so far, there is a little in commonn with the approach presented in [12] as they also considered a time series as an image and apply methods from the area of image analysis and video processing for further analysis. In our approach we also represent the data to be analysed as a time series but apply statistical texture analysis methods, co-occurrence matrices to analyse and visualise the network state.

[1]One for well known ports and the other one for ephemeral ports.

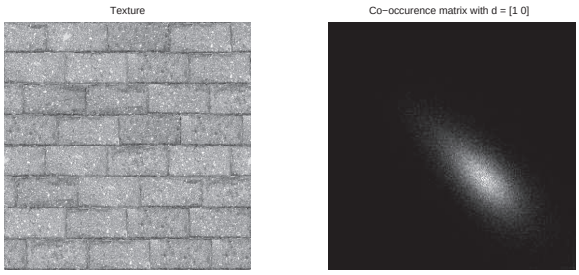

Fig. 1. Texture with brightness coded GLCM

III. Grey Level Co-occurrence Matrix

One challenge in digital image processing is the classification of textures. A digital image, which has been sampled and quantisized can be interpreted as a matrix $\mathbf{G} = g(\vec{x}) = g(n_x, n_y) \in \{0, 1, ..., N_{g-1}\}$ and the description of combinations of pixel brightness values (grey levels) in this image is called *Grey Level Co-occurrence Matrix* (GLCM) $\mathbf{C}(\delta, T) = [s(i, j, \delta, T)]$ or *Grey Tone Spatial Dependency Matrix*.

Each element $s(i, j, \delta, T)$ is a second order probability to go from one grey level i to another grey level j given the displacement vector $\delta = (\Delta x, \Delta y)$. T defines a tile of the original picture. Each element in $\mathbf{C}(\delta, T)$ can be determined as

$$s(i, j, \delta, T) = \frac{\Theta\{\vec{x}|\vec{x}, \vec{x} + \delta \in T, g(\vec{x}) = i, g(\vec{x} + \delta) = j\}}{\Theta\{\vec{x}|\vec{x}, \vec{x} + \delta \in T\}} \tag{1}$$

where Θ denotes the number of elements in each set [16]. The dimension of $\mathbf{C}(\delta, T)$ is N_gxN_g.

In Fig. 1 a texture and the corresponding co-occurrence matrix are shown. Elements along the diagonal of the matrix represent neighboring pixel pairs with less or no difference in the grey level. The farther away from the diagonal the higher the grey level difference becomes.

Haralick et al. proposed 14 criteria extracted from the GLCM to describe a texture [17] and used them as an input vector for a classificator. Conners et al. pointed out six significant parameters from the original 14 in [16] and we use the *Correlation* (8) as a seventh parameter. The parameters with $\sigma_i = \sum_i \sum_j (i-\mu_i)^2 \cdot s(i, j)$ and $\sigma_j = \sum_i \sum_j (j-\mu_j)^2 \cdot s(i, j)$ are defined as follows:

$$ASM = \sum_i \sum_j (s(i, j))^2 \tag{2}$$

$$ENT = -\sum_i \sum_j s(i, j) \cdot \log(s(i, j)) \tag{3}$$

$$IDM = \sum_i \sum_j \frac{s(i, j)}{1 + (i - j)^2} \tag{4}$$

$$INE = \sum_i \sum_j (i - j)^2 \cdot s(i, j) \tag{5}$$

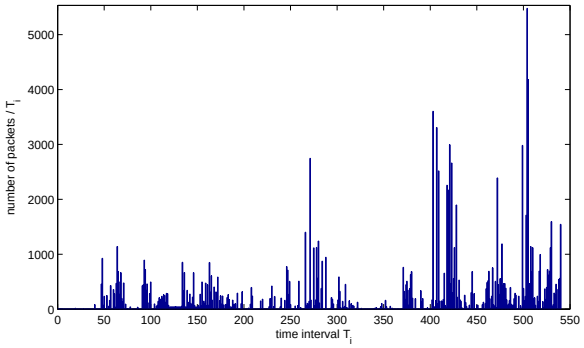

Fig. 2. Typical network traffic time series (T_i = 1 sec)

$$CS = \sum_i \sum_j ((i - \mu_i) + (j - \mu_j))^3 \cdot s(i, j) \tag{6}$$

$$CP = \sum_i \sum_j ((i - \mu_i) + (j - \mu_j))^4 \cdot s(i, j) \tag{7}$$

$$CORR = \sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j) \cdot s(i, j)}{\sigma_i \cdot \sigma_j} \tag{8}$$

The variable μ_i is defined as $\mu_i = \sum_i \sum_j i \cdot s(i, j)$ and $\mu_j = \sum_i \sum_j j \cdot s(i, j)$.

The *Angular Second Moment* (2) describes the energy of the matrix and the *Entropy* (3) reflects the information content. *Inertia* (5) can be interpreted as a contrast to the greyscale image and *Inverse Difference Moment* (4) as an inverse weighted measure of contrast. *Cluster Shade* (6) describes spots with homogene intensity and a high contrast to the remaining structure, the grey level of clusters is characterised by *Cluster Prominence* (7).

IV. Visualisation of Network Traffic

A typical time series for network traffic is shown in Fig. 2. The number of packets captured in a time interval T_i is plotted for successive time intervals. The self-similar and fractal nature of the traffic can be observed as well, there are few peaks and several periods of burst traffic as well as periods with less or even no traffic.

The number of packets captured is a good example for an aggregation function. We use this parameter as an input for our system and are able to analyse multiple levels of details depending on what is of specific interest. In most cases these details are directly related to the protocol type e.g. the number of IP, TCP, UDP or ICMP packets per time interval. Further details such as the number of TCP streams or open TCP connections are the result of a traffic flow analysis. It is also possible to apply filter options to restrict the traffic to a specific protocol such as HTTP or SMTP.

Similar to the windowing mechanism (see T in (1)) in the texture analysis we also use a sliding window of adjustable size to retrieve the parameters, but our basic data is a one dimensional time series of aggregated events in contrast to two dimensional images in texture analysis.

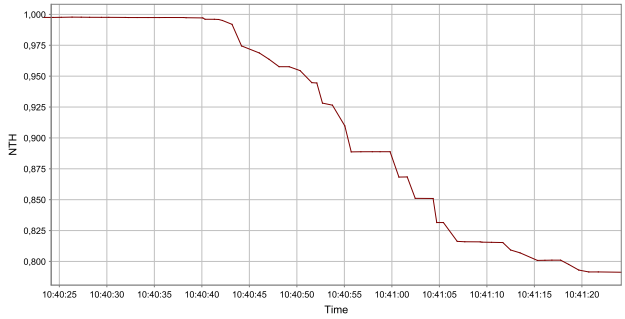

Fig. 3. Network Traffic Homogeneity

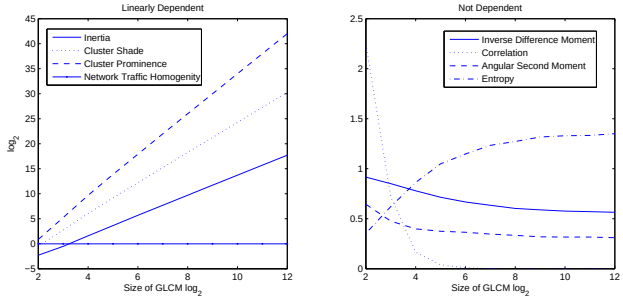

Fig. 4. GLCM parameters as a function of the matrix size N_g

In the scope of texture analysis this two dimensional aspect is taken into consideration in the displacement vector to generate the co-occurrence matrix. A rotation invariant co-occurrence matrix is computed by averaging several GLCMs which are based on different displacement vectors (mostly horizontal, vertical, diagonal-up and diagonal-down). In our case the direction of the vector is implicitly given by the time, thus we have only one displacement vector along the time scale representing the difference between two successive time intervals.

In the following we introduce a new GLCM parameter, discuss the necessary steps to transfer the texture analysis technique to network traffic, present our visualisation software and show our results.

A. A Measurement for Network Traffic Homogeneity

There are several parameters available to describe the roughness of a signal. Most of them, like the roughness of alternating current $w = \frac{\sqrt{2} \cdot i}{I_m}$ or the root-mean-squared roughness for discrete signals $R_q = \frac{1}{\sqrt{N}} \sum_{m=1}^{N} \sqrt{(z(x_m) - \overline{z})^2}$ need the arithmetic mean[2]of the signal. Especially for the non-deterministic and discontinuous nature of network traffic a global measure like the arithmetic mean does not make much sense. Therefore, an additional parameter is needed to locally capture the roughness of our data. This is in line with the co-occurrence matrix which always describes the local behaviour of a signal too. We introduce a new GLCM parameter *Network Traffic Homogeneity* (*NTH*) (9) as a homogeneity measure of the chronological sequence of network traffic.

$$NTH = \sum_i \sum_j s(i,j) \cdot cos\left(\frac{\pi(i-j)}{2N_g}\right) \qquad (9)$$

The characteristics of the texture can be testified to in statistical texture analysis based on the allocation of the co-occurrence matrix. The same can be applied to our network traffic. For example it is known that a strong allocation of the main diagonal is a sign of successive intervals with nearly the same values, whereas a strong matrix allocation on the left border represents jerky leaps to a high value, and a matrix pattern with a strong allocation near the upper border are leaps to a low value.

Fig. 3 shows the progress of *NTH* in a testing scenario. In the first third of the diagram, *NTH* is approximately equal to one, this indicates a very high homogeneity which is caused by no active traffic. A value of one is reached by a constant network traffic. Network traffic is generated manually (HTTP traffic) and the value becomes lower and lower. The effect of the sliding window can be observed directly and the final value of about 0.80 is reached step by step.

B. Determining the GLCM matrix size

In texture analysis, the size of the co-occurrence matrix is explicitly given by the range of the greyscale values, e.g. an 8-bit greyscale image results in a matrix size of 2^8x2^8. In our scenario the co-occurrence matrix is built based on a time series with no explicitly given limit for the values, the limit is given by the bandwidth of the measured link. Without normalisation this would result in a huge matrix size, in addition, a peak in the measured data would also increase the size. Some peaks during very high network activity will result in a matrix size to the magnitude of 10^4x10^4, which is impractical thus, requiring quantisation.

We analysed a linear quantisation to a matrix size of 2^i with $i \in \{2, 3, ..., 12\}$. Fig. 4 shows the computed parameters ((2) to (9)) as a function of the matrix size. On the left side we see a linear dependency of the values *Inertia* (5), *Cluster Shade* (6) and *Cluster Prominence* (7) to the matrix size. The parameter *Network Traffic Homogeneity* (9) has a value of 0.985 nearly constant. From this observation, we can deduce that the distribution of these four parameters does not depend on the size of the co-occurrence matrix. The other values *Inverse Difference Moment* (4), *Correlation* (8), *Angular Second Moment* (2) and *Entropy* (3) do not show such a simple correlation in the diagram of Fig. 4 on the right side. But for values higher than 6, a nearly constant graph for all parameters can be seen. The matrix size is adjustable in our prototype, but, with regard to fast computation, especially in realtime scenarios, a matrix size of 2^6 is quite a good tradeoff between performance on the one hand and detail level on the other.

[2]The arithmetic mean is I_m or $\overline{z}$ respectively.

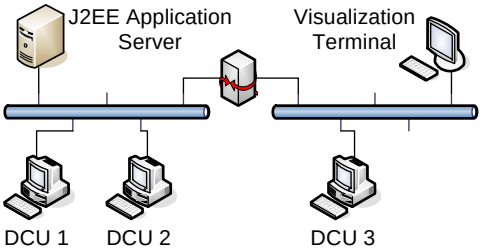

Fig. 5. Distributed Environment

C. System Design

Our prototype is divided into three parts: the data collection unit (DCU), the visualisation frontend (VF) and the data storage middleware (DSM). The data collection units report their data sets to the application server, which can be contacted by each visualisation frontend as shown in Fig. 5, for a simplified scenario in a distributed system environment consisting of two subnets.

The *data collection unit* can access network packets directly at the network interface via the low level packet capture driver `libpcap` or can process captured files saved in the `libpcap`-format. The computational intensive work (computing the GLCM parameters) is done at this point and the results as well as packet header information is then sent to the data storage middleware. Of course, the data collection unit can be directly used within the visualisation frontend for an online network traffic analysis without the overhead of the storage middleware, but this operation mode is limited with regard to user interaction.

The *data storage middleware* is mainly a J2EE application server with two main components: Entity Enterprise JavaBeans (EJB) and Java Message Service (JMS). EJBs are used for data persistence and most of the communication is handled by the JMS. The collected data is sent in a batch to the application server, made persistent and distributed to all listening visualisation frontends. The interval for the batched data transmission is adjustable and can directly be interpreted as the lag between the point of time an event occurs and the moment the impacts are shown. A marginal lag is designated, but especially in an environment with several data collection units, a low value will result in poor overall performance of the application server.

The Java implementation of our *visualisation frontend* is shown in Fig. 6. The upper left side is the main control panel, where different traffic parameters can be selected for analysis. At the left bottom the co-occurrence matrix is shown for the selected packet type. A bigger display of the co-occurrence matrix is presented in Fig. 7. On the right side the statistical texture analysis parameters ((2) to (9)) are plotted in an adjustable time scale. A plot of a subset of parameters of interest can also be performed. If the frontend is used in the distributed mode, connected to the application server, a further analysis of the data stream can be carried out using the VCR functionality (pause, stop, forward and rewind) and timeshifting. If interesting (suspicious) patterns appear, the

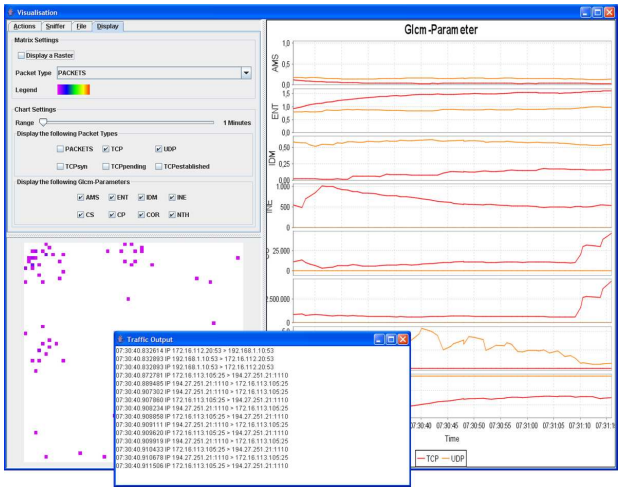

Fig. 6. Java implementation of the visualisation frontend

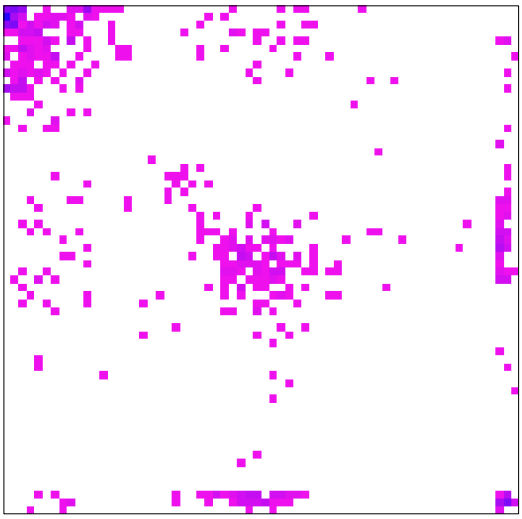

Fig. 7. Visualisation matrix

user can pause the data stream, scroll to the position and can play back the network traffic in text based mode to analyse the time sequence in detail.

V. Conclusion and Future Work

We have presented a novel approach for visualising network traffic by mapping the given time series of network traffic into a co-occurrence matrix and applying statistical texture analysis methods from the domain of digital image processing for the analysis and visual representation of the network traffic.

Fig. 8 shows the graphs for all parameters previously described for three different traffic situations. Beginning with a short section of no active traffic we have manually generated HTTP traffic, we subsequently started a port scan targeted on the data collection unit. For both traffic types (HTTP and port scan), no significant differences were observed in the parameters *ASM*, *ENT* and *IDM*, but *Inertia*, *Cluster Shade* and *Cluster Prominence* (remember, these parameters are independent of the matrix size) clearly indicate the port

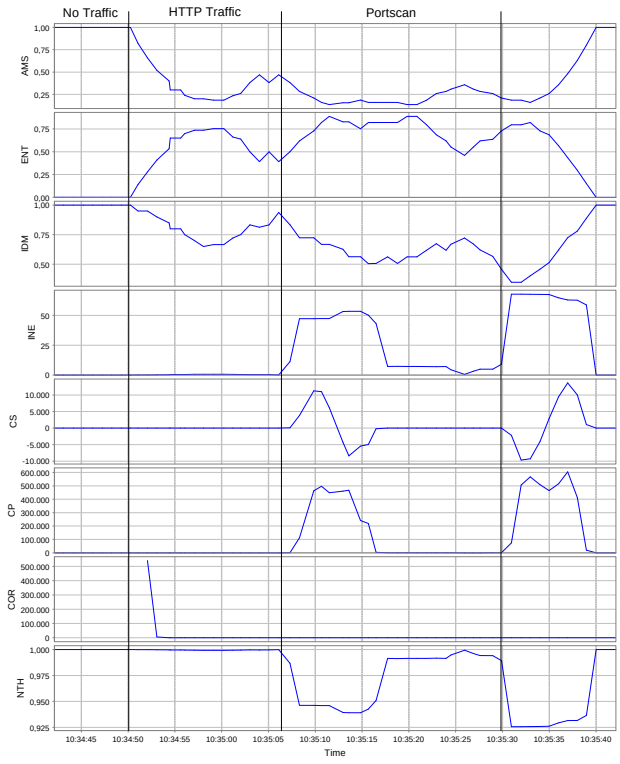

Fig. 8. GLCM parameters of TCP flow analysis data

scanning phase, where especially the wave form of *CS* is characteristic for this type of attack.

The parameter *NTH* (9) is quite a good measure for network traffic homogeneity. In contrast to other calculations this value explicitly keeps the local context of the signal in mind. Even marginal, but steady changes of the traffic, result in little change of the value.

In future work, we will focus on discovering more patterns for specific attacks and offering further user interactivity by extending our visualization model to support multiple levels of granularity to explore the data. Currently we are evaluating our approach with the data sets provided by Lincoln Laboratory (Massachusetts Institute of Technology), the *1998 DARPA Intrusion Detection Evaluation Data Set* with seven weeks of training data and two weeks of testing data including many different (and documented) types of attacks. Due to the increasing rate of Peer-to-Peer traffic this type of traffic will also be investigated in greater detail.

Another aspect for future work is related to our distributed approach. Due to the complex nature of modern attacks, we will analyse the correlation of the feature vectors of different data collection units in the network.

REFERENCES

[1] J. Ryan, M-J. Lin, and R. Miikkulainen, ”Network based intrusion detection using neural networks,” *Advances in Neural Information Processing Systems*, vol. 10, MIT Press, 1998.

[2] W. Lee and S. Stolfo, ”Data mining approaches for intrusion detection,” *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.

[3] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, ”Modeling intrusion detection system using hybrid intelligent systems,” *Journal of Network and Computer Applications*, Elsevier, 2005.

[4] R. Ball, G. A. Fink, and 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 (VizSec/DMSec 2004)*, Washington DC, p. 55-64, 2004.

[5] K. Lakkaraju, W. Yurcik, R. Bearavolu, and A. J. Lee, ”NVisionIP: an interactive network flow visualization tool for security,” *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 3, pp. 2675-2680, 2004.

[6] S. Lau, ”The spinning cube of potential doom,” *Communications of the ACM*, vol. 47, no. 6, pp. 25-26, Jun. 2004.

[7] K. Lakkaraju, W. Yurcik, and A. J. Lee, ”NVisionIP: netflow visualizations of system state for security situational awareness,” *Proc. ACM Workshop Visualization and Data Mining for Computer Security (VizSEC/DMSEC), ACM Press*, pp.65-72, 2004.

[8] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson, ”IDGraphs: intrusion detection and analysis using histographs,” *IEEE Workshop on Visualization for Computer Security (VizSEC 2005)*, Minneapolis, MN, USA, 26 October 2005.

[9] Y. Hideshima and H. Koike, ”STARMINE: a visualization system for cyber attacks,” *In Proc. Asia Pacific Symposium on Information Visualisation (APVIS 2006)*, Tokyo, Japan, pp.131-138, 2006.

[10] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi, ”Preserving the big picture: visual network traffic analysis with TNV,” *Proceedings of the Workshop on Visualization for Computer Security (VizSec*), pp. 47-54, 2005.

[11] H. Koike and K. Ohno, ”Snortview: visualization of Snort logs,” *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security (VizSEC)*, pp. 143-147, 2004.

[12] S. S. Kim and A. L. Narasimha Reddy, ”Netviewer: a network traffic visualization and analysis tool, *”Proceedings of USENIX 19th Large Installation System Administration Conference (LISA 05)*, San Diego, CA, USA, Dec. 2005.

[13] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko, ”IDS RainStorm: visualizing IDS alarms,” *Proceedings of the 2005 IEEE Visualization for Computer Security (VizSec 05)*, pp. 1-10, Oct. 2005.

[14] T. Itoh, H. Takakura, A. Sawada, and K. Koyamada, ”Hierarchical visualization of network intrusion detection data,” *IEEE Computer Graphics and Applications*, vol. 26, no. 2, pp. 40-47, March 2006.

[15] K. Ohnof, H. Koikef, and K. Koizumi, ”IPMatrix: an effective visualization framework for cyber threat monitoring,” *Proceedings of the Ninth International Conference on Information Visualisation (IV05)*, London, England, IEEE/CS, pp. 678-685, 2005.

[16] R. W. Conners, M. M. Trivedi, and C. A. Harlow, ”Segmentation of a high-resolution urban scene using texture operators,” *Computer Vision, Graphics and Image Processing*, vol. 25, pp. 273-310, 1984.

[17] R. M. Haralick, K. Shanmugam, and I. Dinstein, ”Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610-621, November 1973.

[18] O. Mizuki, O. Yoshihiro, A. Hirotake, and K. Kazuhiko, ”Anomaly detection using layered networks based on Eigen Co-occurence Matrix,” *Recent Advances in Intrusion Detection*, Nice, France, pp. 223-237, 15-17 September 2004.

[19] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich ”Data Mountain: using spatial memory for document management, ” *In Proceedings of UIST 98, 11th Annual Symposium on User Interface Software and Technology*, pp. 153-162, 1998.

[20] L. Girardin, ”An Eye on network intruder-administrator shootouts,” *Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, pp. 19-28, 1999.

[21] T. Kohonen, *Self-Organizing Maps*, Springer Series in Information Sciences, vol. 30, Springer, Third extended edition, 2001.