ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20190710100638
Original scientific paper

Classifying Database Users for Intrusion Prediction and Detection in Data Security

Cigdem BAKIR*, Veli HAKKOYMAZ

Abstract: The fact that users and applications acquire information using web sites on the internet means that document and information sharing, banking and other operational processes are increasing day by day. Recently however, with the widespread use of the internet, some security problems, such as unauthorized access, data breaches, code infection, malware infections, data leaks and distributed denial of service attacks have emerged. This situation necessitates the protection of the information used in personal and public spaces. In this study, a common model was created to detect user intrusions by taking into account criteria such as the number of transactions performed, their IP addresses, the amount of data they use, the transaction type they perform and the roles they undertake. In this way, the aim was to ensure database security by detecting risky user groups in advance.

Keywords: database security; intrusion detection systems; intrusion prediction systems; log records; risk analysis

1 INTRODUCTION

With the rapid development of technology, the internet has become an indispensable requirement for personal purposes and in many other areas such as banking, information sharing, e-commerce and communications. However, many problems have emerged using the internet, such as information leakage, seizure of information by unauthorized persons, alteration of information and failure to provide information confidentiality; many studies have been carried out to solve these problems [1-4]. These studies are usually related to the actions to be taken after the attack. In this study, however, the aim was to predict possible user attacks by analyzing log records beforehand and preventing any possible attacks that might create risk in the future, before they occur. In other words, users who threaten the system were identified and a risk analysis was performed. In this study, the words "attack" and "intrusion" are used interchangeably.Since there is no guarantee against being attacked (the inevitability of information/resource sharing and the reality that we do not live in an entirely lawful world), the problem of database security needs to be addressed. Data/information security problems in database systems can be addressed in a number of phases.Before the information security problem occurs the aim is to predict a potential attacker (application/user) before the attack occurs by looking at some behaviour, such as the various types of access to the system or processes, and to ensure that the system administrator and the responsible individuals are alerted to the possibility of such a potential attack.

After an information security problem occurs-when no advance precautions have been taken-the problem has then to been identified. It includes detecting whether an attack or information security problem has occurred once an issue emerges, and includes the operations performed to eliminate any problem that is subsequently identified.

If a system does not guarantee that there will not be any information security problems, and does not offer an intrusion detection/recovery mechanism, an attack to the system may occur, but users cannot distinguish what it is.

In this case, the insecure system will continue to be used by the users and unwanted situations will occur. Although this may seem an unacceptable situation, it is tolerable if the likelihood of an attack is very low and the system does not contain critical information. However, in the opposite case, it is necessary to assess measures that guarantee information security and eliminate the potential for attack, regardless of the cost. This study should be evaluated against the scope of measures to be taken before the emergence of an information security problem.

Rules were created using a variety of criteria that can model user behaviour. Unlike other studies, a risk analysis was carried out against information leakages. In addition, according to the model created, and regarding their risk status, the users were clustered into five groups: the riskiest, risky, medium risk, low risk and lowest risk. During the first phase of the study, the unregulated and scattered log records were collected, and the areas that were unnecessary and where transactions cannot be performed, were cleaned by filtering [5]. Thus, the resulting log records were analyzed and the data they contained was transformed into meaningful information. In the second phase, various rules were created to follow user practises and operations by transferring normalized data to the database. In the third phase, a common model was developed to identify user attacks by using the rules created. In the final phase, depending on the risk ratio calculated by the model, users were grouped according to their risk status and transferred to the database. In addition, users whose risk group was at the highest level were listed along with their IP addresses. A number of studies have been carried out on intrusion detection and prevention systems in the literature [6-8]. However, no model has been developed in these studies that can identify attacker users on the real data set. The work that is done is usually related to the operations that will be done after the attack. In our study, we developed an attack prediction system by classifying users using certain criteria in log records. In our study, risk users who attack the system were determined in advance for the data set we used. We calculated the success and accuracy of the proposed model by comparing the risky users we classified with the pre-determined risky users. We also compared the model we proposed [1, 9, 10, 11] with various studies used in the literature and presented their success in risk analysis. However, in our study, we did not use all the criteria in the log records. We can use all the criteria to develop a more advantageous and more successful attack prediction system.

The remainder of the article is organized as follows: Related and similar studies will be discussed in section 2, while log records and information security will be explained in section 3.The conducted study and design will be explained in section 4, and in the final concluding part (5), the contributions of the study will be summarized and the direction of future studies will be discussed.

2 RELATED WORKS

In this study, the aim was to predict database security problems (especially attacks that could be performed by users defined in the system). In order for a risk-based classification of system users, the risk ratios for the users were calculated by developing a common model. Situations investigated by these type of measurements included malicious software such as viruses, worms, Trojan horses and spyware; intrusion into the system using fake IP addresses; port browser attacks and users attempting incorrect or unauthorized operations that affect the system adversely by performing read and write actions on data that negatively affect database security [12, 13].

Database intrusion detection systems are divided into two types: signature-based and non-signature-based systems. The signature-based systems benefit from a database where the digital signatures of previous attacks are recorded [14]. These attacks are usually perpetrated through malicious software like viruses, worms, Trojan horses and spyware [11]. The disadvantage of this type is that it is costly to keep the signature information up to date, that is, it is not able to detect the signatures of new types of attack until they have already happened.

The non-signature-based intrusion detection systems are used to detect abnormal situations occurring within the system. They are used to detect internal attacks involving user access and role-based access. Most attacks occur when some users access the information they want by using hardware and software vulnerabilities. Generally, it takes the form of SQL injection attacks [14].

Quickprop neural networks, which are a multi-factor statistical prediction system, have been developed to predict database intrusions and detect security vulnerabilities before an attack occurs [13]. In this study, the Pearson correlation coefficient was used to calculate any hidden layers and reveal the users who did not have authorization to access a databank. Abnormal and incorrect user behaviour emerging in the short term was found. However, since only log records for one day were used, potentially risky users and uncontrolled user transactions occurring in the long term were not fully identified. In other words, it did not include a risk analysis for attacks that were carried out over the longer term. Another study that was carried out to identify incorrect user behaviour used a genetic algorithm [15]. The genetic algorithm performed a classification using neural networks with rules obtained by observing the characteristics of the network. The results of this study were compared to other studies. However, as in the former case, it also only suggested a solution for attacks over the short term. Another related study attempted to predict and prevent intrusions by using the Hidden Markov model [16]. The Hidden Markov model is a classification algorithm that is performed to find hidden states depending on given situations. In very large networks, data communications is distributed across the whole network and is therefore vulnerable to serious attacks. In this study, a risk analysis was carried out using a fuzzy logic technique in an attempt to identify dangerously high, outgoing packet rates. In addition, it attempted to determine attacks that might pose a risk in distributed environments.

In the first phase of the intrusion detection system, which uses a role-based access control model, the database log is examined. A classification model (Naive Bayes) is created based on past transactions and the roles in which they are used. User transactions are classified according to this model. The roles found are compared to the users in the database log. Accordingly, if the role is associated with the related user, then it is considered okay, if not, an alarm is issued. However, this study only detects attacks according to the roles of users and does not take into account the individual actions performed by the users [1].

Detection of Misuse in Database Systems (DEMIDS) is an example of an intrusion detection system used to detect attacks caused by user access to a database. This was used in a study to detect internal attacks (misconduct) for relational databases. This system consists of four main components: the controller, the data processor, the profile editor and the sensor. The controller collects the data and logs it in the audit log. The data processor converts the data to the desired structures and types. The profile editor "learns" a profile for each user. During the control phase, whether or not the user activities are suspicious is calculated by comparing them to the frequently used user profiles. Unlike the previous study, it defined a profile not just for user transactions in the system, but also for a user who is not in the database [2]. Database intrusion detection systems use a variety of data mining techniques to analyze the read/write dependencies between data objects [3]. First, the database transactions (in logs) that did not suffer any attack are analyzed and the rules that express the read/write dependencies are determined. These rules form a model learnt from the system. Then the attacks are detected by looking at whether new emerging transactions are in compliance with these rules. This study was not conducted with real data, and a large database with synthetic data was created for reading and writing operations only.

In real-time database systems, there are certain time constraints (deadlines) for transactions. These databases are designed for time-semantic data objects, which change over time and are periodically updated. The transactions are defined within certain time limits. In a study conducted by Lee for real-time database systems, time signatures of transactions were used to update the time-semantic data objects [4]. By contrast, real-time systems only give security warnings for write transactions. In this study, with the implemented model, users were classified according to their risk status regardless of whether or not there were attacks due to user access. All user transactions in the system were taken into account to try and identify risky users. Thus, an attempt was made to develop a database intrusion prediction system.

3 LOG RECORDS FOR INFORMATION SECURITY

Information security involves the protection of information from being used and modified by unauthorized people. It consists of three basic elements: integrity, privacy and accessibility. Integrity means the protection of information from random changes made by unauthorized people; privacy means that unauthorized people are not allowed to access information, and accessibility means that information is accessible only to authorized people [17, 10]. The risks that threaten the system are determined by information security. Transaction continuity is increased by ensuring confidentiality, accessibility and integrity [18]. Today, due to the widespread use of information technologies, especially when transferring applications to the internet, the sharing of lots of information in the internet environment and because almost all transactions are made over the internet, it is increasingly possible for malicious or unauthorized people to damage the system integrity [19, 20]. Log management in the database makes it possible to monitor the changes in information, as well as constituting the most basic structure for information security. To avoid information security violations, continuous log management is required. With log management, all users in the system, transactions, time of transaction, IP addresses, and their success or failure status are monitored, and when a dangerous situation is encountered, its reason is determined by looking at the log records. Log management is of great importance in terms of information security and the elimination of risks [21].

Misuse and attacks on the database are detected by examining the log records. For this, database objects and transactions involving those objects are evaluated. Specifically, access transactions to the database system objects must be monitored. In particular, a transaction that attempts to make changes on a table designed for read-only can be seen as harmful, and its detection is important.

Log management can be performed easily by keeping the log data collected from different sources in a single center [22]. With log management, abnormal situations can be detected in the system, and awareness can be created about future potential security problems for the personnel working in an institution. International standards, such as Cobit and ISO27001, support log management [23]. Log records are collected and stored in a center, and they are accessed quickly when requested. In addition, log records can be restored at any time. To ensure data integrity, consistency and data confidentiality, access permission is determined for data-related users, which is based on the level of relationships. Ready-made software, such as OSSIM, ManageEngine EventLog Analyzer, Swatch and File System Auditor, are free open source software packages used to perform log analysis [23].

4 MATERIALS AND METHODS

Detection of users who may pose a threat to data security by analyzing log records involves multiple steps. The general steps of the study are shown in Fig. 1.

4.1 Collecting and Normalizing the Log Records

In the case study, a log database was created of 4756 users and 108724 transactions entered into the system within 2 years, based on real data received from the bank. However, the collected log records (data) were complex and irregular. In order to make the collected log records meaningful and to achieve the intended goal of the study, the data was normalized and the necessary preliminary steps were taken by using various web mining and data mining techniques [21].

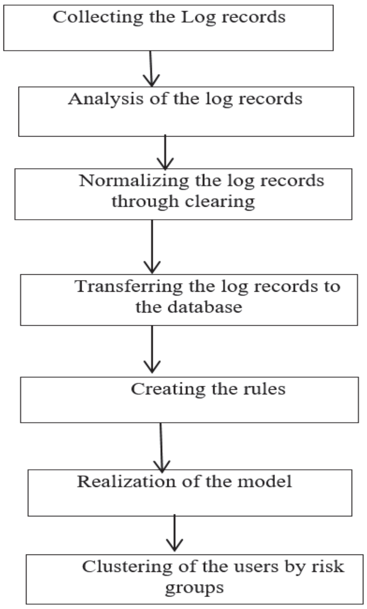
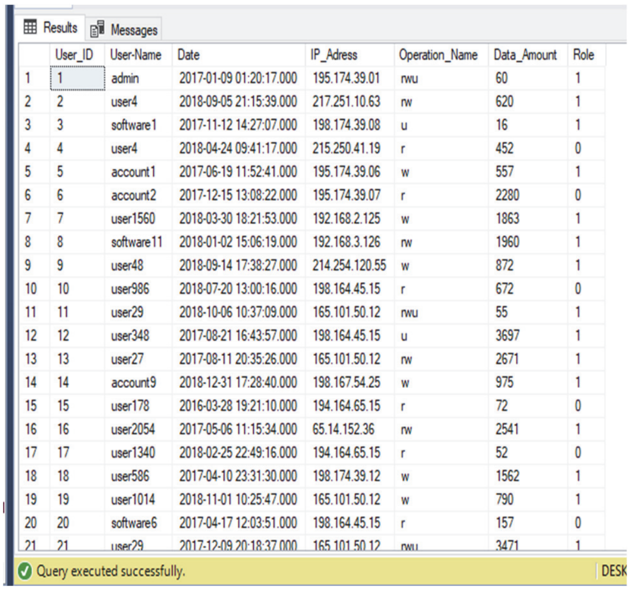

Collecting the Log records
Analysis of the log records
Normalizing the log records through clearing
Transferring the log records to the database
Creating the rules
Realization of the model
Clustering of the users by risk groups

Figure 1 Steps for determining the risk groups via log records

The data to be normalized consisted of information contained in log records, data collected by the server, users, and transactions performed by the users. After all unnecessary and unusable data was cleared from the log records, the data was organized by determining those users performing similar transactions, the transactions themselves, the number of transactions and the amounts of data they used. Normalization is the process of data cleaning, filtering, reporting and indexing to create meaningful information from the data in the log file. In other words, in order to ensure the use of log records in the desired format, the normalization process was performed to remove unnecessary fields. Fig. 2 shows the format of some of the log records that have been normalized.



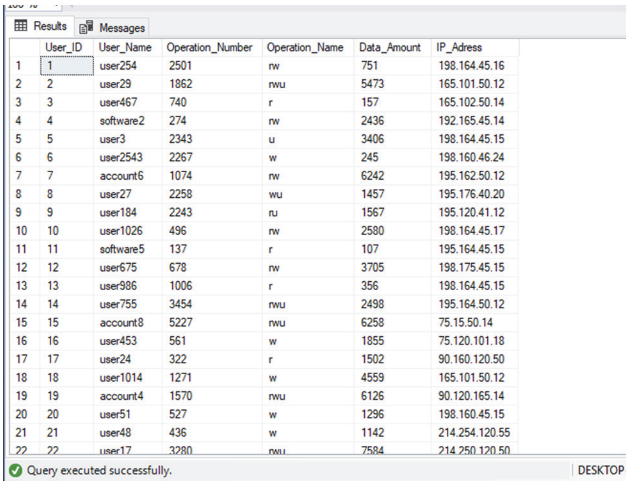
	User_ID	User-Name	Date	IP_Adress	Operation_Name	Data_Amount	Role
1	1	admin	2017-01-09 01:20:17.000	195.174.39.01	rwu	60	1
2	2	user4	2018-09-05 21:15:39.000	217.251.10.63	rw	620	1
3	3	software1	2017-11-12 14:27:07.000	198.174.39.08	u	16	1
4	4	user4	2018-04-24 09:41:17.000	215.250.41.19	r	452	0
5	5	account1	2017-06-19 11:52:41.000	195.174.39.06	w	557	1
6	6	account2	2017-12-15 13:08:22.000	195.174.39.07	r	2280	0
7	7	user1560	2018-03-30 18:21:53.000	192.168.2.125	w	1863	1
8	8	software11	2018-01-02 15:06:19.000	192.168.3.126	rw	1960	1
9	9	user48	2018-09-14 17:38:27.000	214.254.120.55	w	872	1
10	10	user986	2018-07-20 13:00:16.000	198.164.45.15	r	672	0
11	11	user29	2018-10-06 10:37:09.000	165.101.50.12	rwu	55	1
12	12	user348	2017-08-21 16:43:57.000	198.164.45.15	u	3697	1
13	13	user27	2017-08-11 20:35:26.000	165.101.50.12	rw	2671	1
14	14	account9	2018-12-31 17:28:40.000	198.167.54.25	w	975	1
15	15	user178	2016-03-28 19:21:10.000	194.164.65.15	r	72	0
16	16	user2054	2017-05-06 11:15:34.000	65.14.152.36	rw	2541	1
17	17	user1340	2018-02-25 22:49:16.000	194.164.65.15	r	52	0
18	18	user586	2017-04-10 23:31:30.000	198.174.39.12	w	1562	1
19	19	user1014	2018-11-01 10:25:47.000	165.101.50.12	w	790	1
20	20	software6	2017-04-17 12:03:51.000	198.164.45.15	r	157	0
21	21	user29	2017-12-09 20:18:37.000	165.101.50.12	rwu	3471	1

Query executed successfully.

DESK

Figure 2 Normalized log records

4.2 Transferring Log Records to the Database



	User_ID	User_Name	Operation_Number	Operation_Name	Data_Amount	IP_Adress
1	1	user254	2501	rw	751	198.164.45.16
2	2	user29	1862	rwu	5473	165.101.50.12
3	3	user467	740	r	157	165.102.50.14
4	4	software2	274	rw	2436	192.165.45.14
5	5	user3	2343	u	3406	198.164.45.15
6	6	user2543	2267	w	245	198.160.46.24
7	7	account6	1074	rw	6242	195.162.50.12
8	8	user27	2258	wu	1457	195.176.40.20
9	9	user184	2243	ru	1567	195.120.41.12
10	10	user1026	496	rw	2580	198.164.45.17
11	11	software5	137	r	107	195.164.45.15
12	12	user675	678	rw	3705	198.175.45.15
13	13	user986	1006	r	356	198.164.45.15
14	14	user755	3454	rwu	2498	195.164.50.12
15	15	account8	5227	rwu	6258	75.15.50.14
16	16	user453	561	w	1855	75.120.101.18
17	17	user24	322	r	1502	90.160.120.50
18	18	user1014	1271	w	4559	165.101.50.12
19	19	account4	1570	rwu	6126	90.120.165.14
20	20	user51	527	w	1296	198.160.45.15
21	21	user48	436	w	1142	214.254.120.55
22	22	user17	3280	rwu	7584	214.250.120.50

Query executed successfully. DESKTOP

Figure 3 Transferring the criteria used in the log records to the database

The partial log records are shown in Fig. 2, and the normalized versions are shown in Fig. 3 in a way to show the amount of transactions used-created according to the criteria in the model, the type of transaction, the amount of data used, and the IP addresses.

4.3 Implementation of the Model

In intrusion prediction and detection systems, selection of features is very important in creating the model [24]. In this study, the system use, the amount of data used, the user permissions and IP addresses, the user roles, and who managed and operated the system, are all defined as the basic criteria for creating the model. In other studies, model implementation was achieved using one or two criteria [12, 24]. By contrast in this study, the risk situations associated with malicious users who could threaten the security of the system were analyzed by selecting more than one criterion and the most commonly used criteria. In addition, the aim was to implement a common model that covered all of the system users. System usage and the time spent on the system allow the various attacks to be distinguished. The permissions granted to the user are the main indicators in determining unauthorized transactions in the database, or whether the users who are not authorized have entered the system [14]. It reflects whether the users are authorized or not for read/write operations. In addition, port number, and source and destination IP address information, are used to group SQL commands. Database communication is performed on a different port. The port number can be used to edit packets in a session, to check whether packets are error free, and to check whether the data is correctly sent to the server. IP addresses and the amount of data are the criteria chosen to create a model in order to detect entry to the system with a fake IP address and detect port scanner attacks [12, 13].

4.4 Creating Rules

In our study, some criteria were determined by taking into account the users' usage and operations in the system, and rules were developed to be able to determine the risk ratio according to these criteria (see Fig. 3). The rules for calculating the risk ratio were determined as shown below.

4.4.1 System Usage

The criteria, such as the time spent in the system, the number of entries in the system, and how often the user enters the system, indicate how much the user is engaged with the system [25]. As the user's time spent in the system increases, the probability of performing transactions, such as receiving information about the system, changing information, or adding information increases. For this reason, the system usage status for the user was used as a criterion in this study. According to the frequency of system entry, the following rules were determined and the weights of those who enter the system were as follows:

- 1 to 10 times was classed as 0,
- 11 to 20 times was classed as 1,
- 21 to 50 times was classed as 2,
- 51 to 100 times was classed as 3,
- 101 to 250 times was classed as 4,
- at least 251 times was classed as 5.

The rules were created in this way to distribute all sample log records evenly when ranked from large to small, according to the time the user spent in the sample database.

4.4.2 The Amount of Data Used

The amount of data a user uses in the system is important for performing risk analysis. This is because as the amount of data used increases, users can obtain more information on the system and find the opportunity to change it by using more information. According to the data used in the system, the user weights were as follows:

- 0 to 50 KB was classed as 0,
- 51 to 100 KB was classed as 1,
- 101 to 150 KB was classed as 2,
- 151 to 500 KB was classed as 3,
- at least 501 KB was classed as 4.

4.4.3 User Permissions

In a system, user permissions are determined for read, write and update operations [5]. The fact that a user who does not have permission for reading and writing in the system, reads data containing confidential information, makes changes to data and overwrites the data, carries a risk in terms of data confidentiality, data integrity and data reliability. According to the user's permission in the system, the weights of those who perform operations are as follows:

- the read operation was classed as 1,
- the write operation was classed as 2,
- the update operation was classed as 3.

4.4.4 IP Address Frequency

The most commonly used IP addresses were considered risky in terms of database security. The most frequent 100 IP addresses used in the system were determined.

- The weight for users who connected from IP addresses that were not used frequently was classed as 0.

- The weight for users who connected from the most frequent 100 IP addresses that were used frequently was classed as 1.

4.4.5 Roles Undertaken by Users who Managed and Implemented the System

For users who implemented the system, displaying the transactions of other users outside their authority carries risks in terms of data confidentiality, data integrity and data security. A risk and confidence analysis was performed by looking at the roles that these users undertake [1]. According to the roles undertaken by the users who manage and implement the system:

- the weights of those who only display the transactions were classed as 0.
- the weights of those who display the transactions and perform writing were classed as 1.

4.5 Clustering Users

After the rules were created, Eq. (1) was used to calculate the risk ratio:

$$Risk\ Ratio = \sum_{i=1}^{n} \left(w_{a,i}, w_{b,i}, w_{c,i}, w_{d,i}, w_{e,i}, \right) \Big/ T \tag{1}$$

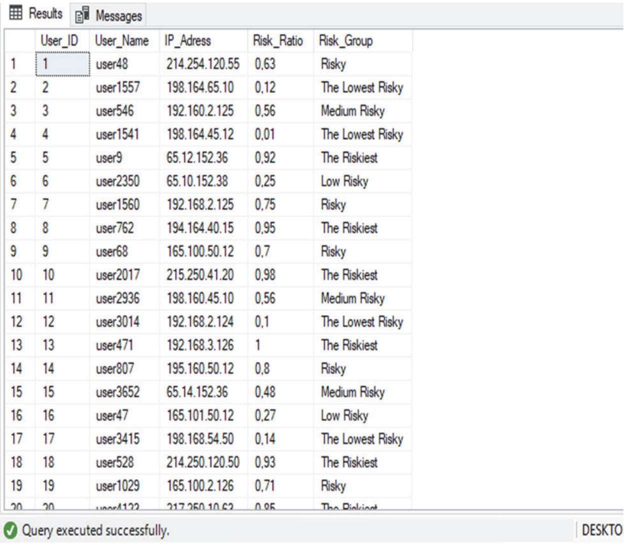
In this equation, $w_{a,i}$ refers to the weight of the "a" feature (number of entries into the system), $w_{b,i}$ refers to the weight of the "b" feature (amount of data used), $w_{c,i}$ refers to the weight of the "c" feature (permission given to the users), $w_{d,i}$ refers to the weight of the "d" feature (IP address) and $w_{e,i}$ refers to the weight of the "e" feature (roles of the users who manage and implement the system). While n represents the number of users in the system, T refers to the total weight of the strongest weights for each criterion according to the rules. However, since a user can be granted permission to read, write and update, when considering the user permission criterion, all the weights of this criterion were transferred to the T variable when calculating the risk ratio. When the above formula is examined carefully, it can be observed that the risk ratio will take values between 0 and 1. If we carry out risk classification according to this, risks are clustered under five groups that threaten the system, as follows:

$$Risk\ Ratio = \begin{array}{c} \text{0.00 to 0.2 is the lowest risky} \\ \text{0.21 to 0.4 is low risky} \\ \text{0.41 to 0.6 is medium risky} \\ \text{0.61 to 0.8 is risky} \\ \text{0.81 to 1.0 is the riskiest} \end{array}$$

Tab. 1 shows the risk ratios and risk groups of some users according to the developed model. In addition, this information was recorded in the database for use by the relevant staff in order to facilitate the determination of risky users (see Fig. 4).

Table 1 Risk Ratio and Risk Groups of Some Users in the System

ID	User	Risk Ratio	Risk Group
1	user1264	1.00	The riskiest
2	user10	0.75	Risky
3	user124	0.25	Low risky
4	user26	0.18	The lowest risky
5	user2350	0.68	Risky
6	user1557	0.12	The lowest risky
7	user65	0.50	Medium risky
8	user546	0.56	Medium risky
9	user42	0.37	Low risky
10	user9	0.06	The lowest risky



	User_ID	User_Name	IP_Adress	Risk_Ratio	Risk_Group
1	1	user48	214.254.120.55	0,63	Risky
2	2	user1557	198.164.65.10	0,12	The Lowest Risky
3	3	user546	192.160.2.125	0,56	Medium Risky
4	4	user1541	198.164.45.12	0,01	The Lowest Risky
5	5	user9	65.12.152.36	0,92	The Riskiest
6	6	user2350	65.10.152.38	0,25	Low Risky
7	7	user1560	192.168.2.125	0,75	Risky
8	8	user762	194.164.40.15	0,95	The Riskiest
9	9	user68	165.100.50.12	0,7	Risky
10	10	user2017	215.250.41.20	0,98	The Riskiest
11	11	user2936	198.160.45.10	0,56	Medium Risky
12	12	user3014	192.168.2.124	0,1	The Lowest Risky
13	13	user471	192.168.3.126	1	The Riskiest
14	14	user807	195.160.50.12	0,8	Risky
15	15	user3652	65.14.152.36	0,48	Medium Risky
16	16	user47	165.101.50.12	0,27	Low Risky
17	17	user3415	198.168.54.50	0,14	The Lowest Risky
18	18	user528	214.250.120.50	0,93	The Riskiest
19	19	user1029	165.100.2.126	0,71	Risky
20	20	user4122	217.250.10.62	0,85	The Riskiest

Query executed successfully.

Figure 4 Grouping users according to risk ratio

5 CONCLUSION

This study aimed to help reduce the vulnerabilities of database systems by conducting a risk analysis. The security analysis was carried out by identifying risky users and taking the necessary measures via recommendations to the employees using the system. According to the model developed using a variety of rules, the risk ratios of those threatening the system were calculated and their risk status was classified. By looking at the log records, and determining users and IP addresses that might create a possible risk group, they could be transferred to the database where the log file is kept. A clustering process was then performed according to the frequency of users entering the system, the amount of data they used, the permissions given to the users, and their IP addresses.

This study will be a guide for employees working on security issues in organizations. The aim was to prevent possible attack scenarios by identifying users in advance who could threaten the system. For future studies, it can be considered as a reference study, and this study is intended to be expanded using new criteria.

In addition, we aim to increase the success of our proposed model by using other criteria in the log records in the future.

6 REFERENCES

[1] Baihaqi, S., Dwiputra, P., & Seniman, F. (2018). Intrusion Prevention System Against Denial of Service Attacks Using Genetic Algorithm. *IEEE International Conference on Communication, Networks and Satellite*, 55-59. https://doi.org/10.1109/COMNETSAT.2018.8684039

[2] Wondimu, K., Zegeye, R., Dean, A., & Farzad, M. (2019). Multi-Layer Hidden Markov Model Based Intrusion Detection System. *Machine Learning Knowledge Extraction*, *1*(1), 265-286. https://doi.org/10.3390/make1010017

[3] Bertino, E. & Terzi, E. (2005). Intrusion detection in RBAC-administered databases. *21st Annual in Computer Security Application Conference*, 10-182. https://doi.org/10.1109/CSAC.2005.33

[4] Chung, C. & Gertz, Y. (2000). Demids:A missue detection system for database systems. *Integrity and Internal Control in Information Systems*, *37*, 159-178. https://doi.org/10.1007/978-0-387-35501-6_12

[5] Li, Q., Snadhu, R., Zhang, X., & Xu, M. (2017). Mandatory Content Access Control for Privacy Protectionin Information Centric Networks. *IEEE Transactions on Dependable and Secure Computing*, *14*(5), 494-506. https://doi.org/10.1109/TDSC.2015.2494049

[6] Swathy, M. & Padmavathi, G. (2018). Taxonomy of Security Attacks and Risk Assessment of Cloud Computing. *Advances in Big Data and Cloud Computing*, 37-59. https://doi.org/10.1007/978-981-13-1882-5_4

[7] Ray, I. & Belyaev, K. (2013). Secure logging as a service-delagating log management to cloud. *IEEE Journal Systems*, *7*(2), 323-334. https://doi.org/10.1109/JSYST.2012.2221958

[8] Kent, K. & Souppaya, M. (2006). Guide to Computer Security Log Management. *National Institute of Standarts and Technology*. https://doi.org/10.6028/NIST.SP.800-92

[9] Xingshuo, A., Jingtao, S., Xing, L., & Fuhong, L. (2018). Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system. *EURASIP Journal on Wireless Communications and Networking*, 1-9. https://doi.org/10.1186/s13638-018-1267-2

[10] Qi, L., Fang, W., Junfeng, W., & Weishi, L. (2019). LSTM-Based SQL Injection Detection Method for Intelligent Transportation System. *IEEE Transactions on Vehicular Technology*, *68*(5), 4182-4191. https://doi.org/10.1109/TVT.2019.2893675

[11] Ramasubramanian, P. & Kannan, A. (2014). Multi-Agent based Quickprop Neural Network Short-term Forecasting Framework for Database Intrusion Prediction System. *CiteSeerX.*

[12] Sahani, R., Shatabdinalini, C., & Rout, J. C. (2018). Classification of Intrusion Detection Using Data Mining Techniques. Progress in Computing. *Analytics and Networking,* 753-764. https://doi.org/10.1007/978-981-10-7871-2_72

[13] Geraldine, L., Gregory, E., Haider, A., & Carsten, M. (2018). Security and Privacy of Things: Regulatory Challenges and Gaps for the Secure Integration of Cyber-Physical Systems. *Third International Congress on Information and Communication Technology*, 1-12. https://doi.org/10.1007/978-981-13-1165-9_1

[14] Nabeela, A., Waqar, A., & Rehan, A. (2018). Comparative Study of Data Mining Algorithms for High Detection Rate in Intrusion Detection System. *Annals of Emerging Technologies in Computing (AETiC)*, *2*(1), 49-57. https://doi.org/10.33166/AETiC.2018.01.005

[15] Subbalakshmi, S. & Madhavi, K. (2018). Security challenges of Big Data storage in Cloud environment: A Survey. *International Journal of Applied Engineering Research*, *13*(17), 13237-13244.

[16] Reza, T., Satyajayant, M., Travis, M., & Gaurav, P. (2018). Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Communications Surveys & Tutorials*, *20*(1), 566-600. https://doi.org/10.1109/COMST.2017.2749508

[17] Zhang, Y. & Ye, X. (2009). A practical database intrusion detection system framework. *In Computer and Information Technology*, *1*, 342-347. https://doi.org/10.1109/CIT.2009.69

[18] Pfleeger, C. & Pfleeger, S. (2002). Security in Computing. *3rd Prentice Hall Professional Technical Reference*.

[19] Burow, N., Carr, S. A., Nash, J., Larsen, P., & Franz, M. (2017). Control-Flow Integrity; Preccision, Security and Performance. *ACM Computing Surveys*, *50*(1), 1-16. https://doi.org/10.1145/3054924

[20] Ramasubramanian, P. & Kannan, A. (2014). Multi-Agent based Quickprop Neural Network Short-term Forecasting Framework for Database Intrusion Prediction System, *CiteSeerX*.

[21] Rana, M., Kubbo, M., & Jayabalan, M. (2017). Privacy and Security Challenges towards Cloud Based Access Control in Electronic Health Records. *Asian Journal of Information Technologu*, *16*(2-5), 274-281.

[22] Chenglu, J., Saeed, V., & Marten, D. (2018). Snapshotter: Lightweight intrusion detection and prevention system for industrial control systems. *IEEE Industrial Cyber-Physical Systems (ICPS).*

[23] Bridges, S. M. & Vaughnn, R.B. (2000). Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection. *23rd National Information Systems Security Conference,* October, Baltimore, MD.

[24] Lee, V. C. & Stankovic, J. A. (2000). Intrusion detection in real-time database systems via time signatures. *Real-Time Technology and Applications Symposium (RTAS)*, 124-133. https://doi.org/10.1109/RTTAS.2000.852457

[25] Bakir, C. & Hakkoymaz, V. (2015). Veritabanı Güvenliğinde Saldırı Tahmin ve Tespit Sistemi için Kullanıcıların Sınıflandırılması. *8th International Conference on Information Security and Cryptology*, 28-33.

Contact information:

Cigdem BAKIR
(Corresponding author)
Yildiz Technical University,
Davutpasa Street, 34220, İstanbul, TURKEY
E-mail: cigdem.bakir@igdir.edu.tr

Veli HAKKOYMAZ
Yildiz Technical University,
Davutpasa Street, 34220, İstanbul, TURKEY
E-mail: veli@ce.yildiz.edu.tr