Efficient Presentation of Multivariate Audit Data for Intrusion Detection of Web-Based Internet Services

Zhi Guo[1], Kwok-Yan Lam[1], Siu-Leung Chung[2], Ming Gu[1], and Jia-Guang Sun[1]

[1] School of Software, Tsinghua University, Beijing, PR China
{gz, lamky, guming, sunjg}@tsinghua.edu.cn
[2] School of Business Administration, The Open University of Hong Kong
slchung@ouhk.edu.hk

Abstract. This paper presents an efficient implementation technique for presenting multivariate audit data needed by statistical-based intrusion detection systems. Multivariate data analysis is an important tool in statistical intrusion detection systems. Typically, multivariate statistical intrusion detection systems require visualization of the multivariate audit data in order to facilitate close inspection by security administrators during profile creation and intrusion alerts. However, when applying these intrusion detection schemes to web-based Internet applications, the space complexity of the visualization process is usually prohibiting due to the large number of resources managed by the web server. In order for the approach to be adopted effectively in practice, this paper presents an efficient technique that allows manipulation and visualization of a large amount of multivariate data. Experimental results show that our technique greatly reduces the space requirement of the visualization process, thus allowing the approach to be adopted for monitoring web-based Internet applications.

Keywords: Network security, Intrusion detection, Multivariate data analysis, Data visualization

1 Introduction

Effective security measures for protecting web-based services are important to organizations providing Internet applications. With the pervasive use of the Internet as a channel for delivering services, distributed applications are often implemented as Internet services over the web architecture. Internet applications usually adopt the 3-tier architecture consisting of a database layer, an application server layer and a web server layer. It is the nature of Internet applications that web servers are open to access from the global Internet. Thus web servers are exposed to serious security threats, and computer crimes targeting at web sites have become increasingly ubiquitous due to vulnerabilities of web servers.

J. Zhou, M. Yung, Y. Han (Eds.): ACNS 2003, LNCS 2846, pp. 63–75, 2003.

Security monitoring is an important requirement in the protection of computer systems against attackers and malicious users. According to a CSI/FBI Computer Crime and Security Survey [1], 90% of large corporations and government agencies in the US detect computer security breaches and 80% acknowledged financial losses due to these breaches. Hence, for organizations with mission-critical Internet applications, such as financial institutions that provide e-financial services through the Internet, the immediate alert of and investigation into attacking and malicious activities are of vital importance to avoid asset losses as well as reputation losses.

Intrusion detection is an effective way to provide security monitoring function. The idea of intrusion detection [2,3] is based on the hypothesis that computers are typically involved in specific types of activity, and the set of programs and commands they execute will normally reflect that activity. Hence, security violations can be detected from abnormal patterns of system use in that exploitation of a system's vulnerabilities involves abnormal use of the system.

Intrusion detection systems typically consist of two major system components, the intrusion detector and the system administration console, as depicted in Figure 1. The intrusion detector compares ongoing activities with established behavior profile and detects abnormal activities by identifying deviation of these ongoing usage patterns from the established behavior profile. Behavior profiles are established based on audit records of activities collected under normal usage conditions. Should abnormal behavior be detected, an intrusion alert is raised and detailed information about the suspicious activities is presented to the security administrator at the system administration console. The administrator then investigates the information closely and decides whether the alert should be escalated to a higher level in accordance with the organization's security policy.

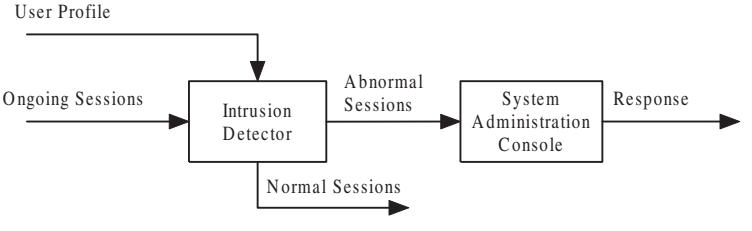


Fig. 1. The intrusion detection scheme.

Statistical-based analysis is an important area in intrusion detection [4,5,6, 7]. These techniques need to have the capability of analyzing a huge volume of multivariate data. Intrusion detection techniques typically involve the manipulation and analysis of audit data. Audit data of computer systems are mostly

multivariate in nature (e.g. measures of set of commands executed or web resources accessed by users). Furthermore, for web-based systems, audit data are voluminous. For example, the web-based digital library service at our university serves over 100,000 clicks per day with over 2,700 different types of web resources being accessed, thus generating a huge amount of audit data to be processed by the intrusion detection system.

A profiling stage that aims to create behavior profiles of the target system is important to the effectiveness of intrusion detection techniques. During the profiling stage, the target system is monitored over a period of time with audit logs being collected and analyzed. The historical activity logs of the system are summarized in form of behavior profiles of the target system. Depending on the underlying hypothesis, the profiles may be created based on different statistical approaches such as correlation analysis or other statistical measures [5,6,7]. The security administrator may closely inspect and fine-tune the profiles. In this connection, human inspection of the audit data is typically required. It is a basic cognitive nature of human being that visual display of complex information is usually more comprehensible than other form of information. Thus, efficient techniques for visualizing the multivariate audit data play an important role in intrusion detection.

In general, the system administration console is the human-machine interface of the intrusion detection system that allows the system administrator to perform visual inspection of audit data. In practice, this system administration console is most likely a machine with limited computing resources such as a high-end PC. Since the audit data to be inspected are voluminous, it may not be feasible to perform computations and manipulation of the multivariate data on such machines. Hence an efficient technique that allows manipulation and visualization of a large amount of multivariate data is needed.

This paper presents an efficient implementation technique for presenting multivariate audit data needed by statistical-based intrusion detection systems. The rest of the paper is organized as follows: Section 2 reviews an important technique for analyzing and visualizing multivariate audit data for intrusion detection purposes. The space requirement of the visualization technique becomes prohibiting when processing audit logs collected on web-based Internet applications. A new data visualization approach with significant improvement in space requirement is presented in Section 3. Experimental results that show the effectiveness of the approach are presented in Section 4. This paper is concluded by Section 5.

2 Multivariate Statistical Analysis of Audit Trails

Multivariate data analysis has been adopted successfully to detect misuses of computer systems [5,6,7]. Depending on the underlying intrusion detection hypothesis, different multivariate statistical analysis such as correlation analysis or

other statistical measures may be applied to analyze audit logs. For instance, the correlation analysis approach is based on the hypothesis that users involving in specific types of applications will execute commands that are strongly correlated. This is especially true in web-based applications where web resources are usually correlated due to the applications' design nature; e.g. multi-frame pages will cause multiple web resources to be accessed simultaneously.

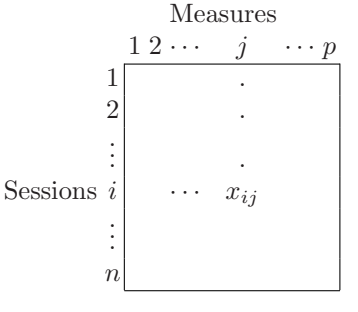
The multivariate data analysis technique is used to create behavior profiles of intrusion detection systems. For example, correlation analysis helps select independent intrusion detection measures [5]. It may also be used to detect anomaly activities by analyzing correlation between user sessions [7]. In this case, intrusion detection is based on the hypothesis that activities of a normal user are strongly correlated, thus leading to the formation of a normal behavior profile. Ongoing user activities are compared against the normal behavior profile. Basically, the correlation between ongoing user activities and the user profile is computed. Ongoing activities that are strongly correlated with the profile are considered normal.

There are three steps in the multivariate data analysis intrusion detection model to detect misuses in web applications. Firstly, intrusion detection measures are extracted from the web server's audit log. The extracted data are represented in multivariate data form. Secondly, correlation analysis on the measures is performed. Thirdly, visualization technique is performed to allow the analysis results to be closely inspected by the administrator.

2.1 Data Representation

Before multivariate data analysis can be performed to detect intrusion, intrusion detection measures on which the intrusion detection model is based need to be defined. In the context of intrusion detection of web-based applications, it is the measures of web resources within user sessions that constitute the multivariate data to be analyzed. In our study, the frequency of access to each web resource is used as an intrusion detection measure.

With user session and intrusion detection measures defined, we can represent the data for multivariate data analysis in the form of a matrix M as follow:



where x_{ij} represents the frequency count of access to web resource j $(1 \leq j \leq p)$ in session i $(1 \leq i \leq n)$, i.e. the frequency count of accessing the j^{th} web resources in the i^{th} time interval (session). An example of M is shown in Figure 2.

	/index.htm	/bookview/index.htm	...	/Java/jv30.htm	/Java/jv31.htm
2003/2/20 8:00	5	6		0	0
2003/2/20 8:01	3	1		0	0
...					
2003/2/27 17:42	7	8		2	0
2003/2/27 17:43	12	6		0	1

Fig. 2. An example of M.

Geometrically, the matrix M can be viewed as consists of n row vectors, $\overrightarrow{X}_1, \cdots, \overrightarrow{X}_n$, in a p-dimensional (p-D) Euclidean space or p column vectors, $\overrightarrow{Y}_1, \cdots, \overrightarrow{Y}_p$, in an n-D Euclidean space. The n^{th} row vectors represent to n^{th} sessions and the p^{th} column vectors represent the p^{th} measures in the web log.

2.2 Correlation Analysis

As mentioned, an effective way to detect misuses is to test for correlation between ongoing user sessions and the behavior profile. Mathematically, a straightforward way is to compute and analyze the coefficient of correlation r_{ij} between user sessions such that

$$r_{ij} = \frac{1}{p} \sum_{k=1}^{p} \frac{(x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)}{\sigma_i \sigma_j}, \quad 1 \leq i, j \leq n \tag{1}$$

where

$$\overline{x}_i = \frac{\sum_{j=1}^{p} x_{ij}}{p} \quad , \quad 1 \leq i \leq n \tag{2}$$

is the mean of the vector of row i and

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{p} (x_{ij} - \overline{x}_i)^2}{p}} \quad , \quad 1 \leq i \leq n \tag{3}$$

is the standard deviation of the vector in row i.

However, it would be inefficient to compute the pair-wise coefficient of correlation for all the row vectors in M. Instead, the matrix is transformed so that correlated sessions can be identified by a cluster recognition approach using, for example, a minimum spanning tree (MST) algorithm. The transformation involves normalization of the row vectors in M and is described in the following:

$$\widetilde{x}_{ij} = \frac{(x_{ij} - \overline{x}_i)}{\sigma_i \sqrt{p}} \quad 1 \leq i \leq n, 1 \leq j \leq p. \tag{4}$$

Let the normalized matrix be $\widetilde{M}$, and let the i^{th} normalized row vector be $\widetilde{X}_i$. Geometrically, for any two normalized row vectors $\widetilde{X}_i$ and $\widetilde{X}_j$, which represent two points in p-D Euclidean space, their distance satisfies the following:

$$d^2(\widetilde{X}_i, \widetilde{X}_j) = 2(1 - r_{ij}) \tag{5}$$

where r_{ij} is the coefficient of correlation between $\widetilde{X}_i$ and $\widetilde{X}_j$. Thus, points which are close to each other are closely correlated (with a relatively small r_{ij}). In this respect, to recognize groups of closely correlated vectors (sessions), a cluster recognition technique can be used.

2.3 Visualization of Multivariate Data

Data visualization technique enables human inspection of the distribution of audit data. For example, abnormal activities can be identified visually if observed data deviate from the normal user profile. Since the number of measures in the audit data is very large, it is necessary to project points in high dimensional space to 2-D (or 3-D) space to allow visualization.

Different dimensionality reduction techniques based on unsupervised data, include factor analysis with principal components analysis (PCA), multidimensional scaling (MDS), Sammon's mapping (SAM), Kohonen's self-organizing maps (SOM) and auto-associative feedforward neural networks (AFN), and so on, are widely adopted to extract the major features of a data pattern in image processing [8,9]. In our study, we adopt factor analysis [10] which, as suggested in [7], projects high dimensional audit data to 2-D or 3-D space and at the same time preserve the major characteristics of the data pattern.

Factor analysis for dimensionality reduction is described as follows:

1. Compute $\widetilde{M}$ with row vectors equal the normalized row vectors of M.
2. Compute $\widetilde{M}^T\widetilde{M}$.
3. Compute the p eigenvalues $\lambda_1, \cdots, \lambda_p$ of $\widetilde{M}^T\widetilde{M}$ and the corresponding p orthonormal eigenvectors $\overrightarrow{u}_1, \cdots, \overrightarrow{u}_p$.
4. Project the n normalized points (i.e. the n normalized row vectors in $\widetilde{M}$) in p-D space to 2-D space with coordinates $(\widetilde{X}_i^T\overrightarrow{u}_1, \widetilde{X}_i^T\overrightarrow{u}_2)$, $i = 1 \ldots n$.

As mentioned in the previous section, audit data visualization is an important step in intrusion detection in that a more accurate estimation of the normal user profile can be achieved. In this respect, the volume of audit data that need to be processed and visualized is inevitably huge in size. For example, audit data extracted from the web log of our digital library are collected for a period of one week, the corresponding matrix M consists of about 600,000 rows (i.e.

sessions). In this case, M is huge and the space complexity of factor analysis is very high. Computing resources may not be enough to make the intrusion detection processes infeasible.

In the next section, a new approach for improving the space complexity of the data visualization processed is presented.

3 The Efficient Data Visualization Technique

A major obstacle to visualization of audit data for intrusion detection is the high space requirement in the matrix computation of the normalized matrix $\widetilde{M}$. In order to reduce the space complexity, a new approach is proposed for the data visualization process. With this approach, the space complexity of the visualization process is largely reduced thus allowing visualization to be executed with a reasonable amount of computing resources.

To reduce the space complexity for data visualization, we investigate the characteristics of the matrices to be processed and try to reduce the space requirement based on these characteristics. In a web application, the possible number of intrusion detection measures (web resources accessible by users) can be very large. In our digital library example, the number of web resources accessible is in the range of 2,000 to 3,000. However, the set of web resources accessed during a typical user session is usually much smaller than the total number of measures available on the web application. Hence, the resulting matrices will be sparse and with a majority of the elements (i.e. frequency count of web resources access within the sessions) equal zero. From our experiment with the web portal of a digital library, over 99 percent of the value of x_{ij} of the matrix M equal zero. Based on this characteristic of the matrices, we can leverage on some algorithms for sparse matrix computations [11] to further enhance the data visualization process.

Therefore, one effective way to improve the visualization mechanism is to take advantage of the sparseness of the matrix M. However, this advantage is hard to realize in practice because, according to the factor analysis technique as presented in the previous section, visualization is performed on the normalized data set. It is worth noting that the cluster recognition and inspection is applicable only to the normalized matrix $\widetilde{M}$. Unfortunately, the sparse matrix M will no longer be sparse after normalization. That means the visualization scheme cannot be improved unless the factor analysis algorithm is applied to the sparse matrix M instead of the normalized matrix $\widetilde{M}$.

Referring back to the data visualization algorithm presented in the previous section, the most intensive computation is dedicated to the matrix multiplication $\widetilde{M}^T\widetilde{M}$. As stated, $\widetilde{M}$ is transformed from M with the row vectors normalized. It has been pointed out that M is a sparse matrix, however, $\widetilde{M}$ will not be sparse

after normalization. Thus, we need to investigate the matrix multiplication steps carefully in order to make use of the sparseness of the original matrix M.

Theorem 1. *Let the matrix*

$$M = [x_{ij}], \quad 1 \le i \le n, 1 \le j \le p \tag{6}$$

and

$$\widetilde{M} = [\widetilde{x}_{ij}], \quad 1 \le i \le n, 1 \le j \le p \tag{7}$$

be the row-wise normalized matrix of M. Then

$$\widetilde{M}^T\widetilde{M} = Y^TY - \begin{pmatrix} \varepsilon_1 \cdots \varepsilon_1 \cdots \varepsilon_1 \\ \vdots \quad \vdots \quad \vdots \\ \varepsilon_s \cdots \varepsilon_s \cdots \varepsilon_s \\ \vdots \quad \vdots \quad \vdots \\ \varepsilon_p \cdots \varepsilon_p \cdots \varepsilon_p \end{pmatrix} - \begin{pmatrix} \varepsilon_1 \cdots \varepsilon_t \cdots \varepsilon_p \\ \vdots \quad \vdots \quad \vdots \\ \varepsilon_1 \cdots \varepsilon_t \cdots \varepsilon_p \\ \vdots \quad \vdots \quad \vdots \\ \varepsilon_1 \cdots \varepsilon_t \cdots \varepsilon_p \end{pmatrix} + c \tag{8}$$

where

$$\alpha_k^2 = \sum_{r=1}^{p} (x_{kr} - \overline{x}_k)^2, \quad \beta_k = \overline{x}_k, \quad 1 \le k \le n \tag{9}$$

$$\varepsilon_t = \sum_{k=1}^{n} \frac{\beta_k x_{kt}}{\alpha_k^2}, \quad 1 \le t \le p \tag{10}$$

$$c = \sum_{k=1}^{n} \frac{\beta_k^2}{\alpha_k^2} \tag{11}$$

$$Y = [y_{kt}], \quad y_{kt} = \frac{x_{kt}}{\alpha_k}, \quad 1 \le k \le n, 1 \le t \le p. \tag{12}$$

Proof *Given $\widetilde{M}$ is the row-wise normalized matrix of M, then*

$$\widetilde{M}^T\widetilde{M} = [\widetilde{m}_{st}], \quad 1 \le s, t \le p \tag{13}$$

with

$$\widetilde{m}_{st} = \sum_{k=1}^{n} \widetilde{x}_{ks}\widetilde{x}_{kt} \tag{14}$$

$$= \sum_{k=1}^{n} \frac{(x_{ks} - \overline{(x)}_k)}{\sigma_k\sqrt{p}} frac(x_{kt} - \overline{x}_k)\sigma_k\sqrt{p} \tag{15}$$

$$= \sum_{k=1}^{n} \frac{x_{ks}x_{kt} - \overline{x}_k x_{ks} - \overline{x}_k x_{kt} + \overline{x}_k^2}{\sum_{r=1}^{p} (x_{kr} - \overline{x}_k)^2} \tag{16}$$

With $\alpha_k^2, \beta_k, \varepsilon_t, c, y_{kt}$ *as defined, then*

$$\widetilde{m}_{st} = \sum_{k=1}^{n} \frac{x_{ks}x_{kt}}{\alpha_k \alpha_k} - \varepsilon_s - \varepsilon_t + c \tag{17}$$

$$= \sum_{k=1}^{n} y_{ks}y_{kt} - \varepsilon_s - \varepsilon_t + c \tag{18}$$

□

Theorem 1 allows us to improve the efficiency of the visualization process by substantially reducing the space complexity of the factor analysis algorithm. Due to the results of Theorem 1, the matrix multiplication $\widetilde{M}^T\widetilde{M}$ can be processed as follow:

1. Given matrix M, compute $\beta_k, \beta_k^2, \alpha_k^2, \alpha_k$ for $k = 1 \cdots n$.
2. Compute ε_t for $t = 1 \cdots p$.
3. Compute the constant c.
4. Compute the sparse matrix Y.
5. Compute the matrix multiplication Y^TY.
6. Compute the matrix multiplication $\widetilde{M}^T\widetilde{M}$ according to Equation 8.

Note that the matrix M is sparse and so is Y. To appreciate the improvement due to Equation 8, we analyze and compare the space complexity of the matrix multiplication using both the straightforward method and the new method.

For the memory requirements, if the straightforward multiplication $\widetilde{M}^T\widetilde{M}$ is used, the memory requirement for storing $\widetilde{M}$ is $O(np)$ and for storing $\widetilde{M}^T\widetilde{M}$ is $O(p^2)$. Hence, the total memory requirement will be $O(np+p^2)$. If Equation 8 is used, since Y is sparse and if r is the ratio of non-zero elements to the size of the matrix, then $r \ll 1$. The memory requirement for storing Y is $O(npr)$ and for storing $\beta_k, \beta_k^2, \alpha_k^2, \alpha_k$ and ε_t will be $O(4n+p)$. Totally, the memory requirements will be $O(npr + p^2 + 4n + p)$. Since $r \ll 1$ and $n \gg p \gg 1$, hence, the memory requirements for the matrix multiplication using Equation 8 is much smaller than the straightforward multiplication of $\widetilde{M}^T\widetilde{M}$.

4 Experimental Analysis

The proposed data visualization algorithm was implemented and experiments on its performance were conducted. Audit data from the access log of our university digital library portal were used for the experiments. The library portal has about 100,000 clicks per day. Audit data of one week (from 20 Feb 2003 to 27 Feb 2003) with a total of 665,902 access records were extracted. The corresponding matrix M has 2,742 columns and 9,717 rows. The ratio of non-zero entry, r, of the

matrix M is 0.905%. The experiments were conducted on a high-end PC with dual 1GHz Pentium CPUs and 1G bytes memory. One CPU and 500M bytes is reserved for the execution of the computations.

Figure 3.a depicts the 2-D projection of the points of M using the traditional data visualization algorithm while Figure 3.b depicts the same projection using the new algorithm. The results show that the new algorithm performs as good as the traditional algorithm. That means the new algorithm achieves substantial reduction in space complexity without compromising projection quality.

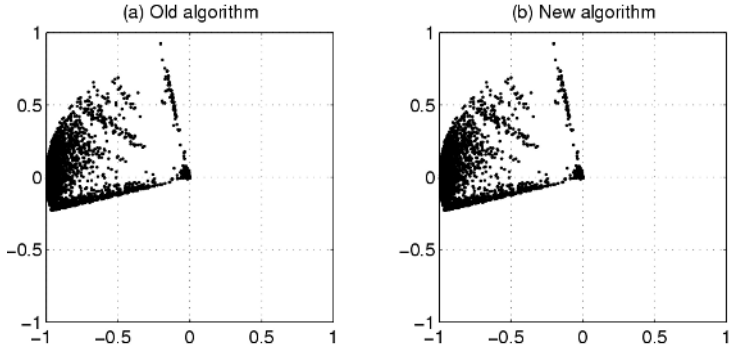


Fig. 3. 2-D projection of points represented by row vectors in matrix M.

To highlight the performance of the new algorithm, two sample data sets were extracted from M. In Data Set 1, the number of measures was fixed while 9 samples with varying numbers of sessions were extracted. In Data Set 2, the number of sessions was fixed while 9 samples with varying numbers of measures were extracted. The memory requirements of the computations are presented in Figure 4. The two data sets are as follow:

Data Set 1				Data Set 2			
Matrix Name	Number of Sessions	Number of Measures	r	Matrix Name	Number of Sessions	Number of Measures	r
M_{11}	1000	2700	1.226%	M_{21}	5000	300	0.503%
M_{12}	2000	2700	1.089%	M_{22}	5000	600	0.424%
M_{13}	3000	2700	0.923%	M_{23}	5000	900	0.495%
M_{14}	4000	2700	0.903%	M_{24}	5000	1200	0.661%
M_{15}	5000	2700	0.894%	M_{25}	5000	1500	0.856%
M_{16}	6000	2700	0.874%	M_{26}	5000	1800	0.915%
M_{17}	7000	2700	0.868%	M_{27}	5000	2100	1.033%
M_{18}	8000	2700	0.928%	M_{28}	5000	2400	0.943%
M_{19}	9000	2700	0.929%	M_{29}	5000	2700	0.894%

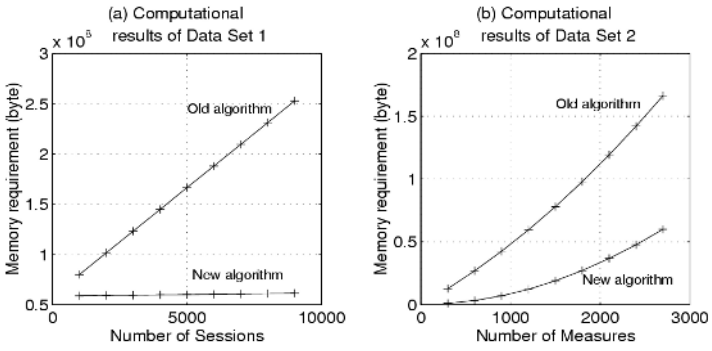


Fig. 4. Comparison of memory requirements for old and new algorithms.

From the results as depicted in Figure 4, it is obvious that the memory requirements were lower for the new algorithm with both varying number of sessions and measures. The improvement is more significant for Data Set 1. It indicates that for the traditional algorithm, the memory consumption grows linearly with the increase of number of sessions. Whereas, with the new algorithm, the increase in memory consumption is much more moderate when number of sessions increases. This is a significant improvement since, as mentioned, visualization of multivariate audit data is an important step in the profiling process. Therefore, the new algorithm out-performs the traditional algorithm significantly and is more suitable for use in practical intrusion detection systems.

5 Conclusion

Multivariate data analysis is an important tool in statistical intrusion detection systems. Intrusion detection typically involves a profiling process, an intrusion detection process and an alert inspection process. Behavior profile of the target system is usually established by statistical analysis techniques or correlation analysis techniques. Intrusion detection alert is raised if the system behavior deviates from the established pattern. Thus creation of the profile and close inspection of its deviation from ongoing activities are of great importance to the performance of the intrusion detection scheme.

Multivariate statistical intrusion detection systems usually require visualization of the multivariate audit data in order to facilitate close inspection by security administrators during profile creation and intrusion alerts. Due to the importance of the profile creation process, security administrators tend to inspect the established profile closely and perform fine-tuning of the profile if possible. Besides, during intrusion detection, the security administrator will also inspect the multivariate audit data closely in order to

decide the event warrants an escalation of the incident to a higher level in accordance with the organization's security policy. Visualization is therefore commonly required as a means for close inspection of the multivariate audit logs.

This paper presented an efficient implementation technique for presenting multivariate audit data needed by statistical-based intrusion detection systems. Though visualization of multivariate data has been in use in the past for supporting statistical-based intrusion detection systems, the need for performing host-based intrusion detection on web-based Internet applications introduced new challenges to the problem. As explained in this paper, the number of intrusion detection measures that need to be analyzed by the multivariate analysis technique increases substantially due to the large number of URL resources managed by typical web servers. Thus, when applying the intrusion detection scheme to web-based Internet applications, the space complexity of the visualization process is usually prohibiting. In order for the approach to be adopted effectively in practice, an efficient technique that allows manipulation and visualization of a large amount of multivariate data is needed. The efficient technique presented in this paper is proven to achieve the same level of visualization quality, and experimental results reported in this paper showed that the new technique greatly reduces the space requirement of the visualization process, thus allowing the approach to be adopted for monitoring web-based Internet applications.

Acknowledgement. We thank the System Division of the Tsinghua University Library for their support in the experiments of this research. This research was partly funded by the National 863 Plan (Projects Numbers: 2001AA414220, 2001AA414020), P. R. China and the Computer Systems Intrusion Detection project of PrivyLink International Limited (Singapore).

References

1. Richard Power. "2002 CSI/FBI Computer Crime and Security Survey". *http://www.gocsi.com*, 2002.
2. D.E. Denning, "An intrusion detection model". *IEEE Trans on Software Engineering*, SE-13, pp 222–232, 1987.
3. R.K. Cunningham, R.P. Lippmann, D.J. Fried, S.L. Garfinkel, T. Graf, K.R. Kendall, S.E. Webster, D. Wyschogrod, M.A. Zissman, "Evaluation Intrusion Detection Systems without Attacking your Friends: The 1998 DAPRA Intrusion Detection Evaluation". Lincoln Laboratory, MIT, USA.
4. E. Biermann, E.Cloete, L.M. Venter. "A Comparison of Intrusion Detection Systems". *Computers & Security*, 20, pp.676–683, 2001.
5. N. Ye, S. M. Emran, Q. Chen, S Vilbert. "Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection". *IEEE Trans. on Computers*, Vol. 51, No. 7, 2002.
6. Nong Ye, Xiangyang Li, Qiang Chen, Syed Masum Emran, Mingming Xu. "Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data". *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol.31, No.4, 2001.

7. Kwok-Yan Lam, Lucas Hui, Siu-Leung Chung. "Multivariate Data Analysis Software for Enhancing System Security". *J. Systems Software*, Vol.31, pp 267–275, 1995.
8. S. De Backer, A. Naud, P. Scheunders. "Non-linear dimensionality reduction techniques for unsupervised feature extraction". *Pattern Recognition Letters*, 19, pp 711–720, 1998.
9. L. Girardin, D. Brodbeck. "A visual appraoch for monitoring logs". *Proc. of the Twelth Systems Administration Conf.*, p. 299, USENIX Association: Berkeley, CA, 1998.
10. Bill Jacob. *Linear Algebra*. New York: Freeman, 1990.
11. Golub G H, Yon Lean C.F. *Matrix Computation*. John Hopkins Univ Press, 1983.