Int. j. inf. tecnol. (June 2021) 13(3):877–887
https://doi.org/10.1007/s41870-021-00659-1

ORIGINAL RESEARCH

Intrusion detection system for data warehouse with second level authentication

Amar Arora[1] · **Anjana Gosain**[2]

Received: 12 October 2020 / Accepted: 26 March 2021 / Published online: 17 April 2021

Abstract Data Warehouse (DW) security has always been a critical challenge for DW designers because of its global availability and accessibility. Over time, different researchers have suggested different DW security solutions, such as Role Based Access Controls (RBAC), Extended RBAC, Temporal RBAC (TRBAC), Risk-based access control, etc. Intrusion Detection System (IDS) and some other customized security solutions for DWs have also been proposed. Here, Risk-based access control provides additional security by utilizing risk value for each access decision. In RBAC systems, if an attacker obtains access to the system using some compromised credentials, the RBACs has no mechanism to secure DW elements which are accessible to the compromised user's role. The Intrusion Detection System (IDS) aims to solve this limitation; it monitors the user activities and alerts the system administrator whenever a user deviates from routine behavior. However, in the IDS solution for DWs, most of the real intrusions go undetected. In this work, we propose a second level authentication within the IDS, where a minute deviation from the user's past behavior is detected. It brings more robustness to the user's historical profile and makes the system less susceptible to false negatives. The proposed solution has been implemented on standard TPC-H databases, and results indicate a significant decrease in undetected real intrusions, which is one of the main achievements of the proposed mechanism.

✉ Amar Arora
amar.arora@nic.in

[1] Scientific Assistant-B, National Informatics Centre, MeitY, Govt. of India, Delhi, India

[2] USICT, Guru Gobind Singh Indraprastha University, Delhi, India

Keywords Data warehouse security · Second level authentication · Intrusion detection system

1 Introduction

Data Warehouse [1] is designed to generate business knowledge and store sensitive information about a business. Data Warehouse's online usability and global availability make them the big target of attackers [2]. Securing them from attacks or data leakage, therefore, becomes an essential challenge for DW's defense. Organizations introduced various security measures to protect their business secrets from those security challenges. Similarly, according to Data Warehouse's design requirements, various security measures have been implemented to safeguard DWs from these security threats [3–6].

Role-Based Access Control (RBAC) [7] has also been a measure for controlling access to DW entities with various users and associated privileges. However, RBAC's static nature and its unclear definition of groups and users make way for Extended RBAC [4]. Temporal RBAC (TRBAC) [8], an extension of RBAC, allows temporary limitations on the roles and summarized data [9]. Risk-based access control systems [10], another extension of RBAC, provides access control by combining static policies with dynamic and real-time features. If any user accesses DW with compromised user credentials in all the RBAC solutions, RBAC will not prevent the system's exploitation. Intrusion detection systems (IDSs) aim at detecting attacks on computer systems and networks [11] or information systems in general. Though it is challenging to provide secure information systems and maintain them in such a safe state for their entire lifetime, IDS helps to provide security to information systems to some extent [3]. IDSs detect

unauthorized access automatically as per their design structure [11]. They work mainly in two ways: "misuse detection" where user actions match well-known predefined patterns of attack; and "anomaly detection" where user actions are analyzed to detect deviations from a determined normal behavior [5].

Earlier IDSs have focused on intrusion at the network or operating system level. Due to its lack of knowledge of application-level semantics, these IDSs have not been proven effective in dealing with application-level attacks [5]. However, Database Intrusion Detection System (DIDS) is aimed at detecting attacks at the application level. DIDSs are often marred by low intrusion detection rates or large numbers of false alarm [12, 13]. While some DIDS methods have been suggested to minimize false positives [14, 15], according to Santos [5], they are not appropriate for heterogeneous environments such as DWs. Due to the uniqueness of certain data warehouse features, an intrusion detection mechanism tailored explicitly to the data warehouse was proposed [5]. All IDS solutions, including IDS for DW, have their agreed share of false negatives (FN), i.e., missing warnings on a real attempted intrusion [5, 16]. In this article, we have proposed IDS for DW with a second level authentication mechanism, resulting in the reduction of false negatives, i.e., undetected attempts at the true intrusion. Second level authentication triggers whenever a deviation in user behavior from its historically recorded behavior is detected. A correct response to the second level authentication challenge is mandatory; else, the current user's session will be terminated. The answer will be updated in the user's historical profile accordingly. It brings more robustness to the user's historical profile and makes the system less susceptible to false negatives. It also automates the IDS system and reduces dependence on the system administrator in case of IDS' intrusion alarm.

The main contributions of this paper are as follows:

- A user profile based on their access trends over time has been created. These user profiles act as the respective user's signature, which is used to identify them individually.
- A customized intrusion detection system has been developed, which includes second level authentication. It also includes prompting predefined secret questions or any other mechanism such as OTP (One-time password) to the user whenever the intrusion detection system produces an alarm.
- It also updates the user profile on positive second level verification with the current input sequence. It helps to build a robust user profile and reduces false negatives.

The rest of the article has been structured as follows: Sect. 2 discusses relevant DIDS research and related work. Section 3 discusses the proposed second level authentication by IDS and its user profile implications. Section 4 provides an experimental assessment and comparative analysis of the TPC-H decision support benchmark. Section 5 concludes the inferences of the outcomes and outlines possible future work.

2 Related work

Role-Based Access Control (RBAC) [7] has been considered one of the best ways of controlling access to DW entities with various users and associated privileges. Once all user functions are filled into the database, role-based rules are formulated, followed by workflow engine modules. Through these components, role-based privileges can be accessed and updated easily through multiple systems, networks, applications, and geographic locations. Here, companywide control over data and resources can be managed by RBAC [4].

While RBAC is commonly used and can handle the whole system, there are some problems, such as an unclear definition of groups and users and no mention of duties and responsibilities. An Extended RBAC for stable warehousing of data was introduced as a solution [4], a robust administrative and decision process with temporal dependencies, mutability, and identity management. Another extension of RBAC was introduced as Temporal RBAC (TRBAC) and its various variations [17–20], allowing for temporary limitations on the roles themselves, user-permission assignments (UA), permission-role assignments (PA), and role hierarchies (RH) [8]. Another role based access technique targeting summarize data [9] has also been proposed. It introduces restrictions on the summary of data as per the user's role. This summarized data based restriction will help in further improvement in security as more summarized data having more information as compared to less summarized data. All of the above research studies on RBAC ensure that users access only their authorized sections of the DW.

On the other hand, dynamic access control methods like Risk-based access control systems [10] employ static policies and dynamic and real-time features to make access decisions. These dynamic features can involve context, trust, history events, location, time, and a security risk [21], which brings flexibility to the access control mechanism. In all the access control mechanisms discussed above, if an attacker obtains access to the system using some compromised credentials, the access control has no mechanism to secure DW elements accessible within the compromised user's risk limits.

So, if an attacker with a compromised user credentials reaches DW, RBAC will not be able to restrict the system's

exploitation. Intrusion detection acts as a far better solution to overcome this limitation. There are various methodologies for handling intrusion detection for network and systems [11, 16, 22, 23], but they are not appropriate for database security [24, 25]. As a solution, some research analyzes intrusion detection in databases based on Role Based Access Control (RBAC) [24–28]. Some research focused on insider user behavior and looked at intrusion attacks from these users [29]. It presents a function-extraction method for modeling an internal user's access patterns. Here, an insider is someone with access, privilege, or knowledge of information systems and services [30]. Some research work suggests their targeted IDS keep their attention on specific types of databases [31, 32].

One of the recent studies [24] discussed the mechanism of intrusion detection in databases using data mining techniques such as clustering and classification. This article, unlike other works, discussed querying actions in detail rather than anomalies in data. It also proposed a novel method using clustering and classification for intrusion detection in databases. In another approach, the summarization of raw SQL queries into a compact data structure called hexplet for anomaly detection [28] was proposed. Here, hexplets are used for modeling normal access behavior and recognize intruders within RBAC systems. However, most of these articles restrict their consideration of intrusion detection for the RBAC in the database, and they work considering a role as the input for profiling purposes. Because of the uniqueness of certain data warehouse features, an intrusion detection mechanism tailored explicitly to the data warehouse was proposed by Santos [5]. Here, the intrusion detection solution for DW [5] built user profiles based on the various user's input queries and evaluated their results on the TPC-H decision support benchmark. Although it provides a targeted IDS solution for DW, it is also dependent on the system administrator in case of intrusion detection.

For security analysis of the above-discussed articles, we have divided the categories of the security mechanism discussed above into the following categories:

- *Access Control* Restriction on the user's access to the resources as per their role and associated privileges.
- *User Behavior Analysis* Access decisions based on analyses of user's access history and current user's request.
- *Access Decision Based on Risk Factors* Access decisions based on a computation of security risk on various risk factors like sensitivity of the information, history of access decision, etc.
- *Intrusion Detection for DW* Customized intrusion detection mechanism for catering of data warehouse's needs. It automatically detects unauthorized access by analyzing the user's current query profile and its historical access profile at the SQL command level.
- *Flexible Access Control in Minute Diversions* Allows another chance for the user to prove its identity whenever a user deviates minutely from its access history.
- *Access Profiling* The user or the user's role access history is used to creating access behavior. This behavior pattern acts as the benchmark for IDS to find any deviation.

The primary techniques for databases and data warehouses security have been discussed in Table 1 as per the security mechanism employed. It helps to identify any research gap in the data warehouse's security issues related to intrusion detection. We have also included Risk-based access controls in the analysis as they also look to have a similar working as the IDS solutions.

By analyzing Table 1, it can be an inference that Risk-based access control and IDS for DW are the categories that provide access controls based involving other factors like user behavior or risk factors. In both of these categories, the threshold values act as a benchmark for the risk factor or user access history. The user's current access parameters have to be below these thresholds to allow access to the required values. There is no provision for another authentication level in all the above-discussed articles if they deviate minutely from their normal behavior. In this work, we semi-automate the intrusion alert mechanism by introducing the second authentication level instead of rejecting the request altogether. It reduces the rate of false negatives and improves the user's profile robustness and semi-automation of the entire intrusion detection mechanism.

3 Attacks on data warehouse

The DW has to be always available to the outer world for reporting and analytics; thus, it is susceptible to security attacks. The attacks on data warehouse have been classified into the following three broad categories [5]:

1. Data Corruption: Intruder targets the integrity of the data warehouse by modifying the entries or complete deletion of table, view, or number of rows.
2. Information Stealing: Here, the intruder's target is to steal valuable information like trade secrets of future trends, etc. Here, the entries of the data warehouse are not damaged or modified.
3. Denial of Service: The focus of these attacks to make the data warehouse services unavailable by erasing the database object, flooding data warehouse services with

Table 1 Analysis of various security techniques related to IDS

Security method	Security method	Access control	User behavior analysis	Access decision on risk factors	Intrusion detection for DW	Flexible access control in minute diversions	Access profiling
Sandhu [7]	Role-based access control (RBAC)	Yes	No	No	No	No	No access profile
Thuraisingham and Iyer [4]	Extended RBAC	Yes	No	No	No	No	No access profile
Joshi et al. [17]	Generalized temporal RBAC	Yes	No	No	No	No	No access profile
Emre et al. [8]	Temporal RBAC security analysis	Yes	No	No	No	No	No access profile
Ali et al. [9]	Authorization model on summarized data	Yes	No	No	No	No	No access profile
Ramachandran et al. [24]	IDS in the relational database	Yes	Yes	No	No	No	Role-based
Ramachandran et al. [25]	Anomaly detection in the relational database	Yes	Yes	No	No	No	Role-based
Rao et al. [26]	Machine learning based IDS for relational database	Yes	Yes	No	No	No	Role-based
Darwish et al. [27]	Role administered based IDS	Yes	Yes	No	No	No	Role-based
Darwish [28]	Hexplet based database IDS	Yes	Yes	No	No	No	Role-based
Mathew et al. [29]	Insider attack detection in database	Yes	Yes	No	No	No	Role-based
Santos et al. [5]	IDS for DW	Yes	Yes	No	Yes	No	User action-based
Atlam et al. and Santos et al. [21, 33]	Risk-based access controls	Yes	Yes	Yes	No	No	Role-based risk profile

the unwanted request with a considerable workload, halting the data warehouse server instances, etc.

Given the classification of possible attacks on the DW, the RBAC and its variants, Risk-Based Access Controls and IDSs have been classified in Table 2.

In case a user's authentication with full access to the system has been compromised, then RBAC and its variants will have very little control over user behavior if this attacker tries to execute commands for data theft or planned data manipulation. To solve with problem in this work we have proposed the second authentication level to track the user's behavior and verify its historical action signatures. The proposed method prevents DW from data corruption and information stealing. The issues related to

Table 2 Classification of security techniques while attacks on DW

Security method	Data corruption	Information stealing	Denial of service
Role-based access controls [4, 16–20]	Yes	Yes	No
Authorization model on summarized data [9]	Yes	Yes	No
Risk-based access controls [21]	Yes	Yes	No
IDS solution for RBAC [24–28]	Yes	Yes	No
Insider attack detection in the database [29]	Yes	Yes	No
IDS for DW [5]	Yes	Yes	No

denial of service have not been part of the scope of this article.

4 Data warehouse intrusion detection system (DW-IDS)

As shown in Fig. 1, the DW-IDS works as an extension to existing DIDS. The user initially passes a standard authentication process by supplying their credentials in a username, password. Once authorized, every action performed by the user will be monitored, logged, and add to its user profile. The second authentication level will be activated whenever user actions deviate from its historical profile generated over time.

The two-level authentication IDS system allows the system to ask for second level authentication whenever user activity deviates from its regular user profile [34]. The detail working of the architecture has been explained below:

Whenever the user wants to access DW, they provide their user credentials and password to the DBMS. Once signed in successfully, the system executes user queries according to the user's need for meaningful results.

To build the current query profile based on the predefined parameters, the DBMS transfers query and its expected response from Enterprise DW to the command and response analyzer.

The UAPR includes parameters such as query execution time, response size, processed rows, and the number of columns in response, etc. to form the current query profile.

UAPR loads the user's access profile from BAR (Behavioral Analysis Repository), provided by the user during the training process and their historical access over time. UAPR sends both the current query profile and user's access profile to the threat analyzer for comparison and further action, if any.

Then, the threat analyzer compares both user and current query profiles within the security administrator's threshold. The security administrator uses a security manager interface to set various threshold parameters.

A threat analyzer sends its comparative analysis to the authorizer to decide further course of action.

Then authorizer chooses to allow results to be transferred to the user immediately or to initiate authentication of the second level before that. This decision has been made based on input from the threat analyzer and the security administrators' policy rules through the security manager interface. If second level authentication is initiated, the system asks a security question to reverify the user's identity. The security question gets randomly selected from the list of security questions. The user provides the answers to these security questions during its registration time.

If the user successfully answers it correctly, the authorizer informs DBMS about providing the user results. Otherwise, the user will be forced to end the session without any results. In positively verified cases, the query profile will be passed to UAPR in order to update the user's historical profile accordingly.

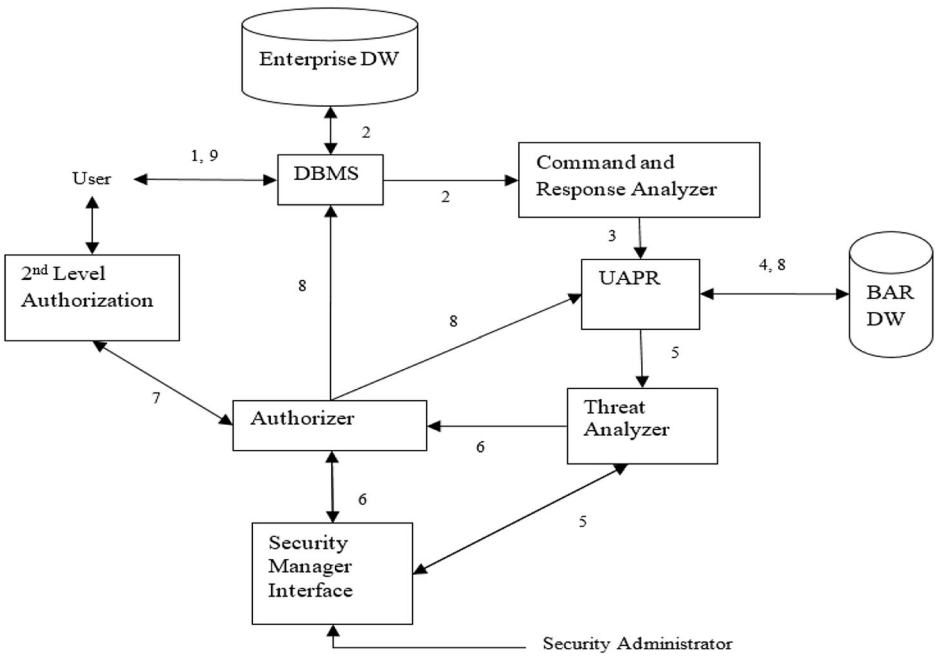

Fig. 1 The architecture of the data warehouse intrusion detection system with second level authentication

As per the authorizer's decision in step 8, the DBMS provides the user's desired results. The results may contain an error message if the user has not answered the security question on second-level authentication initiation.

Here, the system has a provision to ask the user to prove its identity again with another verification level. It gets initiated whenever the current query profile deviates from the user's historical access profile established over time. Once the user has successfully passed the second level authentication, the present query profile is incorporated into the user's historical profile. It further expands user access profile resulting in the reduction of false negatives. In case of a failed attempt of second level authentication, the system administrator and user will be simultaneously notified of a suspicious login attempt. This notification will help the user take appropriate action to change a password, update security questions, etc.

4.1 User profile creation

The user can retrieve some information from the data warehouse only once he gets an approved entry into the system. The extraction of information involves the execution of single or multiple DW-level queries. These queries and their corresponding responses can be broken down into different parameters to form a user access profile that can be modified incrementally over time. These user profile parameters include query content under execution and result after execution [5]. Such parameters form their respective probabilistic distribution for each user at a confidence level of 95 percent, i.e., the settings range from [Mean-2* (Standard Deviation)] to [Mean + 2* (Standard Deviation)]. The user profile has been created for every parameter per user during the training phase. It will be updated incrementally at each verified user access by making respective entries in the BAR repository.

The parameters considered for the creation of a user profile are:

1. *QueryExecutionTime* Total time taken by the query for execution.
2. *QueryLength* The number of characters in the executed query.
3. *ResponseSize* Size (in bytes) in the result of the executed query.
4. *ResponseColumns* Numbers of columns returned in the result of the executed query.
5. *ResponseTables* Number of tables included in the result of the executed query.
6. *GroupByCount* Number of columns included in the GROUP BY clause of the executed query.
7. *AndOrCount* Number of AND clause and OR clause included in the executed query.
8. *JoinsCount* Number of JOINS of any type included in the executed query.

Here, the following hypothesis has been considered by the system:

1. *Null Hypothesis (H_0)* Every user has been deemed an intruder, and the user must prove their identity by matching the current access profile to their user profile inside the BAR repository. When it does not match, a threat analyzer may produce a warning. Threat analyzer may allow a second chance by redirecting it to second-level authentication based on the system administrator's criteria.
2. *Alternate Hypothesis (H_a)* Every user was considered a legitimate user. Whenever the user's current access profile matches the BAR repository's historical user profile or has a successful clearance of second level authentication, the user will be considered a legitimate user. In the event of a positive alternative hypothesis, the user's current access signature in BAR will also be included in an existing user profile.

Given the description of H_0 and H_a, α (alpha) error occurs when a regular user is believed to be an intruder and results in false positives. It means that it should be called α error when a regular user is identified as an intruder. On the other hand, β (beta) error, i.e., false negative, occurs when an attacker is treated as a regular user, enabling access to DW information. Our current research goal is to reduce β error, although reducing both α and β errors lead to an increase in IDS performance [35]. These errors are usually used to identify possible mistakes made in a phase of statistical decision making [36]. Nonetheless, in our case, β error reduction reduces the IDS's primary objective, i.e., detecting an intruder attempt.

4.2 Intrusion detection

When the user profile is created and modified in BAR, any DW query would be translated to user profile parameters. Such individual values will then be compared to the parameter ranges of the respective user's profile to determine the number of outliers. The percentage of mismatch has been compared with the system administrator's agreed threshold. If this percentage of mismatch is below the threshold limit, the user will proceed with the current query. Otherwise, the current user will be offered another chance and diverted to a secret question to prove its identity. The user has already registered the answers to a particular secret question during registration time. If the user answers the secret question successfully, then the

current user's query profile will also be added to the existing user profile in BAR and displays the user's results. Otherwise, the system will terminate the user session and logs it as an intrusion attempt. It will also notify both system administrators and the user about the suspicious login attempt for further action on their part.

The recording of a good second level authentication attempt further strengthens the user access profile. If the user profile is enhanced, it can reduce true intrusion (β error).

5 Experimental evaluation

In MySql, we have implemented the TPC-H [37] Database benchmark to evaluate the effect of second level authentication algorithm on various alerts for data warehouse created by the IDS. For the same, TPC-H schema was introduced with eight different tables (the Base Tables) having relationships, as shown in Fig. 2. The schema was filled with 1 GB storage data, as shown in Fig. 2.

The TPC-H database was set up on MariaDB 5.5.48 running on RHEL 7, 64 Bit Machine with 6 Virtual CPU and 6144 MB of RAM. This virtual machine was built in VMware ESXI 6.0.0 Hypervisor on HP ProLiant DL320e Gen 8 Server running 4X Intel Xeon CPU with 32 GB RAM, 1.63 TB Raid 6 primary storage, and 2.46 TB backup disk. A similar scenario [5] has been taken for testing and comparison, comprising ten web connections to the DW in which 7 were "true" DW users (non-intruders) and 3 "intruders." Every true user executes their own allocated set of queries. The workload for every user contains a combination of three categories of queries:

1. Original benchmark queries of TPC-H (Oq_i), where i is the TCP-H benchmark query number with i = 1–23.
2. Original benchmark queries of TPC-H with modified parameters (Mq_i), where i is the TCP-H benchmark query number with i = 1–23.
3. Every user workload includes random queries created by randomly picking up tables, columns, grouping, and sorting, literal column constraints involved in the WHERE clause.

The snapshot of the workload of each user has been given in Table 3.

The mean and standard deviation of each user function was determined to construct each user's user profile. The user workload was executed 25 times in succession, and values are recorded for each run. When the user profile was developed, the number of user actions and intruder actions

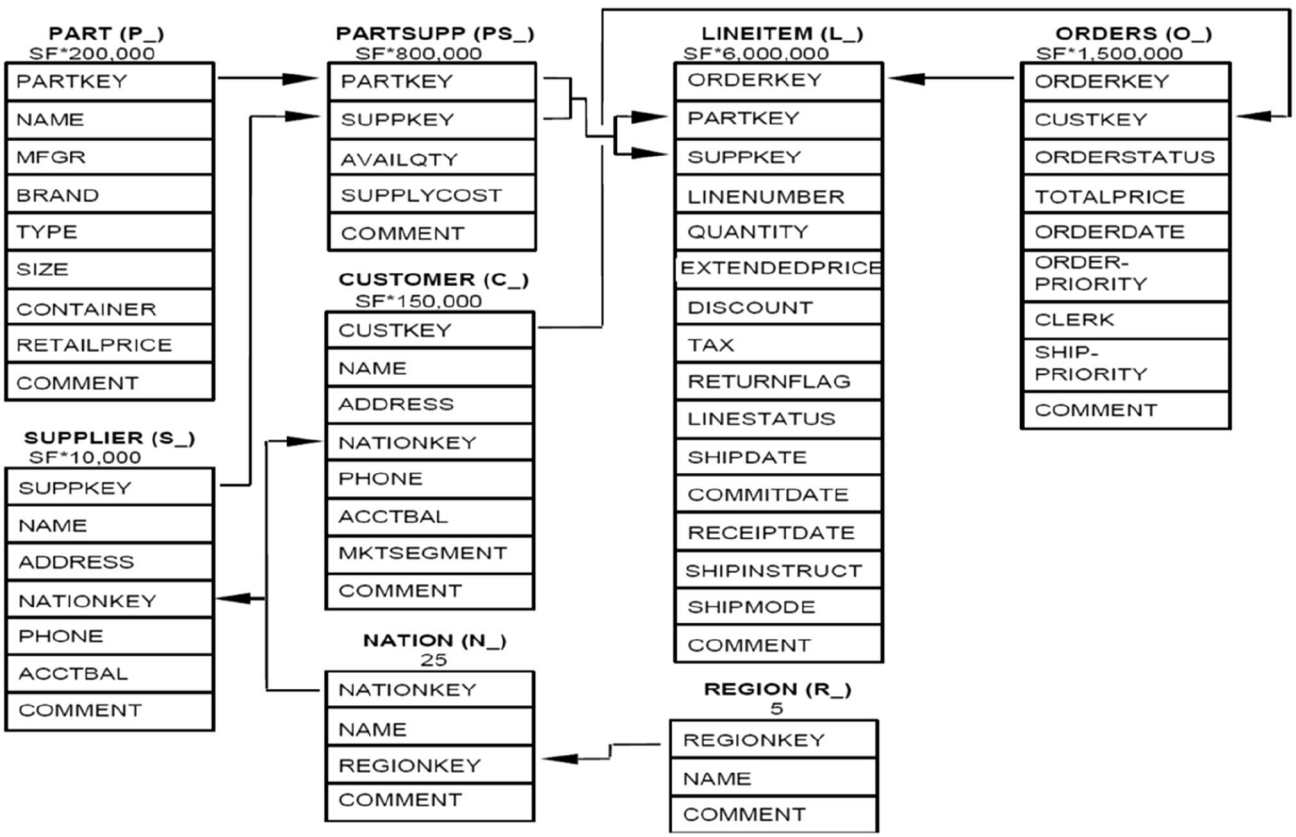

Fig. 2 TPC-H Schema of eight different and individual tables (Base Tables) connected to columns of those tables

Table 3 Query workload of true users

True user	Query workload
1	Oq1, Mq3, Oq6, Mq8, Oq11, Mq12, Mq15, Oq16, Mq19, Oq21 + 2 random queries
2	Oq1, Mq2, Oq4, Mq6, Oq8, Oq10, Oq13, Mq15, Oq17, Mq18, Oq20, Mq22 + 3 random queries
3	Oq2, Mq4, Mq7, Mq9, Oq12, Oq14, Mq16, Oq23 + 1 random query
4	Mq5, Oq7, Oq9, Mq14, Mq23 + 5 random queries
5	Mq1, Oq3, Oq5, Mq10, Mq11, Mq13, Mq17, Oq18, Oq19, Mq20, Mq21, Oq22 + 3 random queries
6	Oq2, Mq4, Oq7, Mq9, Oq12, Oq15, Oq18, Mq19, Mq21, Mq23 + 2 random queries
7	Oq3, Mq5, Mq8, Oq10, Mq12, Oq15, Oq17, Mq18, Oq20 + 5 random queries

were conducted to assess the IDS's performance being proposed. Here, intruder actions are composed of intrusion queries of various categories, i.e., SQL injection tautologies, changing the random number of rows, selecting all rows from the number of tables, etc.

For IDS performance review, every true and intruder query execution has been categorized into the following:

True Positive (TP) True intrusion action has been detected.

False Positive (FP) True user action has been marked as a true intrusion alert resulting in a false alarm.

True Negative (TN) True user action has generated no true intrusion alert resulting in regular user access.

False Negative (FN) True intruder action has not been detected and allowed as a regular user.

The second level authentication process aims to reduce the False Negative (β error), resulting in the lower undetected true intrusion. In this case, for evaluating performance, precision, and accuracy [5] are defined as follows:

$$Precision : \left[\frac{TP}{TP + FP}\right]$$

$$Accuracy : \left[\frac{TP}{TP + FP + TN + FN}\right].$$

Initially in S.No. 1 (Intruder Queries Without Second Level Authentication), about 721 user queries and 240 intruder queries were executed without second level authentication, as seen in Table 4 and Fig. 3. The second level authentication algorithm was applied in S. No. 2 (Intermediate Performance after Implementation of Second Level Authentication) to improve intrusion detection cases. After implementing the second level authentication algorithm, the performance was recorded, and we noted a small shift in the FN rate that dropped from 18.75 to 18 percent. Further executions for both valid user and intruder actions were performed to test it further. A new dip in the FN rate (S. No. 3) (Final Performance after Implementation of Second Level Authentication) was reported from 18 to 17.3 percent. It means including a successful second level authentication query profile in the historical user profile leading to reduced FN rate (β error), i.e., undetected true intrusions. It has also been observed that there is also an increase in FP rate and TP rate due to second level authentication. There was also a decrease in precision and accuracy levels. The FN rate (β error) and other proposed solution parameters are much lower when comparing these findings with the initial experiments [5], as shown in Table 5 and Fig. 4.

The findings indicate a substantial decrease in the FN rate (β error) by including the proposed second level authentication. It was accomplished by increasing robustness through the incorporation of previously unknown behavior into the user profile.

6 Conclusion and future work

Providing second level authentication has been able to boost DW performance significantly. It is evident from the apparent reduced FN rate (β error). Second level

Table 4 Experimental results of various true and intruder queries

S.No	TUA	IA	TP	TP Rate (%)	FP	FP Rate (%)	TN	FN	FN Rate (%)	Accuracy (%)	Precision (%)
1	721	240	195	81.25	18	2.4	703	45	18.75	93.40	91.5
2	804	271	220	81.8	24	2.9	780	51	18.0	93.02	90.1
3	924	311	257	82.6	32	3.4	892	54	17.3	93.03	88.9

S.No. Serial number, *TUA* True user actions, *IA* Intruder actions, *TP* True positive, *FP:* False positive, *TN* True negative, *FN* False negative

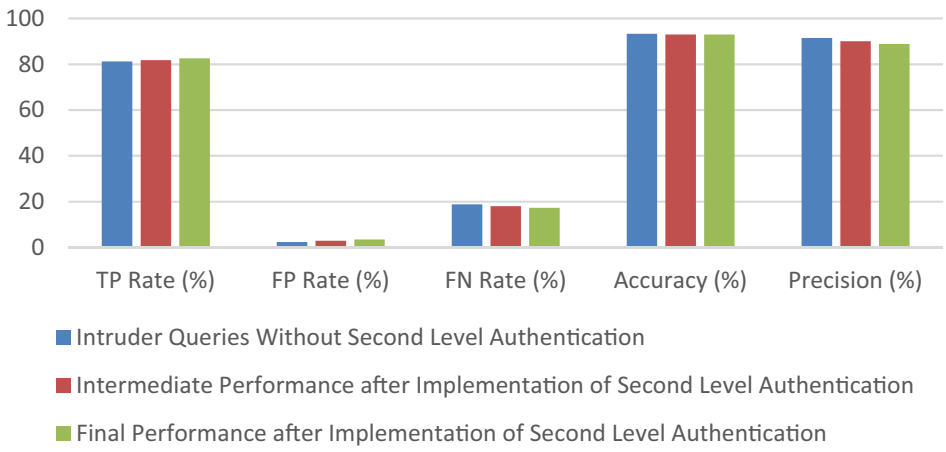

Fig. 3 The bar graph representation for comparison among Intruder queries detection with and without application of Second Level Authentication

Table 5 Comparison among second level authentication based IDS with DBMS application layer intrusion detection for data warehouses

Solution	TP rate (%)	FP rate (%)	FN Rate (%)	Accuracy (%)	Precision (%)
DBMS application layer intrusion detection for data warehouses	78.6	9.5	21.33	87.74	71.3
Second level authentication based IDS	82.6	3.4	17.3	93.03	88.9

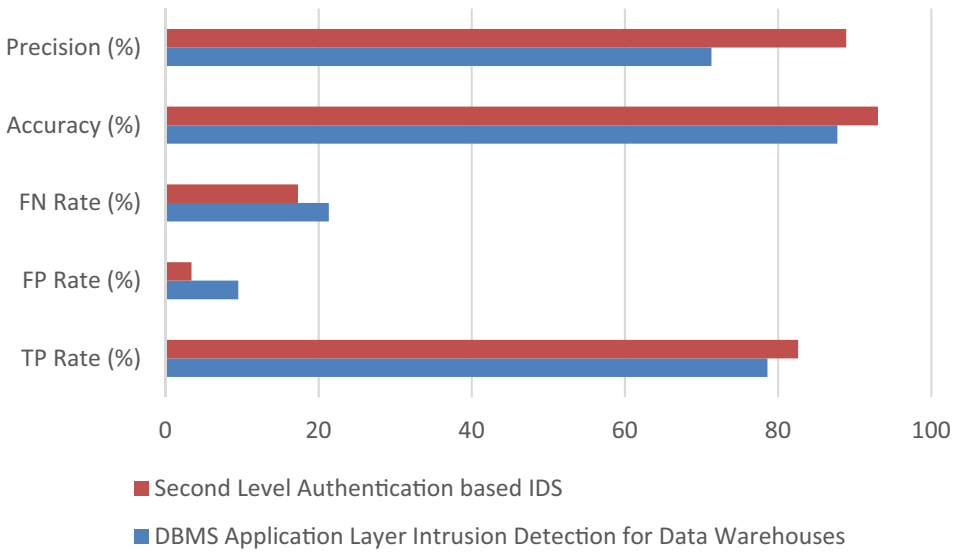

Fig. 4 The bar graph representation for comparison among Second Level Authentication based IDS with DBMS Application Layer Intrusion Detection for Data Warehouses

authentication also helps to simplify the query execution cycle and reduces system administrator reliance. Because if the user's current application profile diverts minutely from its historical profile, instead of stopping the application entirely user may be redirected to second level authentication. Nevertheless, if the application profile diverts by a significant percentage from the user's historical profile, second level authentication will not be enabled. The system administrator must determine the threshold for the provision of second level authentication according to DW data necessity and sensitivity. It helps the system to strike a balance between security and user comfort. Here, FN rate reduction (β error) is also a significant success of current research as it leads to a decrease in IDS failure rate, i.e., failure to detect a real intruder.

We implemented the proposed model on the TPC-H database, and the results were compared with the existing proposal [5]. The results show a significant decrease in the FN rate by 18.92% and the FP rate by 64%. However, the FP rate reduction is mainly due to the user's second chance during the initial FP event. The conversion of FPs to TPs via successful second chances events also contributed to an increase in TPs by almost 5%.

In the future work, analysis of the threshold for initiation of second level authentication can be performed. It may lead to a standard threshold value, which is acceptable to most applications. Analysis of some additional user and query profile parameters may be performed, leading to a further reduction in the FN rate.

References

1. Inmon WH (1991) Building the data warehouse. Wiley and Sons, New York
2. Santos RJ, Bernardino J, Vieira M (2014) Approaches and challenges in database intrusion detection. ACM SIGMOD Rec 43:36–47
3. Debar H, Dacier M, Wespi A (1999) Towards a taxonomy of intrusion-detection systems. Springer, Heidelberg 31:805–822. https://doi.org/10.1016/S1389-1286(98)00017-6
4. Thuraisingham B, Iyer S (2007) Extended RBAC—based design and implementation for a secure data warehouse. ARES'07. IEEE, Vienna, pp 367–382
5. Santos RJ, Bernardino J, Vieira M (2013) DBMS application layer intrusion detection for data warehouses. In: Building sustainable information systems. Springer, Boston
6. Gosain A, Arora A (2015) Security issues in data warehouse: a systematic review. Elsevier, Amsterdam, pp 149–157
7. Sandhu R (1995) Issues in RBAC. In: RBAC '95. ACM, New York, Gaithersburg, Maryland, USA, p 6
8. Emre U, Vijayalakshmi A, Jaideep V et al (2014) Security analysis for temporal role based access control. J Comput Secur 22:961–996
9. Ali S, Rauf A, Khusro S et al (2014) An authorization model to access the summarized data of data warehouse. Life Sci J 11:608–610
10. Shaikh RA, Adi K, Logrippo L (2012) Dynamic risk-based decision methods for access control systems. Comput Secur 31:447–464. https://doi.org/10.1016/j.cose.2012.02.006
11. Singh PB, Chugh U, Kathuria M (2019) A review on intrusion detection system. Int Res J Eng Technol (IRJET) 6:1351–1358
12. Pietraszek T (2004) Using adaptive alert classification to reduce false positives in intrusion detection. International workshop on recent advances in intrusion detection. Springer, Berlin, Heidelberg, pp 102–124
13. Pietraszek T, Tanner A (2005) Data mining and machine learning—towards reducing false positives in intrusion detection, vol 10. Elsevier, Amsterdam, pp 169–183. https://doi.org/10.1016/j.istr.2005.07.001
14. Hu Y, Panda B (2004) A data mining approach for database intrusion detection. In: SAC '04: Proceedings of the 2004 ACM symposium on Applied computing. Association for Computing Machinery, New York, NY, United States, Nicosia, Cyprus, pp 711–716
15. Bockermann C, Apel M, Meier M (2009) Learning SQL for database intrusion detection using context-sensitive modelling. Detection of intrusions and malware, and vulnerability assessment, DIMVA 2009. Springer, Berlin, Heidelberg, pp 196–205
16. Ficke E, Schweitzer KM, Bateman RM, Xu S (2019) Analyzing Root Causes of Intrusion Detection False-Negatives: Methodology and Case Study. In: MILCOM 2019—2019 IEEE Military Communications Conference (MILCOM). pp 1–6
17. Joshi JBD, Bertino E, Latif U, Ghafoor A (2005) A generalized temporal role-based access control model. IEEE Trans Knowl Data Eng 17:4–23. https://doi.org/10.1109/TKDE.2005.1
18. Atluri V, Gal A (2002) An authorization model for temporal and derived data: securing information portals. ACM Trans Inf Syst Secur 5:62–94. https://doi.org/10.1145/504909.504912
19. Ray I, Toahchoodee M (2007) A spatio-temporal role-based access control Model. In: Barker S, Ahn G-J (eds) Data and applications security XXI. Springer, Berlin Heidelberg, pp 211–226
20. Uzun E, Atluri V, Vaidya J et al (2014) Security analysis for temporal role based access control. J Comput Secur 22:961–996. https://doi.org/10.3233/JCS-140510
21. Atlam HF, Azad MA, Alassafi MO et al (2020) Risk-based access control model: a systematic literature review. Future Internet 12:103
22. Anil S, Remya R (2013) A hybrid method based on genetic algorithm, self-organised feature map, and support vector machine for better network anomaly detection. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). IEEE, pp 1–5
23. Divya T, Muniasamy K (2015) Real-time intrusion prediction using hidden markov model with genetic algorithm. In: Suresh LP, Dash SS, Panigrahi BK (eds) Artificial Intelligence and evolutionary algorithms in engineering systems. Springer India, New Delhi, pp 731–736
24. Ramachandran R, Arya P, Jayanthi PG (2017) A novel method for intrusion detection in relational databases. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, Udupi, India
25. Ramachandran R, Nidhin R, Shogil PP (2018) Anomaly detection in role administered relational databases—a novel method. In: 2018 International conference on advances in computing, communications and informatics (ICACCI). IEEE, Bangalore, India
26. Rao UP, Sahani GJ, Patel DR (2010) Machine learning proposed approach for detecting database intrusions in RBAC enabled databases. In: 2010 second international conference on computing, communication and networking technologies. pp 1–4
27. Darwish SM, Guirguis SK, Ghozlan MM (2013) Intrusion detection in role administrated database: transaction-based approach. In: 2013 8th international conference on computer engineering systems (ICCES). pp 73–79
28. Darwish SM (2016) Machine learning approach to detect intruders in database based on hexplet data structure. J Electr Syst Inf Technol 3:261–269. https://doi.org/10.1016/j.jesit.2015.12.001
29. Mathew S, Petropoulos M, Ngo HQ, Upadhyaya S (2010) A data-centric approach to insider attack detection in database systems. In: Jha S, Sommer R, Kreibich C (eds) Recent advances in intrusion detection. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 382–401
30. Anderson RH, Brackney RC (2004) Understanding the insider threat. In: Proceedings of a March 2004 workshop. RAND CORP SANTA MONICA CA
31. Kamra A, Terzi E, Bertino E (2008) Detecting anomalous access patterns in relational databases. Springer-Verlag 17:1063–1077. https://doi.org/10.1007/s00778-007-0051-4
32. Parmar J, Jain P (2013) A different approach of intrusion detection and Response System for Relational Databases. In: 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE). pp 894–899
33. dos Santos DR, Marinho R, Schmitt GR et al (2016) A framework and risk assessment approaches for risk-based access control in the cloud. J Netw Comput Appl 74:86–97. https://doi.org/10.1016/j.jnca.2016.08.013

34. Gosain A, Arora A (2016) Two Level Signature Based Authorization Model for Secure Data Warehouse. Springer, Singapore, pp 251–257
35. Anuar NB, Sallehudin H, Gani A, Zakaria O (2008) Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree. Malays J Comput Sci 21:101–115
36. Gowadia V, Farkas C, Valtorta M (2005) PAID: a probabilistic agent-based intrusion detection system. Comput Secur 24:529–545. https://doi.org/10.1016/j.cose.2005.06.008
37. TPC (2018) TPC Benchmark H, Decision Support Benchmark. In: TPC-H. http://www.tpc.org/tpch/. Accessed 26 Mar 2020