Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FALSE ALARM REDUCTION SCHEME FOR DATABASE INTRUSION DETECTION SYSTEM

AYMAN MOHAMED MOSTAFA, FATEN AYIED ALMUTAIRI, M.M. HASSAN
Faculty of Computers and Informatics, Zagazig University, 44519, Egypt
E-mail: am_mostafa@zu.edu.eg

ABSTRACT

Database intrusion detection system is considered a mandatory security layer in recent database applications. The detection of intrusions in database applications is mostly based on anomaly methods like access patterns, association rule mining and mining data dependencies between data items. These countermeasures achieve good results in traditional applications but new forms of attacks on computer systems lead to the depreciation of intrusion detection systems due to the high rates of false positive alarms. The goal of this paper is to improve the accuracy of intrusion detection system by reducing false alarms using alert clustering mechanism and system hibernation capabilities. In this paper, a three-stage access control framework is developed for detecting malicious users in database. This framework is embedded with an alert clustering mechanism for reducing false alarms by correlating low-level alerts into one cluster. A post security countermeasure is developed by merging system hibernation capabilities into the developed application. The hibernation mechanism is used for maintaining the availability of data in case of intrusion detection. The experimental results of the proposed algorithm achieve high detection rate with low false positive and low false negative alarms when compared to recent researches in intrusion detection systems.

Keywords: *Intrusion Detection System, Alert Clustering, Hibernation Mechanism*

1. INTRODUCTION

The security system in recent organizations is based on network and database security. Network security is considered the outer layer for preventing malicious access to the network. Database security is considered the last line of defense for protecting the confidentiality of information stored in database. Computer security is used for protecting networks and database from attack activities such as violation of privacy, fraud, deterioration of data and access to unauthorized information. Given their essential role in detecting intrusive activities, intrusion detection systems (IDS) are considered the fundamental components of computer system security. Therefore, the precision of database intrusion detection systems depends on their ability to detect real attacks on the system and to send alarms to the database administrator.

Attacks on database are based on external attacks and internal attacks [1]. External attacks are performed by malicious users who try to disclose confidential information. Internal attacks are performed by authorized users who misuse their privileges intentionally or unintentionally. Based on internal attacks, high false alarms are raised due to the inability of the intrusion detection system to differentiate between real attacks and unintentional misuse. A false alert is defined as a signal triggered by an intrusion detection system recording an attack although it is normal network traffic. Due to the high false alarms triggered by intrusion detection systems, the accuracy of the overall security system decreases. The contribution of this paper is as follows:

Developing an access control mechanism based on an authentication code and secret key for authorized users.

- Monitoring and tracking false alerts detected by the access control mechanism.
- Developing an alert clustering algorithm to verify the relationship between false alerts and correlate low level alerts into one cluster.
- Building a hibernation mechanism for protecting confidential data from disclosure by converting suspicious users to an alternate system until they were verified.

This paper is organized as follows: section 2 presents recent researches of intrusion detection systems and hibernation methodologies. Section 3 presents an access control mechanism of the proposed system. Section 4 presents the proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

authentication mechanism. Section 5 presents alert clustering algorithms for managing and minimizing false alerts. Section 6 explains the hibernation mechanism. Section 7 presents experimental results of the proposed application and algorithms. Finally, the paper is concluded in section 8.

2. RELATED WORK

The role of intrusion detection system and system hibernation is very important to protect the security of confidential information. Many types of research present approaches that develop the concept of intrusion detection system and system hibernation based on data mining, clustering, decision tree, classification, machine learning, and chameleon agent.

Intrusion detection systems observe the network package independently within the web server and database systems [2]. In multi-tier applications, the backend database server may be threatened behind the firewall while the web server is remotely accessed over the internet. This can increase the intrusions that may disclose confidential information from database. One of the recent researches of intrusion detection systems is presented in [3]. In this research, a set of artificial intelligence machine learning methods is used to decrease the number of false positive alarms in an anomalous intrusion detection data. This method combines data clustering using the simple k-means algorithm, and self- organization maps to effectively reduce false positive alarms. This method is applied for intrusion detection over networks not database. As presented in [4], a three stage process to detect false alerts and outliers are developed. In the first stage, a set of elementary alerts is clustered to create a set of meta–alerts. In the second stage, the outliers are removed from the set of meta–alerts using a binary optimization problem. Finally, a binary classification algorithm is proposed to classify meta–alerts if it is false alerts or real attacks. An intrusion detection system based interaction on mobile agents and cluster density algorithm is presented in [5]. In this research, the IDS system is able to detect known and unknown attacks and reduce the rate of false positive and negatives by coupling two recent technologies: mobile agents and a new data mining algorithm called clust-density in one single intrusion detection system named ids-am-clust. The simulation results are obtained by implementing the system for proving high level of intrusion detection. As presented in [6], a predictive model for network intrusion detection is developed based on data stream mining techniques into which the data stream is processed with a class prediction associated with it. An intrusion detection system for wireless sensor networks is presented in [7]. In this research a methodology for anomaly detection is proposed based on binary logistic regression. A recent research of intrusion detection systems is presented in [8]. This research deals with the problem of redundant data that may lead to unreliable intrusion detection. Neural network is used to test the redundancy of data and its effect on the performance of intrusion detection.

3. ACCESS CONTROL MECHANISM

Access control mechanism is considered as a comprehensive preventative measure for keeping the internal and external perimeters secure from unauthorized access and malicious intrusions. As presented in [7], four categories of attacks are used to disclose confidential information from database. First: denial of service. Second: remote to user (R2U) which means unauthorized access from a remote machine. Third: user to remote (U2R) which means unauthorized access to root privileges. Finally: probing which means an attempt to access computer applications through a probable weak point. As presented in [9], a database intrusion detection system for enhancing detection rate is proposed. In this approach, a user profile matching between user activity and his/her database profile is used. If an exact matching is applied between the current user profile and the valid profile, high false alarms will be raised. This is due to the low level of the system tolerance to any change in user activity. For reducing false alarms, an f-triplet format is applied based on altering of consecutive select commands and allowing a subset of attribute access patterns to be accessed in new transactions.

In this paper, an access control mechanism for reducing false alarms of intrusion detection systems is implemented. As presented in Figure 1, during the login process, three-level authentication is provided for controlling user access to database applications. First: the user provides the system with username, password, and email. Second: user authentication code. Third: user secret key. If the parameters of first, second, or third level are authentic, the system proceeds to the next level otherwise the system hibernates to a spoofing session.

Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

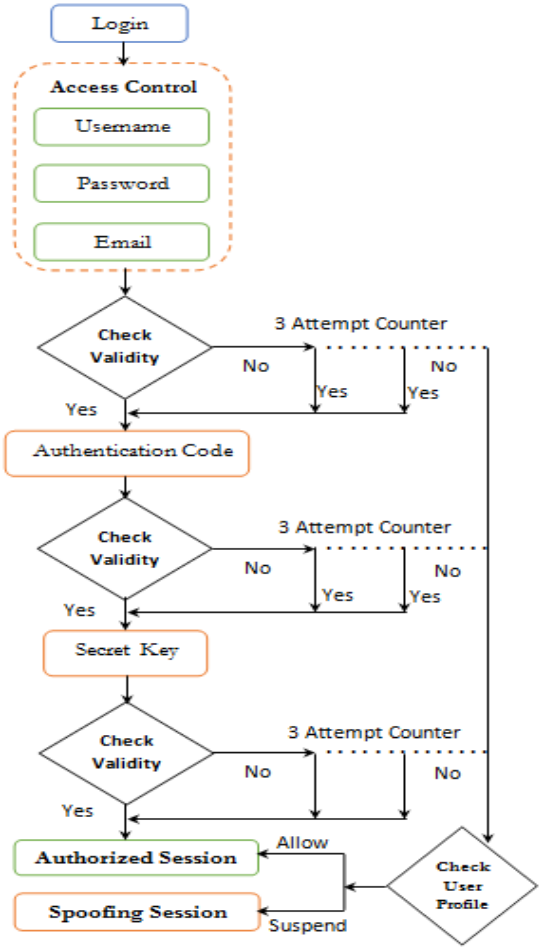

Figure 1. Access Control Mechanism

Spoofing is a mechanism used as a defense for confusing attackers. As presented in Figure 1, if the authorized user fails in logging into the system three times, the activity profile of the user is checked by the administrator to allow or suspend the request. The authorized session is permitted; if the user is allowed to login into the system. Otherwise, the spoofing process diverts the user to an alternate session that passes false information to the user without affecting the original database. In order to authenticate user profile, an authentication mechanism is applied for verifying user profile and enumerates false alarms generated in the system.

4. AUTHENTICATION MECHANISM

In this stage, an authentication mechanism is developed based on three-layer security parameters: user verification, authentication code, and secret key certificate. These layers are presented in the following sections.

4.1 User Verification

The user verification is based on 4 parameters: username, password, email, and captcha for human-machine verification. In this stage, the four parameters must be authentic to allow the user to proceed to the next layer. Otherwise, the hibernation mechanism will be activated and an alarm will be raised to the administrator.

4.2 Authentication Code

The authentication code is generated after verifying access control parameters. Based on these parameters, a new authentication code is randomly generated each time the user logs in into the system. As presented in Algorithm 1, before generating a new authentication code; the system must perform three nested pre-security stages for verifying the specified user. First: the system checks the IP address of the connected user whether it is locked or not. If the IP is locked from a previous process, the system blocks the user connection to the system. Otherwise, the system proceeds to the next stage. In the next stage, the system verifies the username of the user. If it is disabled by the database administrator, then this user is a suspicious user. As a result, the system blocks the connection. The last stage, the system checks the number of alarms recorded to the connected user. If the number of alarms is >= 3, the system blocks the user connection. Otherwise, the system generates a random authentication code using MD5 authentication algorithm.

Algorithm 1: Authentication Code Generation

```
1.  Begin
2.  Check IP address
3.   If IP = Lock Then
4.      Block_Connection
5.   Else
6.      If username = disable Then
7.        Block_Connection
8.      Else
9.        If Alert_count >= 3 Then
10.          Block_Connection
11.     Else
12.       Generate auth_code
13.     End If
14.  End If
15. End If
16. End
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.3 Secret Key Certificate (SKC)

After verifying the authentication code, a random private key is generated by the database administrator (DBA) that will be merged with the authentication code to generate the secret key certificate (SKC). Firstly, the database administrator (DBA) generates his own public and private key $\mathbf{PU_{DBA}}$ and $\mathbf{PR_{DBA}}$ respectively.

As presented in Figure 2, each user must send a request to the DBA to obtain his own private key. This request contains the timestamp of the request ($\mathbf{T_n}$), identification of the user ($\mathbf{ID_n}$), and the user authentication code ($\mathbf{Auth_{Un}}$). The database administrator (DBA) will replay on this request by encrypting the previous parameters and generates the private key ($\mathbf{PR_{Un}}$). Even if two distinct users sent requests on the same timestamp, the user identification ($\mathbf{ID_n}$) and user authentication code ($\mathbf{Auth_{Un}}$) will remain different. As a result, each user will obtain a distinct private key. The private key cannot be counterfeited because it was encrypted by the private key of database administrator $\mathbf{PR_{DBA}}$.

The secret key certificate of the user is generated by merging the authentication code and private key as presented in formula 1.

$$C_{SK} = PR_U \,||\, Auth_U \qquad (1)$$

Where C_{SK} is the secret key certificate, PR_U is the private key, and $Auth_U$ is the user authentication.

Based on this mechanism, many features have been achieved:

- Each user can obtain his secret key based on his identification and authentication code.
- Each user can verify that the private key is originated from the database administrator and is not counterfeit.
- The database administrator has only the ability to generate and update the private key.

5. FALSE ALARM REDUCTION

As presented in authentication mechanism, several layers are implemented to authenticate system users in order to reduce false positive and false negative alarms. False positive alarms refer to normal actions executed by normal users but are detected by the system as abnormal activities. False negative alarms refer to the number of attacks that breach the system.

Many research papers focused several methods for reducing false alarms. As presented in [10], a framework for database intrusion detection is proposed. This framework consists of two modules: specifications and detection modules. The first one identifies transactions and stores them where the second module identifies malicious transactions based on the pre-stored transactions. As presented in [11], data mining is being used for detecting intrusions in database. In this research, different classification algorithms for detecting malicious users are listed. One of the recent researches that focused on reducing false alarms is presented in [12]. In this research, an intrusion detection mechanism based on the user profile is presented. This mechanism is based on

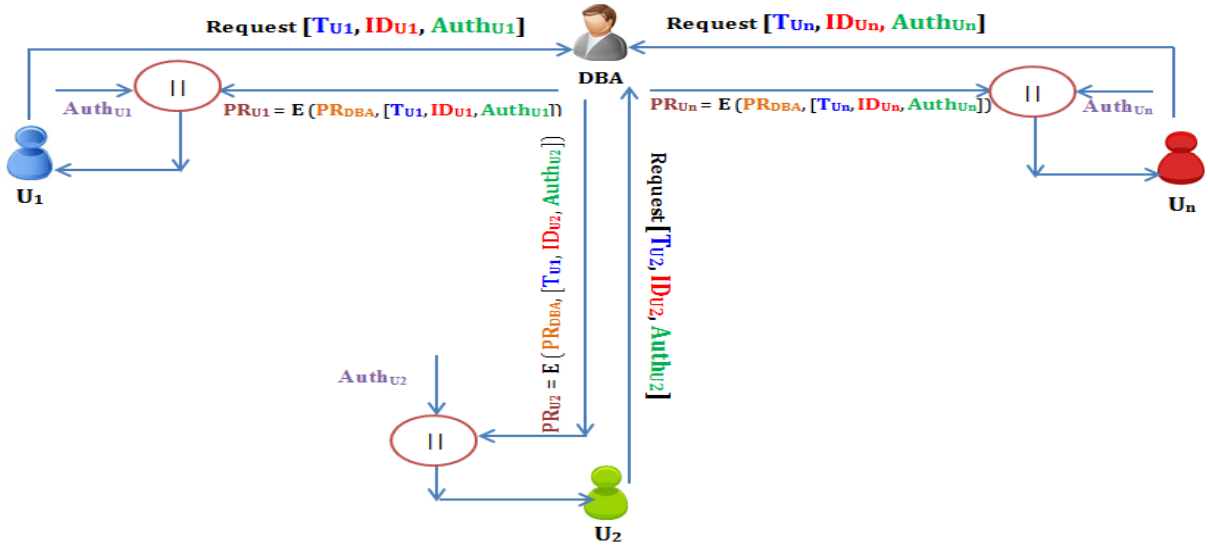

Figure 2. Secret Key Certificate Transmission

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

using machine learning techniques for clustering data. As presented in [13], two approaches are being used to reduce false positive alarms: detection techniques and alert processing technique. Detection techniques are based on detecting anomalous users during the execution of transactions. This means that users have successfully accessed the system and may abuse their privileges in disclosing confidential information. Alert processing techniques are based on detecting users by analyzing previous alarms that have been raised during their activities on the system.

In order to achieve high detection rate with low false positive and low false negative alarms, an alert clustering algorithm is implemented by correlating similar users that have the same behavior in one cluster. As presented in [4], the correlation process is based on increasing the similarity degree between objects that exist in the same cluster and decreasing it between clusters.

Algorithm 2: Alert Clustering

```
Set username, password, email
INPUT:
Log_count = 0
Alert_count = 0
Activity_B = OFF
Activity_S = OFF
Activity_A = OFF
 Begin
Check IP address
 If IP = Lock Then
  Block_Connection
End If
Else
Check Login Dictionary
If username = disable Then
    Block_Connection
End If
Else
Check User State
Check Login Dictionary
If user = new Then
   If Log_count >= 3 Then
     Raise_Alarm
     Block_Connection
     Alert_count += 1
   End If
End If
Else
 If user = old Then
   Check Alert Dictionary
  If Log_count >=3 and Alert_count= 0 Then
     Raise_Alarm
     Suspend_Connection
     Open Spoofing_Session
  Else
If Log_count >=3 and Alert_count >0 Then
    If Activity_B = ON Then
      Block_Connection
      IP = Lock
    End If
    Else
      If Activity_S = ON Then
     Open Spoofing_Session // Admin Req
      End If
      Else
        If Activity_A = ON Then
      Open Spoofing_Session // Admin Req
        End If
   End If
  End If
End If
End
```

As presented in Algorithm 2, the system checks the user profile if it is new or old. If the user is new, all security procedures that have been explained will be followed. Each time the user fails in passing correct parameters, the system increments the alert counter by 1 until the counter reaches the maximum number of limit allowed then the connection is blocked.

The system records all user attempts in the alert dictionary for future checking. If the user is old, the system checks the alert dictionary. If the log counter >= 3 and no alerts exist, the system suspends the connection and hibernates the system. If the log counter >= 3 and the alert counter > 0, the system blocks the connection and the user IP will be locked.

As presented in Table 1, the database administrator has the authorizations to block or suspend any suspicious user based on his user activity profile. If the user is authorized, the database administrator can allow connection to him/her as explained with user U_1. This can reduce future alarms to be raised. For user U_3 and U_5, the system increments alert counter by 1 because the next alert status is turned ON.

All user information profiles with their alerts are presented in Table 2. In this table, the IP address of all users and alert date and time with alert options are explained.

Table 1. Alert Dictionary

Table 2. User Alert Management

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

User	User Status	Log Count	Alert Count	Alarm	Alert Status	Next Alert
U_1	old	3	1	ON	Allow	Alarm OFF
U_2	old	4	1	ON	Block	Alarm ON
U_3	old	3	0	ON	Suspend	Alarm ON
U_4	new	0	0	OFF	Allow	Alarm OFF
U_5	new	3	0	ON	Block	Alarm ON
Next Alarm Status						
U_2	old	4	1	ON	Block	Alarm ON
U_3	old	3	1	ON	Suspend	Alarm ON
U_5	new	3	1	ON	Block	Alarm ON

No	User Name	IP	Address	Alert Date	Alert Time	Alert No	Alert Status	Alert Options		
1	Frank	192.168.1.9	Alert: 3 times login	05/01/2018	3:54 am	2	Suspended	Allowed	Block	Suspended
2	User1	192.168.8.112	Alert: 3 times login	18/10/2017	9:50 am	3	Allowed	Allowed	Block	Suspended
3	User2	192.168.8.106	Alert: 3 times login	08/01/2018	12:12 pm	3	Allowed	Allowed	Block	Suspended
4	Scott1	192.168.8.107	Alert: 3 times login	23/12/2017	10:22 am	3	Blocked	Allowed	Block	Suspended
5	Allan2	192.168.8.123	Alert: 3 times login	15/08/2017	8:56 am	1	Allowed	Allowed	Block	Suspended
6	Adams2	192.168.8.115	Alert: 3 times login	06/01/2018	12:22 pm	3	Allowed	Allowed	Block	Suspended
7	John1	192.168.8.137	Alert: 3 times login	16/11/2017	01: 24 pm	1	Allowed	Allowed	Block	Suspended

6. HIBERNATION MECHANISM

The security system must be hibernated until the suspended transactions are approved or disapproved. As presented in [14], an alternate schema is developed to obtain all transactions from normal users until the suspension process is finished. As presented in Figure 3, when the security system verifies a suspicious user, his transactions are suspended in the original database which resides in the database server. The transactions are verified whether to be saved into database or not. The time spent in the verification process will delay other transactions that are executed by normal users. This will increase the time complexity.

In order to keep an efficient, flexible, and solid security system; data hibernation is executed by transferring data from the original data source to the alternate one. An important measure that is required to be taken for data hibernation is to design an alternate data source that is in compliance with the original one. If the alternate data source is in compliance with the original one, it is possible to transfer the data from original to alternate data source [15].

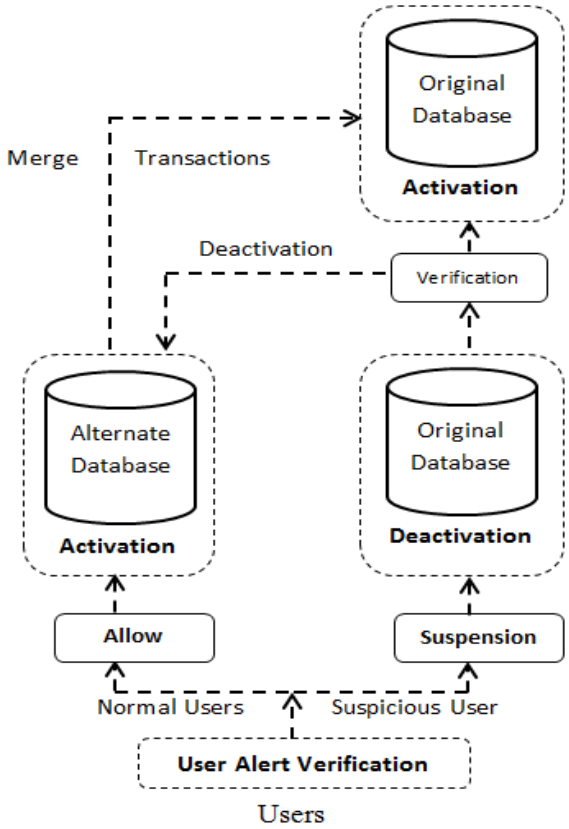

Figure 3. System Hibernation

7. EXPERIMENTAL RESULTS

Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The database security engine was implemented using Microsoft SQL server 2008 for storing database transactions, alert dictionary, and alert management options. The authentication code, secret key certificate exchange and alert clustering algorithm were implemented using Microsoft Visual Studio C# 2015 for deploying the security system components on a client-server application. The experiments were conducted on an Intel(R) Core (TM) i5 @2.40 GHz machine with 8 GB RAM. The operating system was Microsoft Windows 7.

As presented in Table 3, a set of 480 normal events and 171 abnormal events were divided into two groups and were tested using the proposed authentication and alert clustering algorithms.

Table 3. Testing Data Distribution

Data Groups	Normal Events	Abnormal Events
Group 1	210	75
Group 2	270	96

The experimental results are compared to the results of four algorithms presented in [4] which are: binary classification algorithm (BCA), self-organizing map (SOM), decision support classification (DSC) and cluster and nearest neighbor (CANN).

As presented in Figure 4, the detection rate is calculated by dividing the total number of detected users (N_d) to the total number of users (N). This is presented in formula 2.

$$D_R = \frac{N_d}{N} \times 100\% \quad (2)$$

The security system produces a high detection rate with 97.33% and 93.75% in group 1 and group 2 respectively, while the detection rate using binary classification algorithm (BCA) presented in [4] is 97.2% and 91% respectively.

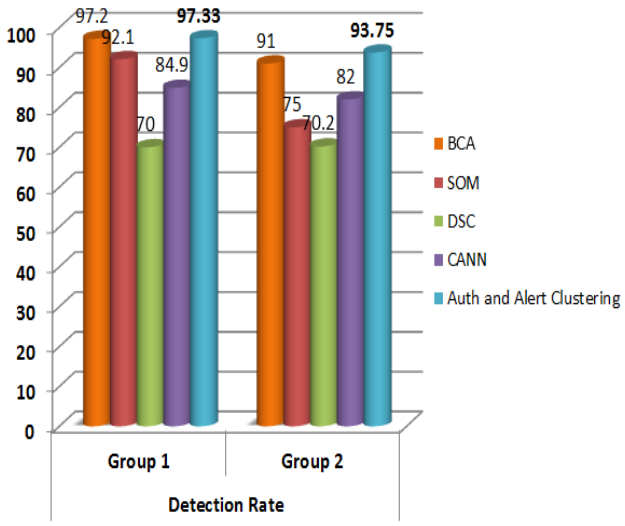

Figure 4. Detection Rate of Alert Algorithm

As presented in Figure 5, the false negative alarms (FN) are calculated by dividing the number of passed malicious users (Np) to the total number of examined users (N). This is presented in formula 3.

$$F_N = \frac{N_p}{N} \times 100\% \quad (3)$$

The security system produces low false negative alarms with 2.67% and 6.25% in group 1 and group 2 respectively, while the false negative alarms using binary classification algorithm (BCA) presented in [4] is 2.8% and 9% respectively, while SOM, DSC, and CANN techniques achieve high false negative alarm rates.

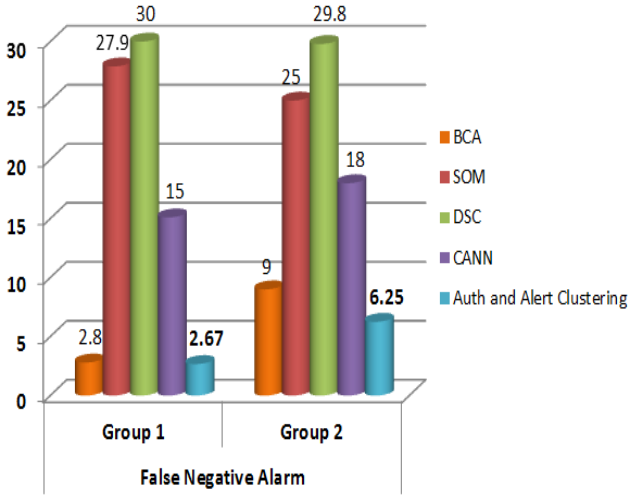

Figure 5. False Negative of Alert Algorithm

Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

As presented in Figure 6, false positive alarms (FP) refer to the probability that normal actions being alarmed as abnormal actions. This is presented in formula 4.

$$F_P = \frac{N_f}{N} \times 100\% \quad (4)$$

In Figure 6, the authentication and alert clustering algorithm achieves low false alarm rates with 4.76% and 4.81% in in group 1 and group 2 respectively, while the false positive alarms using binary classification algorithm (BCA) is 6.5% and 5.8% respectively, while SOM, DSC, and CANN techniques achieve high false positive alarm rates.

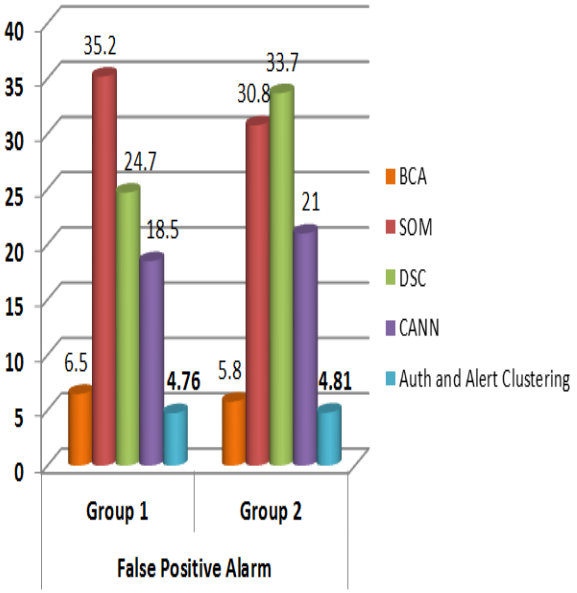

Figure 6. False Positive of Alert Algorithm

The correctness rate (CR) of the proposed experimental study is presented in Figure 7. The correctness rate refers to the probability of correct detection and is calculated by subtracting the false positive and false negative alarms from 1 as explained in formula 5.

$$C_R = (1 - F_P - F_N) \times 100\% \quad (5)$$

As presented in Figure 7, the correctness rate achieves high correct detection with 92.57% and 88.94% in group 1 and group 2 respectively while other algorithms achieve lower correct detection rate.

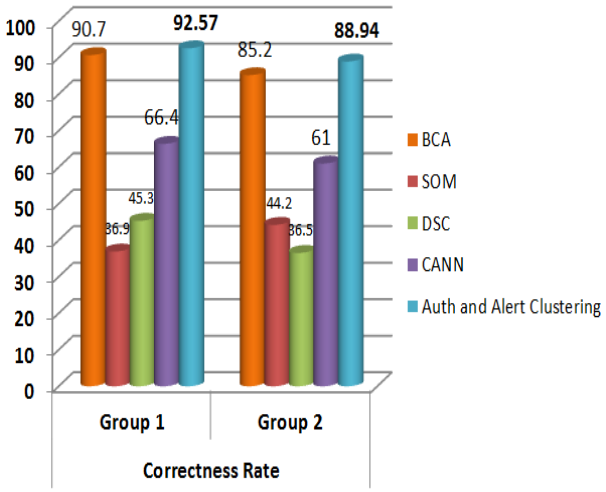

Figure 7. Correctness Rate of Alert Algorithm

In order to ensure the security system efficiency, it was tested again using 50, 100, 150, and 200 transactions and the experimental results were compared with the results presented in [16]. As presented in Figure 8, the detection rate of the proposed algorithm recorded 96%, 97%, 98, and 97.5% respectively, while the intrusion detection system presented in [16] recorded lower detection rates.

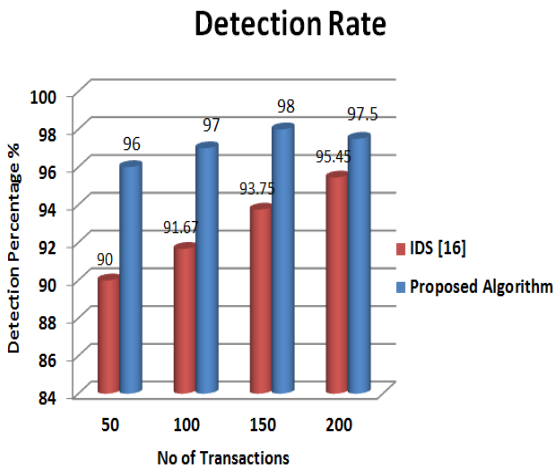

Figure 8. Detection Rate

In Figure 9 and 10, the proposed algorithm proves low false negative and low false positive rates when applying the four data groups 50, 100, 150, and 200. The false negative alarms recorded 4, 3, 2, and 2.5% respectively while false positive alarms recorded

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

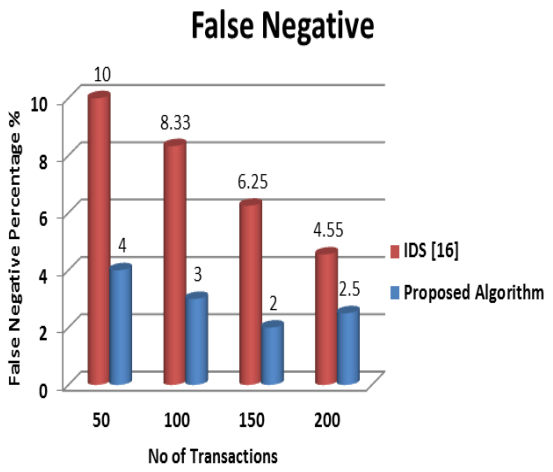

Figure 9. False Negative

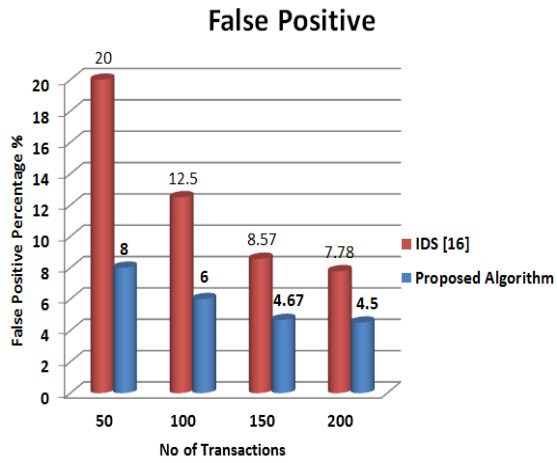

Figure 10. False Positive

As presented in Figure 11, the correctness rate of the proposed algorithm recorded 88%, 91%, 93.33% and 93% on all data groups respectively.

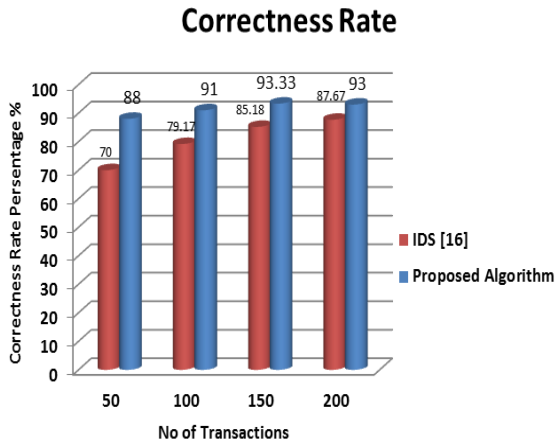

Figure 11. Correctness Rate

8. ALARM REDUCTION SCHEME KEY FEATURES

Prior research works focused on reducing false alarm by analyzing alarm rates using proposed classification algorithms and machine learning techniques for measuring accuracy as presented in [4] and [16]. Various features have been deployed in this paper to reduce false alarm rates as follows:

- An access control mechanism is presented for applying user authorizations based on multi-layer security parameters.
- An authentication code is generated for verifying users accessing the system.
- A secret key certificate (SKC) is applied by DBA based on user authentication code and user private key.

These security countermeasures are deployed for reducing a proportion of false alarms. In order to reduce the greatest possible level of false alarms, two additional features are added:

- An alert clustering mechanism is implemented for analyzing similar alerts based on previous user profile.
- A hibernation mechanism is deployed for hibernating original database if suspicious user is detected.

9. CONCLUSION

Traditional techniques of intrusion detection systems cause high false positive and high false negative alarms due to the inability to identify unauthorized users from authorized users who may misuse their privileges. Recent advances in intrusion detection systems deal with applying new methodologies and algorithms for reducing false alarms in intrusion detection systems. In order to achieve high detection rate with low false positive and low false negative alarms, an authentication mechanism with alert clustering algorithm is proposed and implemented. A hibernation mechanism is embedded in the security system to divert suspicious users to an alternate system instead of recording them as malicious users; until they were verified. The system was tested with two recent experimental studies to verify its efficiency in reducing false alarms. The system achieved high detection rate with low false alarms. In future research, the proposed scheme will be deployed on cloud infrastructure to check the performance of the system based on user privileges on cloud. An enhanced alert mechanism will be proposed based on cloud services.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] D. Nandasana, and V. Barot, "A Framework for Database Intrusion Detection System," IEEE International Conference on Global Trends in Signal Processing, Information Computing and Communication, 2016, pp. 74-78

[2] J. Parmar, "Data Security, Intrusion Detection, Database Access control, Policy Creation and Anomaly Response Systems-A Review," IEEE International Conference on Advances in Engineering and Technology Research, 2014, pp. 1-6

[3] A. Landress, "A Hybrid Approach to Reducing the False Positive Rate in Unsupervised Machine Learning Intrusion Detection," IEEE International Conference on Computer Science and Convergence Information technology, 2016, pp. 1-6

[4] F. Hachmi, K . Boujenfa and M .Limam, "A three – stage process to detect outliers and false positive generated by intrusion detection systems" IEEE International Conference on Computer and Information Technology, 2015, pp.1749-1755

[5] C. Saddi, and H. Chaoui, "Intrusion Detection System based Interaction on Mobile Agents and Clust-density Algorithm "IDS-AM-Clust", IEEE International Colloquium on Information Science and Technology, 2016, pp. 681-684

[6] D. Correa, F. Enembreck, and C. Silla, "An Investigation of the Hoeffding Adaptive Tree for the Problem of Network Intrusion Detection," IEEE International Conference on Neural Networks, 2017, pp. 4065-4072

[7] C. Ioannou, V. Vassiliou, and C. Sergiou, "An Intrusion Detection System for Wireless Sensor Networks," IEEE International Conference on Telecommunications, 2017, pp. 1-5

[8] M. Al-Rawi, et al, "Data Redundancy May Lead to Unreliable Intrusion Detection Systems," IEEE International Conference on Wireless Communication and Mobile Computing, 2017, pp. 1897-1902

[9] U. Rao, N. Singh, A. Amin, and K. Sahu, "Enhancing Detection Rate in Database Intrusion Detection System," IEEE International Conference on Science and Information, 2014, pp. 556-563

[10] M. Doroudian, and H. Shahriari, "Database Intrusion Detection System for Detecting Malicious Behaviors in Transaction and Inter-Transaction Levels", IEEE International Symposium on Telecommunication, 2014, pp.809-814

[11] V.Singh, and S.Puthran, "Intrusion Detection System Using Data Mining: A Review," IEEE Conference on Global Trends in Signal Processing, Information Computing and Communication, 2016, pp. 587 - 592

[12] A.Landress , "A Hybrid Approach to Reducing the False Positive Rate in Unsupervised Machine Learning Intrusion Detection" IEEE Conference on SoutheastCon, 2016, pp.1-6

[13] N. Gupta, K. Srivastava, and A. Sharma, "Reducing False Positive in Intrusion Detection System: A Survey," International Journal of Computer Science and Information Technologies, vol. 7, no. 3, 2016, pp.1600-1603

[14] S. Safdar, M. Hassan, M. Qureshi, and R. Akbar, "Data Hibernation Framework in Workflows under Intrusion Threat," IEEE Symposium on Computers and Informatics, 2011, pp.680-685

[15] K. Marin, and C. Doungsa "Test Data Generation From Hibernate Constraints," IEEE International Conference on Software Knowledge, Information Management and Application, 2014, pp. 1-6

[16] R. Ramachandran, P. Arya, and P.G. Jayanthy, "A Novel Method for Intrusion Detection in Relational Databases," IEEE International Conference on Advances in Computing, Communication, and Informatics, 2017, pp.230-235