MASS

Model for an Auditing Security System

A. LIEBENBERG[1], J.H.P. ELOFF[2]

[1]*aliebenberg@monotix.co.za*
[2]*eloff@rkw.rau.ac.za*
Department of Computer Science
Rand Afrikaans University
PO Box 524
AUCKLAND PARK
2006
South Africa
March 1999
Tel: +27 11 883-8692 Fax: +27 11 883-8470

Key words: Neural network, expert system, auditing, real time sales system, visualisation, information security.

Abstract: This paper describes a method, by means of an example, which may be utilised to identify risk patterns in an audit log file. A prototype for auditing is develop by looking into several techniques, such as Artificial Intelligence (AI), Expert Systems (ES), Compression, Visualisation and Neural Networks (NN). The MASS (Model for an Auditing Security System) model described in this paper consists out of three components - an Expert System, a Neural Network and the Visualisation of the output. MASS is demonstrated by means of an example which uses an online sales system audit log file.

1. INTRODUCTION

Audit log files of online transaction based systems may contain in excess of a million transactions, which makes the task of identifying risks complex and time consuming. Different techniques are used for auditing transactions today, but all have their own shortcomings. Techniques used currently include spot checks and electronically produced graphs. Spot checks are

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*

done on a certain percentage of the transactions at irregular intervals. This technique will not always pick up all the risks, but has the advantage that the expert personnel responsible for the auditing have fewer transactions to deal with. These personnel are a risk in itself, and a margin for human error has to be allowed for. Although graphs may give an instant view of all the transactions, they are usually compiled by using a limited number of characteristics of each transaction. For example, the graph may show all transactions according to time, but does not include any other information about the transaction.

By visualising the transactions using various risk factors, fraudulent transactions can be detected more easily. Risk factors can be calculated by looking at each individual transaction as well as the context in which the transaction appears. For example, in case of individual transactions the time and date of the transaction could be important. The context of transactions can be evaluated by looking at the success of a specific transaction, together with the success of successive transactions. If a number of successive transactions were unsuccessful a user may be attempting to execute without the necessary authorisation. An example of these visualised transactions is displayed in figure 6.

2. MASS PROTOTYPE

The **MASS prototype** describes a method that addresses the problem of identifying risk transactions in an audit log file. Risks in the log file are identified through the use of visualisation techniques. In the original model, colour was used to indicate the risk level connected to an individual transaction. Since the proceedings of the World Computer Congress 2000 are only printed in black and white, colour was substituted with patterns.

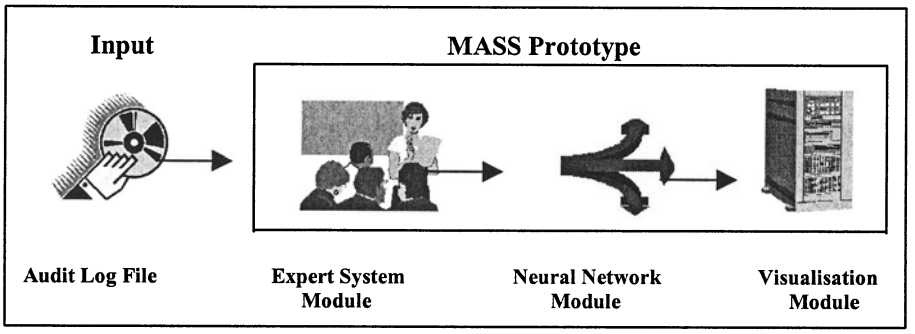

Figure 1. Configuration of the MASS Prototype

MASS consists out of three components – an Expert System Module, a Neural Network Module and a Visualisation Module. Figure 2 displays the

configuration of the MASS prototype. Each module is discussed in more detail in the following sections.

2.1 The Audit Log File

Computer security auditing constitutes an important part of any organisation's security procedures. Because of the many inadequacies of the currently used methods, thorough and timely auditing is often difficult to obtain. The **audit log file** used in the MASS prototype was taken from an online sales environment. This data is stored in a relational database table with the following structure:

Table 1. Structure of the Audit Log Table

Field Name	Description
UserType	The type of user, '1' indicates an external user while '0' indicates the user is internal.
TranType	The type of transaction, 'W' indicates a write transaction while 'R' indicates a read transaction.
TTime	The time the transaction was processed
TDate	The date the transaction was processed
WDay	This is a derived field indicating whether the transaction took place during the week or over weekends. The value of the day of the week is stored, for example Sunday = 1 and Tuesday = 3.
UserCode	The code of the user responsible for the transaction.
Successful	Indicates whether or not the transaction was successful.

Table 2 displays an extraction of the table. The table contains information regarding the transactions that occurred on a mainframe system, for example times and dates of sales transactions. Transaction 1 is extracted and discussed, to illustrate the working of the MASS model in the following sections.

Table 2. Extraction from the Audit Log Table

No	User Type	Tran Type	Conv .Tran Type	Conv. Tran Time	Tran Time	Tran Date	Week Day	User Code	Success
1	1	1	W	17:55:00	175500	01/01/1999	6	1005	Yes
2	0	1	W	17:55:00	175500	01/01/1999	6	201	Yes
3	1	1	W	10:00:34	100034	02/01/1999	7		Yes
4	0	1	W	17:55:00	175500	02/01/1999	7	1014	Yes
5	0	1	W	17:55:01	175501	02/01/1999	2	214	No
6	0	0	R	17:55:01	175501	02/01/1999	7	1014	No

2.1.1 Transaction 1: An example of a write transaction

- The user type indicates that an external user, logged on to the network from outside the premises, executed the transaction.
- The transaction type indicates a write transaction.
- This transaction was executed after normal working hours (08:00 to 17:00).
- The weekday field indicates that this transaction occurred during a normal working day.
- This transaction was successful.
- This transaction is the first of five successive write actions. Table 2 shows that transaction 1,2,3,4 and 5 are all write transactions

The Expert System Module processes the transactions contained in the Audit Log File.

2.2 Expert System Module

Artificial Intelligence (AI) techniques are successfully used in a variety of information security applications. The AI technique used in the MASS prototype is a knowledge-based **expert system**. Since expert systems can provide explanations for recommended actions, it is very suitable for auditing purposes [YELR95].

Another advantage of using a knowledge-based expert system is the fact that a selection of knowledge from several experts can be combined into one system. Research indicates that expert systems are best suited to tackle unstructured audit tasks [ABDO91].

Two types of rules are contained in the expert system - Business Rules and Context Rules. Business rules describe business processes and possible danger areas. For example, if the transaction was executed outside normal working hours it should indicate a slight risk, since this is very unlikely in a sales environment.

Rules describing how each transaction should be evaluated in context are classified as context rules. For example if five successive transactions from the same entry point were unsuccessful, a user might try to execute transactions for which no security clearance exist. Table 3 and 4 display extractions of business and context rules from the MASS prototype. These rules are explained by means of an example in the following paragraphs.

The knowledge base information represented by the above business and context rules is used to convert each original transaction into a string of 1's and 0's. To illustrate this process transaction 1 is used again.

Table 3. Extraction of the Business Rules

	Function	Field Name	Value	And / Or	Function	Field Name	Value	True Val.	False Val.
1		UserType	=1					1	0
2	Weekday	TDate	=1	or	weekday	TDate	=7	1	0
3		TranType	=1					1	0
4		Successful	=false					1	0
5		TNum	>= 170000	or		TNum	<= 80000	1	0

Table 4. Extraction of the Context Rules

	Field Name	Value	No of following transactions to check	True Value	False Value
1	TranType	11111	5	1	0
2	TranType	00101	5	0	1
3	Succesful	falsefalsefalsefalsefalse	5	1	0

2.2.1 Transaction 1: An example of a write transaction

2.2.1.1 Business Rules

- ξ The user type is 1, which indicates that this is an external user. There is a possibility that a hacker could have masquerade the identity of a legitimate user. A high- risk value is assigned to this field. The first value for this transaction is a 1.
- ξ This transaction was a write transaction. Since this means data have been changed or added, a high-risk value is assigned to this field. The second value for this transaction is a 1.
- ξ This transaction occurred after hours. Since this is very unlikely in a sales environment, a high-risk value is assigned to this field. The third value is a 1.
- ξ This transaction occurred on a normal weekday. The fourth value for this transaction is a 0.
- ξ This transaction was successful. The fifth value is a 0.

Table 5 shows the converted values for transaction 1 based on the business rules.

Table 5. Transaction 1 converted by applying the business rules

1	1	1	0	0

2.2.1.2 Context Rules

- ξ This transaction is the first of five successive write transactions. In the sales environment there is no reason to execute so many write actions successively, which indicates a possible risk. The first value is a 1.
- ξ The correct context for a normal sales transaction is '00101', which indicates 'read, read, write, read, write'. In a sales transaction the account number is read first, after which other client information is read. The items bought are then entered and the balance is read. Finally the sales clerk's personnel number is entered. This transaction does not occur in the correct context. The second value is a 1.
- ξ The current transaction as well as the successive three was successful. The third value is a 0.

Table 6 shows the converted values for transaction 1 based on applying the context rules.

Table 6. Transaction 1 converted by applying the context rules

1	1	0

Table 7 shows the conversion for Transaction 1. The converted values are obtained by joining the values of table 5 and table 6. These values are the eight inputs for the Neural Network Module.

Table 7. Converted transaction 1

1	1	1	0	0	1	0	1

2.3 Neural Network Module

A neural network is known for its ability to learn, generalise, and categorise data. The MASS prototype employs a neural network allowing the system to "learn" from the experience of past audits. After a few iterations of processing inputs, the neural network will recognise frequent inputs and deliver the known output. Inputs similar to known inputs will be categorise as the same input, and handled in the same manner. This means the neural network will recognise certain risks from past audits, but even new similar risks will be picked up and categorised as one of the known risks [FAUS94].

The **Neural Network Module** receives the eight inputs required from the knowledge-based expert system, and processes these through the network. The five outputs of the neural network are then used to visualise the transaction's risk factors. The risk value applicable to the individual transaction is responsible for the first three outputs, which determines the pattern of the bubble. The risk determined by the context of the transaction

is responsible for the last two outputs, which determines the size of the bubble. Figure 3 and 4 illustrates the visualisation process for the outputs.

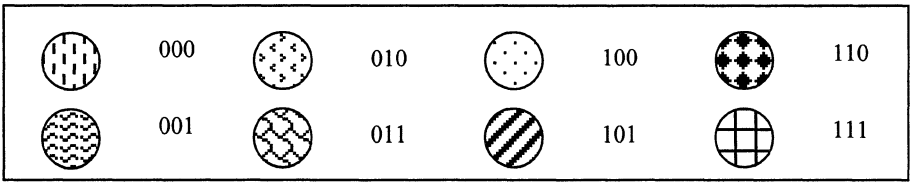

Figure 2. Illustrates the visualisation process for the first three outputs

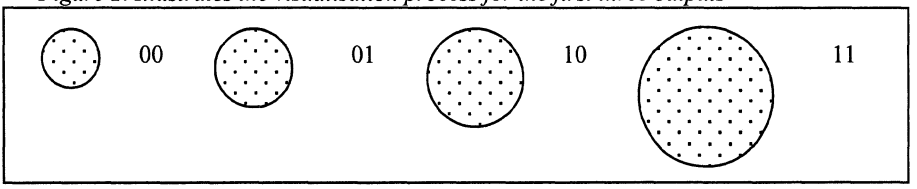

Figure 3. Illustrates the visualisation process for the last two outputs

Figure 5 depicts the configuration of the neural network. To illustrate this process, transaction 1 is used again.

2.3.1 Transaction 1: An example of a write transaction

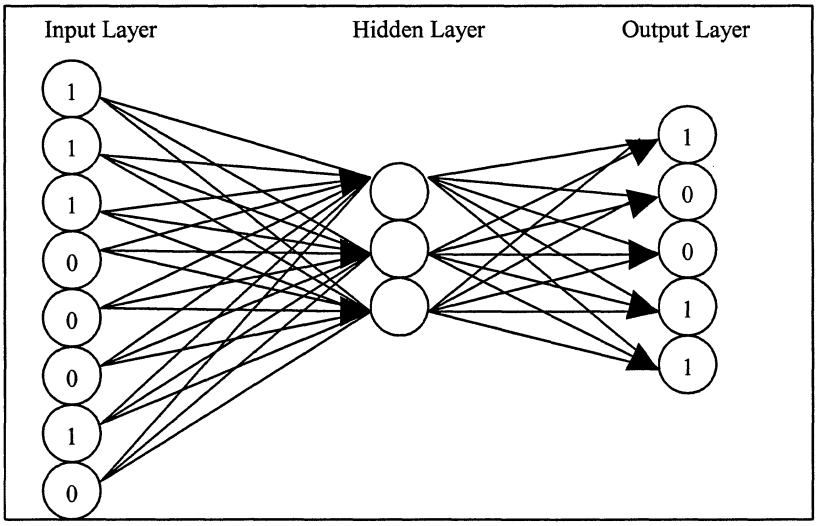

Figure 4. Neural Network Configuration, with values for transaction 1

Every node in the neural network has an associated weight. Since not all inputs have the same importance, different weights are assigned to each input value. The first input to the neural network is a '1'. A weight of '2' is assigned to the input. The weight is multiplied with the input to produce a value of '2'. The value of '2' is forwarded to the nodes in the hidden layer

for input 1. The second input is also a '1'. A weight of '4' is assigned to this input node. The weight is multiplied with the input and a value of '4' is sent to the nodes in the hidden layer for input 2. This process is followed for each of the eight inputs.

Each node in the hidden and output layer has an activation function. This function is responsible for processing all the inputs to the node to produce a single output. The activation function used in the model is the summation of all inputs to that node.

In the hidden layer the activation function processes the inputs to each node to produce a single value. If the value is above the activation value for each node, a value of '1' is sent out to the nodes in the output layer. If the value is below the activation value, a value of '0' is sent out to the nodes in the output layer. The first node in the hidden layer receives a value of '2' from the first input node and a value of '4' from the second input node. Similar inputs are received from all the other nodes in the input layer. All the inputs received ads up to 9. Since the activation value for this node is set to 5, a value of '1' is sent out to each of the nodes in the output layer. This process is followed for each node in the hidden layer.

The nodes in the output layer use the same process, to determine the final output. The weight assigned to the first node in the hidden layer is a '3'. The output value for the node, which is '1', is multiplied with the weight to produce a value of '3'. This value is sent to all the nodes in the output layer.

The first node in the output layer receives a value of '3' from the first node in the hidden layer, as well as other inputs from the second and third nodes. The input adds up to '5'. Since the activation value for the first output node is '1', a value of '1' is given as the first output. This process is followed for each node in the output layer.

Table 8. Output for transaction 1

1	0	0	1	1

Table 8 shows the output the Neural Network delivered for transaction 1 after processing. These values are the five inputs for the Visualisation Module.

2.4 Visualisation Module

By providing a graphical representation of the risk factor for each transaction from the Audit Log File, a manager can obtain an overview of risks by just scanning the graph. Certain areas, which depict a high risk, can then be investigated.

A single bubble filled with blocks or diamonds depicts a lower risk than a group of bubbles together filled with the same patterns. Instead of using

experts to evaluate every transaction, only the necessary ones need to be looked at. Figure 6 shows an extract of the output generated by the MASS prototype for the sales audit log file.

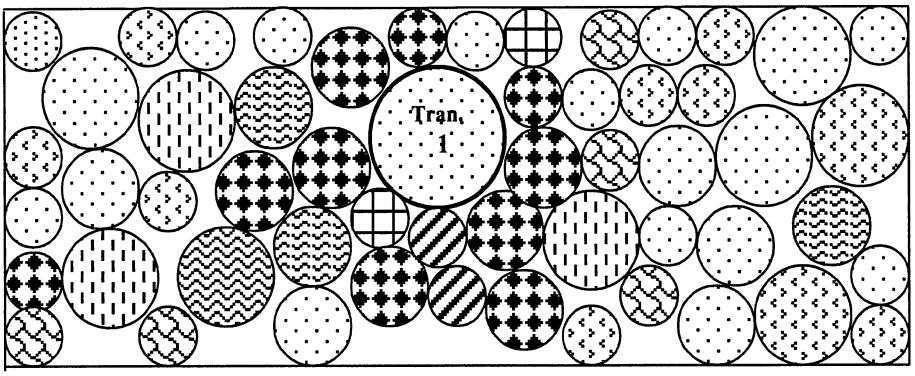

Figure 6. An extraction of the output generated by the system

To illustrate this process, transaction 1 is used again.

2.4.1 Transaction 1: An example of a write transaction

Figure 6 shows transaction 1 displayed as a big bubble filled with dots. The risks connected to the individual transaction determine the pattern of the bubble displayed. Transaction 1 was executed after hours by an external user and was a write transaction. All these risks form part of the calculated risk factor. The risks connected to the context the transaction is found in, determines the size of the bubble. Transaction 1 was the first of five successive write transactions, which fit in with the end result. The bubble depicting the risk factors for transaction 1 is in a group of bubbles filled with blocks and diamonds. This area prompts for further investigation. Figure 3 and 4 illustrate the transformation process for the outputs from the neural network. A bigger size bubble shows a higher risk where the context is concerned, while a fill pattern of blocks or diamonds depict a higher risk for the individual transaction.

3. CASE STUDY

The Audit Log File used for the case study is from an online sales environment with 200 branches. The average number of transactions per day for each of the branches is 1500, totalling 300 000 transactions daily. Each of these transactions is logged together with a user ID and reports identifying unsuccessful transactions are printed. Expert personnel then perform random checks on these reports to detect any fraudulent transactions. At the moment

the average number of fraudulent transactions per day picked up is about 10. By using the **MASS model** an instant view can be obtained of all the risks, and only the areas of interest need to be investigated. This not only overcomes the problem of large volumes of data, but also increases the efficiency and accuracy of identifying risks.

4. CONCLUSION

In practice, auditors make judgements by using their expertise and experiences while auditing transactions. However, the lack of systematic and consistent criteria for risk assessment can prevent auditors from determining the real risk factors in an effective manner. An expert system can provide explanation of results, while a neural network is known for its ability to learn, generalise, and categorise data. By applying these two technologies, a prototype such as MASS can help auditors perform risk assessment tasks more systematically and consistently.

5. REFERENCES

[YELR95] Ye,LR, The Value of Explanation in Expert Systems for Auditing: An experimental Investigation, Expert Systems with Applications, Vol. 9, No. 4, pp. 543-556, 1995.

[ABDO91] Abdolmohammadi,M.,J., Identification of tasks for Expert Systems development in Auditing, Expert Systems with Applications, Vol. 3, pp. 99-107, 1991

[CHIU94] Chiu, Chi-Tien , Scott, Robert, An intelligent Forecasting Support Systems in Auditing: Expert System and Neural Network approach, Proceedings of the twenty-seventh annual Hawaii International Conference on Systems Sciences

[FAUS94] Faussett, Laurene, Fundamentals of Neural Networks, Architectures, Algorithms, and Applications,1994