International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Online Detection of Malicious Transactions from Database System

Dhanashree Parchand[1], Harmeet Kaur Khanuja[2]

[1, 2]Department of Computer Engineering, MMCOE, Karvenagar, Pune, Maharashtra, India

Abstract: ***As today's era mostly concentrated on online activities like net banking, e-transaction, security and privacy issues are at the peak with respect to their significance. Every organization is related with vast amount of information which is valuable. As database contains huge amount of information; data should be consistent, accurate and correct. Large numbers of database security breaches are occurring at a very high rate on daily basis. Today many approaches are used to protect the data as well as networks from attackers (attacks like SQLIA, Brute-force attack). Intrusion Detection System (IDS) is a way to make data more secure. Many researchers are concentrated on networks and operating system in this intrusion detection field. Proposed approach is for database so that it will prevent the data loss, maintain consistency and accuracy. Database security research is concerned about the protection of database from unauthorized access and malicious behavior. The unauthorized access is of many types; it may be in the form of execution of malicious transaction and this may lead to break the integrity of the system. Banking is the sector which is affected by this unauthorized activities and malicious transactions. So, it is today's need to detect all this malicious transactions and also to provide some recommendation. In this paper, we proposed a system here to detect online malicious transactions. We also aim to reduce the future attacks and to detect major database attacks. In order to detect malicious transactions, we used data mining algorithm for framing a data dependency miner for our banking database IDS. Our methodology extracts the read-write dependency rules from the normal transactions and then these rules are used to check whether the coming transaction is malicious or not. Our system overcomes some drawbacks of existing system like it finds the malicious transactions that corrupt data items and also identifies the transactions that write data without permission and read data without permission. Eventually, we are pointing out the challenges in the field of database security and how these challenges can be used as opportunities to stimulate the area of database privacy and security. In this detection, we are using data mining algorithm ODADRM for designing a data dependency miner for our database intrusion detection system. ODADRM extracts read-write dependency rules and it tracks normal transaction and also detects malicious ones more effectively than existing approaches.***

Keywords: Malicious Transaction, Data Mining, Data Dependency, Intrusion Detection System.

1. Introduction

As every system is depends upon database; information is called as main asset of any organization which is essential to its continuity. So it is very important to secure database system from any malicious activity. Therefore, information security is very important to protect the confidentiality, integrity and availability of the information. Many systems and tools are used to achieve the needs of the information security and to prevent database systems from any possible incident. Some data stored in databases is worth millions of dollars, but existing security models are not sufficient to prevent the misuse, especially insider abuse by legitimate users [3]. Therefore, a growing number of researches have concentrated on handling the vast variety of malicious attacks to theses data. Data is corrupted when malicious attacks are performed by attacker and this data can be spread across the network very fast through the authenticated users. Especially the robustness of a system is challenged by some irresponsible people in the banking system. So the banking organizations need to be able to detect the intrusions into their network and database systems as early as possible so that they can prevent further damage to sensitive data.

Intrusion is nothing but set of action that attempts to compromise the integrity, confidentiality or availability of resource. Database intrusion detection is of two types:

1) Signature Based Approach: In this, it matches number of signatures of online attacks and tracks the malicious transaction [1].
2) Non-Signature Based Approach: In this approach, IDS profiles patterns of normal user transaction and uses these patterns for identifying intrusive behavior [1]. This approach is also called as anomaly detection approach. Our system is under this non-signature based approach.

Our proposed system first extracts only interesting read and write dependency rules from all the normal transactions and these extracted rules are used to check whether the incoming transaction is malicious or not. This system not only extracts dependency between data items, but also considers the access operations of these data items (read or write) in extracting rules. The extracted rules are used to verify the malicious activity.

1.2 Motivation

From real-world database applications; it is observed that the whole database structure and essential data correlation rarely changes although the transaction program changes often.

So, the use of this data dependency among data items for detecting malicious transactions in database is introduced. In now days, database system plays an important role in many organizations. Database size increases continuously because of different reasons like the number of records, or the number of fields or objects in the database and attributes per object [4]. As a result, an administrator faces difficulties to keep the track of whether the attributes are being accessed only by genuine transactions or it is done by some unauthorized

users. All data is valuable specially in banking database and so it should be protected from any of the attacks. SQL Injection Attack is nothing but a technique where the structure of the query is changed by the attacker by injecting some input to the query formed by the programmer and gaining the access of database which results deletion or modification of user's data. Existing data mining based IDSs uses support-confidence framework for extracting dependencies among data items. The major drawback of this existing system is that the accuracy in setting up minimum support directly affects the number and the quality of the discovered classification rules and it also concentrate only on frequent item [1]. So to overcome these drawbacks, our system extracts only the interesting rules that have lower frequency than minimum support threshold. And this system also manages the number of extracted rules and avoids extracting uninteresting rules as we are interested in only few rules which are based on both access type and order of the data items.

2. Literature Review

Intrusion is any set of actions that attempt to compromise the confidentiality, availability or integrity of a resource [1]. Intrusion Detection Systems determines the illegal actions performed against computer systems and alerting the system administrator. The existing IDSs are of two types: signature-based and anomaly based systems. In signature-based approach, security experts constructs a set of signature manually by analyzing previous attacks and it matches the collected signatures against currently occurring activities to detect intrusions. Clearly signature-based IDS are not able to detect unknown attacks without any pre-collected signatures as it totally depends upon previous attack. In 2004, Hu and Panda proposed a data mining approach where they implemented a system to detect intrusive transactions that are targeted at corrupting data [3]. They have used a data mining approach for mining dependencies among data items, which are in the form of some classification rules [3]. In 2008, Hashemi proposed an approach based on mining data dependencies among data items, finding abnormal update patterns in the specific time series related to each data item's update history, and using a behavior similarity criterion between normal transactions and the incoming transaction [4]. This system has mainly three advantages (1) dependency rules among data items are extended for detecting transactions that read data without permission and for detecting transactions that write data without permission, (2) a novel behavior similarity criterion is introduced to reduce the FPR of the detection, (3) it conducts time-series anomaly analysis to detect intrusion transactions, which modify data items with some unexpected pattern [1]. In 2010, Hu and Panda proposed inter-transaction data dependencies approach for database intrusion detection; they have considered that the malicious transactions are launched by the attacker and can be so well crafted that an attacker may launch a group of malicious transactions which appears as a normal transaction [2]. They have found that it is difficult to design an algorithm that clusters user transactions into user tasks to discover the inter-transaction data dependencies. It improves the approach offered by Hu and Panda in 2004 [2].

Author and year	Technique	Features
Chung et al. (2000)	Access patterns of users	Generates profiles of users by using audit logs and detects insider abuse with the derived profiles
Lee at al. (2000)	Tagging time signatures to data items	Suitable for Real-time databases.
Barbara et al. (2002)	Building database behavioural models by HMM	Recognizes malicious patterns by using HMM.
Hu & Panda (2004)	Mining data dependencies among data items	It detects malicious transactions targeted at corrupting data. It's less sensitive to the change of user behaviours and database transactions.
Bertino et al. (2005)	Access patterns of users	Under an RBAC system: Suitable for large databases. Can be deployed very easily in practice.
Srivastava et al. (2006)	Weighted data dependency rule miner (WDDRM)	It improves the approach offered by Hu & Panda (2004). It generates rules for possibly less frequent but more important attributes by considering the sensitivity of the attributes.
Kamra et al. (2008)	Access patterns of users	They have developed three models, of different granularity, to represent the SQL queries appearing in the database log files.
Hashemi et al. (2008)	Dependencies among data items and time-series anomaly analysis	It improves the approach offered by Hu & Panda (2004). The concept of malicious transactions is extended from those that corrupt data to those that either read data or write data or both without permission.
Hu & Panda (2010)	Mining Inter-transaction Data Dependencies	It improves the approach offered by Hu & Panda (2004). They cluster legitimate user transactions into user tasks for discovery of inter-transaction data dependencies.
M. Sohrabi et al. (2014)	Optimal Data Access Dependency Rule Mining	It avoids many limitations of previous data dependency miner algorithm. It is extension of k-optimal rule discovery algorithm. Uses leverage as key measure value to extract dependency rules.

Figure 1: Comparison of Existing System

3. Problem Statement

Data security is an important issue in today's era. IDS help to secure the information in the database system. We propose a system which will be able to detect malicious transaction more effectively than previous approaches. It detects malicious transactions that corrupt the data. The aim of this system is to design a intrusion detection system using dependency relationship among data items to detect and prevent the malicious activities in database management systems. It will extract the interesting read-write dependency rules although it is not frequent item [1] and according to the rules formed by normal transaction it is able to decide whether the transaction is malicious or not. Based on all this process it determines the type of attack and it will provide some preventive measures to reduce or avoid the future malicious activities.

So the specific objectives are-
1. To detect malicious transaction in database system.
2. To prevent data items.
3. To avoid future malicious transaction or attack.
4. To detect intrusion at the runtime.

4. Existing Systems

Association rule mining is one of the most popular data mining techniques for extracting knowledge from data [1]. In this, support and confidence these terms are used to measure

the quality of a given rule. Support tells number of transaction from dataset that was used to generate the rule. Confidence tells number of transaction that includes items from both the sides. It is apriori based approach which is used to identify frequent item set. The input to this system is log of normal transaction and then according to minimum support it mines the frequent sequential patterns. It also generates read-write sequence set.

Limitation of Existing System: Frequent association rule mining based database IDS uses apriori based approach which totally prefers frequent data item set. It ignores sequences that have frequency lower than minimum support threshold and so it doesn't produced sequences of data items that are infrequent and of great interest. And the appropriate setting for minimum support, minimum confidence is necessary otherwise system is not feasible.

5. Proposed System

Database IDS makes list of all patterns of normal user transactions and uses these patterns to identify intrusive behavior, such that the transactions not compliant with the patterns are identified as malicious. By mining the data dependencies among data items in database systems; database IDS obtains patterns of normal transactions. The data dependency miner component discovered data dependencies and are employed as rules for identifying anomalies. Data dependency miner component plays very important role in data dependency based IDSs [1]. In fact the accuracy of a data dependency based IDS is depends on its data dependency miner component because the extracted rules are the main criterion for detecting the new malicious transactions. The input to the system is a log file of normal transactions and it generates read and writes dependency rules from the log file.

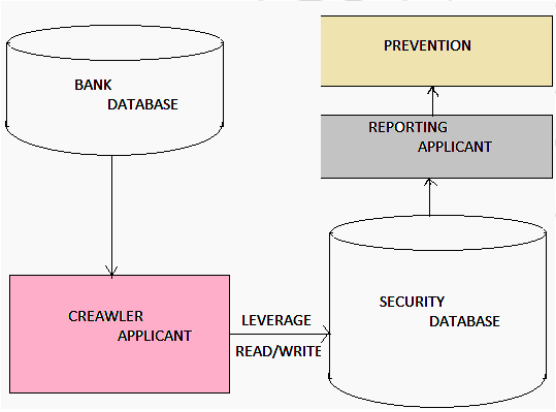


Figure 2: System Architecture

In the above block diagram, there are five blocks which represent the modules of our architecture. This diagram shows the stepwise working of our system. As this system detects intrusion transaction in the database so there must be some huge amount of database.

1) Bank Database:
Banking database contains all the information about transactions like date, time, Mac-id, transaction-id, account-no etc… And also contains all data about employees, customer, accounts, account-types etc... This contains huge amount of valuable and secret data about bank. It maintains the data according to read, write transactions.

2) Crawler Application
1. It reads each transaction from the database.
2. It categorizes the transaction in the form of Read or Write.
3. It generates Read sequence.
4. It also generates Write sequence.
5. It calculates key measure value that is Leverage value.
6. According to leverage value all transaction which shows abnormal behavior is transferred to security database.

Based on a dependency factor we generate read and write operations to create the normal transactions. The dependency factor is defined as the average numbers of read operations immediately before a particular write operation w(x) or the average number of write operations immediately after that write operation.

3) Security Database
According to data dependency and calculated leverage value the transaction which shows intrusive behaviors are stored into this database. The transactions in this database are not showing normal behavior as it may be malicious ones so we have to analyze it.

4) Reporting Application
The report is generated on the basis of date and category when we found the malicious transaction. According to the date and time one record is generated and it also gives Mac-id (IP address) of that attacker. Database administrator takes one look on it and analyzes it to detect malicious transaction (attacker) and type of attack. This also generates report on the basis of category like SQLIA, Brute-Force attack.

5) Prevention
Admin can prevent this type of attack after analyzing that report by giving some preventive measures like we can block that attacker as we know the Mac-id of the system to prevent any future attacks.

In this way our system can find online malicious transaction and can prevent data items. This system is most reliable and real-time system. It is faster than any other system. It prevents the malicious transactions. It can be a mobile application.

6. Implementation Details

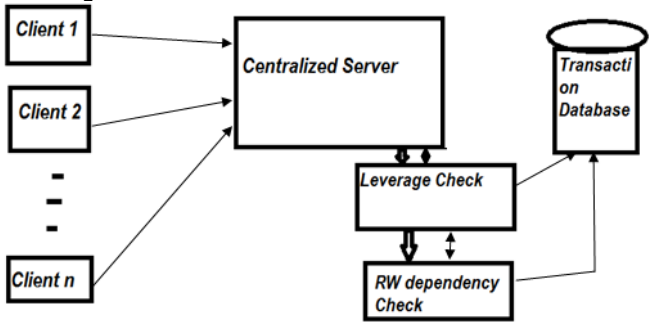


Figure 3: Client-Server Architecture

Above diagram shows the simple client server architecture of our online intrusion detection system.

Algorithm and Analysis

PART 1. Read Dependency Rules:
- Add all data items with read operation to AvailableLHS vector.
- Add all data items with write operation to AvailableRHS vector.
- ODADRM with CurrentLHS, AvailableLHS, AvailableRHS, IsReadRule values.

PART 2. Write Dependency Rules:
- Add all data items with write operation to AvailableLHS vector.
- Add all data items with write operation to AvailableRHS vector.
- ODADRM with CurrentLHS, AvailableLHS, AvailableRHS, IsWriteRule values.

PART 3. ODADRM (CurrentLHS, AvailableLHS, AvailableRHS, RuleType)
- Initialize CurrentSolution to empty value.
- Initialize SoFar to empty.
- Loop through each value in AvailableLHS

If Coverage (P) > Min(Coverage) then consider CurrentLHS as NewLHS with addition of P
- If the size of NewLHS is equal to MaxLHSSize and Coverage(NewLHS)*(1-Coverage(NewLHS)) is less than Minimum Coverage then no need to access data to check NewLHS -> Q is in solution or not.
- ODADRM loops through each condition which has write access operation in NewAvailableRHS and If Coverage(NewAvailableRHS(Q))is greater than (1-(MinLeverage/Coverage(NewLHS))) then it removes Q from NewAvailableRHS.
- Otherwise, it checks whether NewLHS -> Q might be in solution.
- If the size of NewLHS is equal to MaxLHSSize and Coverage(NewLHS)is more than (1-(MinLeverage/Coverage(NewLHS))) then no need to access data to check NewLHS-> Q is in solution or not.
- Leverage(NewLHS -> Q); it checks whether this rule is good to be in solution or not.
- It finds the MinLeverage value from CurrentSolution and then it compare Leverage (NewLHS -> Q) to Leverage of Current Rule.
- If Leverage (NewLHS -> Q) > Leverage (CurrentRule) then (NewLHS -> Q)is replaced by Current Rule which has MinLeverage value.
- ODADRM is recursively called to extract the read and write dependency rules.

Above algorithm is divided into two parts-
1. Extracting Read Dependency Rules.
2. Extracting Write Dependency Rules.

And ODADRM is called by both the parts. ODADRM not only considers the data item but also concentrate on access operations because extracted dependency rules in database intrusion detection field are based on the correlation between access operation of data items. ODADRM algorithm prunes the search space based on leverage value and upper limit on number of rules, k. So ODADRM results in an appropriate number of optimal read-write dependency rules for computation to be feasible [1].

Firstly we are giving normal transactions as the input to ODADRM algorithm to form rules. Then these extracted rules are used to identify the current transaction is malicious or not that means the coming transaction is compared to these rules if it does not follow the extracted rules then that transaction is malicious one.

For the database, one module is there in which we can insert any number of entries of transactions. We are using client server application so it can be used in mobile and desktop or wherever there is internet connection.
We are using two filters:
1. One for leverage value.
2. One for read write dependency.

Firstly we can add or enter any number of entries in the database.

Then our module forms the read sequence, write sequence, read-write sequence according to transactions.
Then it will find the leverage value with the help of formulae:
Leverage(X->Y)= P(X&Y)- P(X).P(Y)

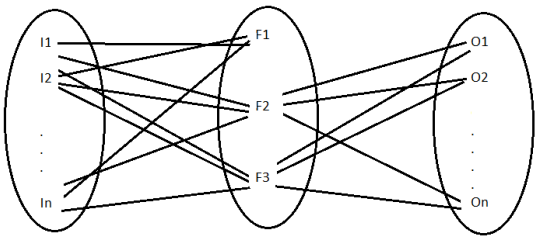
We are using one filter over here to find the transactions which have leverage value greater than our range. Then we are finding malicious transactions from SQLlogs. Then we are using our second filter over here for read-write dependencies...

For the execution of this project we are assuming some conditions if that condition satisfied then this problem is solvable. So, this problem is satisfiable if and only if the conditions are fulfilled.
That means, given problem is NP-Complete.

7. Mathematical Model

S={I, F, O}
Where I – Input log of transactions.
F- Set of functions applies to input dataset.
O- Detected malicious transactions.

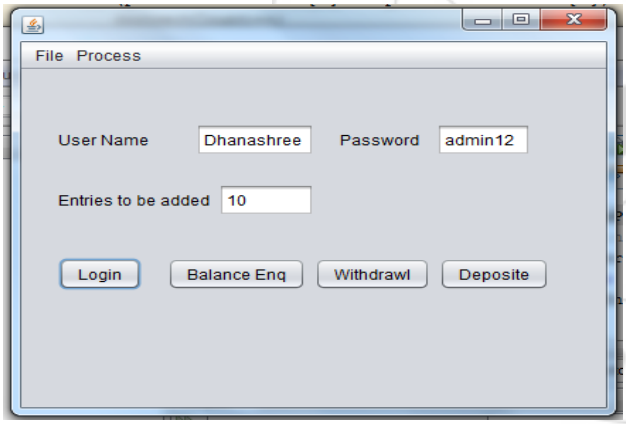
I={I1,I2,.....In}
F={F1,F2,F3}
F1=Apply ODADRM algorithm.
F2=Report generation.
F3=Prevention scheme.
O={O1, O2, ...On}

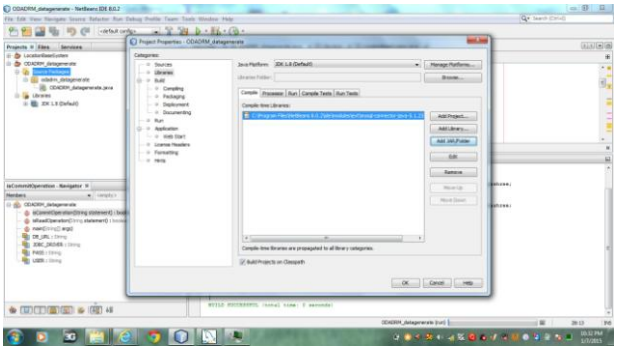


- Leverage$(X \rightarrow Y) = P(X \text{ and } Y) - (P(X) \times P(Y))$
  P(X and Y) is equal to support(X →Y, D)
- Support(X →Y, D) – no. of records that interaction of (X → Y) involves.
  If we assume X and Y are independent, the probability of X and Y is P(X) × P(Y), i.e., cover(X, D) × cover(Y, D),
- Cover(X, D) × cover(Y, D) -- product of the no. of records that involve X and no. of records that involve Y.
  Therefore leverage is as follows
- Leverage$(X \rightarrow Y,D) = cover(X,D) - confidence(X \rightarrow Y,D) \times cover(Y,D)$

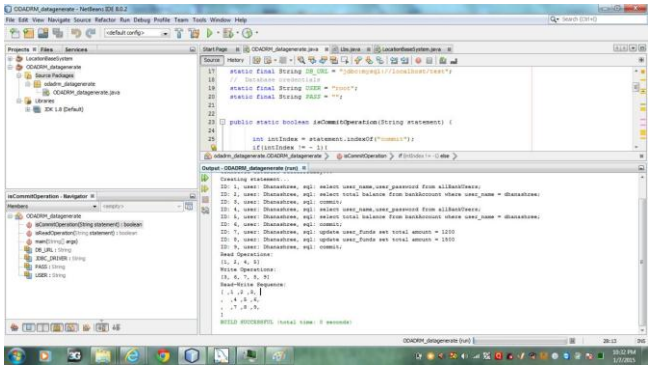
8. Results

Our work is totally focused on development of database intrusion detection system. According to [1], M. Sohrabi et al. compared two different approaches i.e. Hu and Panda method and ODADRM algorithm. They found ODADRM is better approach with respect to complexity, time, and accuracy. First screenshot is of user screen for database entries.

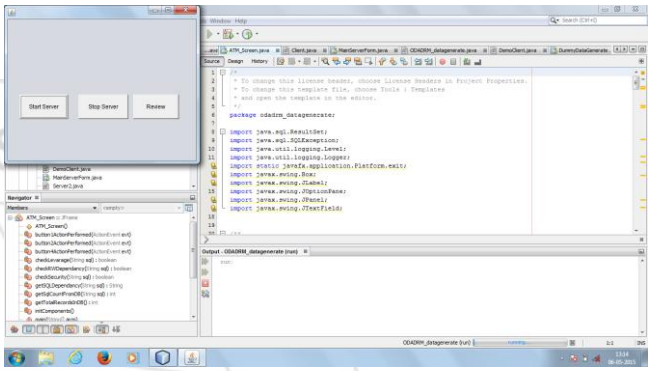




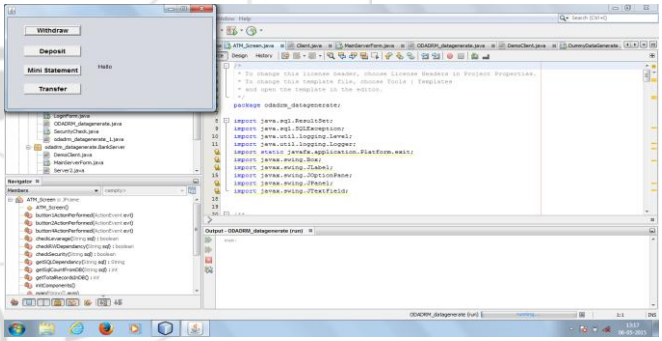
By finding the leverage value, we are getting all the read sequence, write sequence and read-write sequence.



As we are using client-server application for our system, we have to start server on the machine then only our system can be work properly.



It is like the ATM application so we can withdraw, deposit money from any account.



9. Conclusion

Malicious detection mechanisms play a crucial role in the security landscape of an organization. In this we have focused malicious detection system for database. Existing data mining based IDSs use support-confidence framework for extracting dependencies among data items [2]. The major limitation of these approaches is that the accuracy in setting up minimum support directly influences the number and the quality of the discovered rules. Our approach can extract interesting rules that have lower frequency than minimum support threshold. Also it manages the number of extracted rules and avoids extracting uninteresting rules. On the basis of these leverage value and dependency rules we can find the malicious transaction. There are many interesting issues to investigate as future work. For instance ODADRM only finds malicious transactions that corrupt data items and cannot identify transactions that read data without permission. This

results in a significant reduction in detection rate when most malicious transactions are only reading data items illegally. We also create report which shows all malicious transaction carried out and on the basis of that we give some preventive measures to avoid future attacks.

References

[1] Mina Sohrabi, M.M.Javidi, S.Hashemi "Detecting intrusion transactions in database systems: a novel approach" J Intell Inf Syst 42:619-644 DOI 10.1007 Springer 2014.
[2] Hu and Panda, B. "Mining inter-transaction data dependencies for database intrusion detection". In Proceedings of innovations and advances in computer sciences and engineering. Springer 2010.
[3] Hu and Panda, "A data mining approach for database intrusion detection." In Proceedings of the ACM symposium on applied computing (pp. 711716) ACM 2004.
[4] Hashemi, S., Yang, Y., Zabihzadeh, D., Kangavari, M. "Detecting intrusion transactions is databases using data item dependencies and anomaly analysis". Expert Systems, 25(5), 460473 2008.
[5] Lee, V.C., Stankovic, J., Son, S.H. "Intrusion detection in real-time database systems via time signatures." In Proceedings of the 6th IEEE real time technology and applications symposium (RTAS00) (pp.124133). New York: IEEE Press 2000.

Author Profile

Dhanashree Parchand is currently working towards her M.E. (Computer Engineering) degree at Savitribai Fule Pune University, India. She received her B.E. (Computer Engineering) degree from the RTMNU, Nagpur University, India in 2012. Her research interests include Computer Security, Data Mining and Networks, Data Security.

Harmeet Kaur Khanuja is currently working in Marathwada Mitra Mandal's College of Engineering, Pune as Head of the Computer Department. She received her B.E and M.tech degree from Pune University, India. She is now doing her Ph.D. from RTMNU, Nagpur University, India. Her research interests include Data Mining and Database Forensics, Mobile Computing.