Taxonomy of Various Database Security Technique And Their Analysis

Shivnandan Singh, Pratibha Singh, Neetu Kushwah

Abstract - **Security is an important issue in database management system because information stored in a database is very important , very sensitive commodity. So the data base management system need to should be protected from unauthorized access and updates. Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes It is popular belief that hackers cause most security breaches, but in reality 80% of data loss is to insiders. Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes. Database security is also a specialty within the broader discipline of computer security.**

INTRODUCTION

The objective of database security is to prevent undesired information disclosure and modification of data while ensuring the availability of the necessary service. Database Security has become an important issue in today's world. Organizations have become highly dependent on the database for their daily operations. With the increase in the use of World Wide Web in recent years emphasize the web database security. Data security has become a necessity for every individual who is connected to internet and uses the internet for any purpose. In this survey we are going to present different methods or frameworks explained in different papers for common problem of database security In this ever growing cyber world where millions and trillions of bytes of data is transferred everyday over the internet, the security of this data is a top priority and a major challenge. Data security has become a necessity for every individual who is connected to internet and uses the internet for any purpose.

Lost or stolen data, especially customer data, can result in brand damage, competitive disadvantage and serious fines. In high-profile cases, compromised data presents organizations with long-term customer acquisition and retention difficulties. For many organizations, databases are a treasure trove of sensitive information containing data ranging from customers' personal details and confidential competitive information to intellectual property .As a result, database security is a top priority for today's IT director. Yet, the shortcomings of many traditional database security techniques such as firewalls and application security have been exposed in recent years and it is now broadly recognized that these approaches to database security are no longer sufficient to protect businesses and data in today's modern, open and complex IT environment. In trying to mitigate the risk of security breaches and to comply with numerous existing and emerging regulations, database encryption is often seen as the solution

VARIOUS DATABASE SECURITY TECHNIQUES

1. Securing Database using Encryption

The basic idea behind encryption is to apply an encryption algorithm to the data, using a user-specified encryption key. The result of the algorithm is the encrypted data. There is also a decryption algorithm which takes the encrypted data and decryption key as input and returns the original data. . Why is encryption often heralded as the best defense against database security breaches and how can companies overcome the oft-cited challenges associated with its implementation. Both the algorithms i.e. Encryption and Decryption algorithm are known to the public but the keys either both or anyone is secret. In practice, the public sector uses database encryption to protect citizen privacy and national security.

Manuscript received May 22, 2014.

Shivnandan Singh, PG Scholar, Galgotia's University Greater Noida, UP, India.

Pratibha Singh, PG Scholar, Galgotia's University Greater Noida, UP, India.

Neetu Kushwah, PG Scholar, Galgotia's University Greater Noida, UP, India.

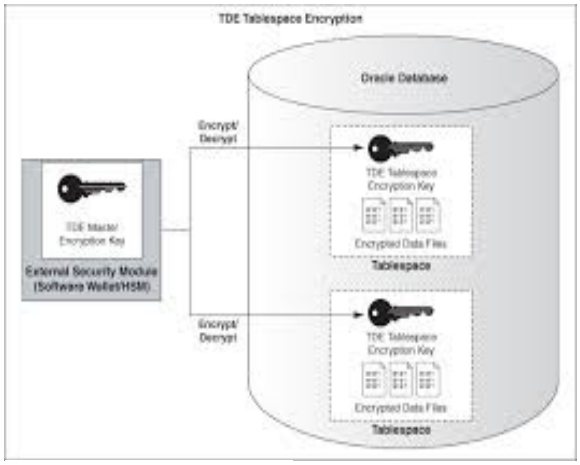

Figure-1

There are two types of encryption techniques, Symmetric encryption and Asymmetric encryption. In symmetric encryption, the encryption key is also used as the decryption key. It uses an encryption algorithm that consists of character substitutions and permutations. The main weakness of symmetric encryption is that all authorized users must be told the key, increasing the likelihood of its becoming known to an intruder. Another approach to encryption called public-key encryption has become increasingly popular in recent years. Each authorized user has a public encryption key, known to everyone, and a private decryption key, known only to oneself. Company databases manage the most sensitive enterprise data. Without a doubt, database encryption should be a priority for organizations intent on protecting this data. But encryption must also be accompanied by best practice key management in order to provide the highest levels of security. If companies follow this best practice, they will find that not only are they protecting their company's most sensitive information, but they are also assisting compliance with government and industry regulations and rules, helping to prevent data breaches and, crucially, protecting their corporate brand and reputation Although encrypted data is difficult to decipher, it is relatively easy to detect. Encryption only obscures a message's meaning, not its existence. Therefore, steganography, a technique that hides the existence of a message, is often used to supplement encryption. It's easy to use and works by replacing bits of unused data in computer files or communication channels, such as telephone lines or radio broadcasts.

2. Securing Database using Stenography

Steganography is beneficial for securely storing sensitive data, such as hiding system passwords or keys within other files. However, it can also pose serious problems because it's difficult to detect. Network surveillance and monitoring systems will not flag messages or files that contain stegano graphic data. Therefore, if someone attempted to steal confidential data, they could conceal it within another file and send it in an innocent looking email.

To detect hidden messages, an organization must actively monitor network traffic, which is time- and processor-intensive. However, those who are familiar with the network's normal traffic patterns can simply look for changes, such as increased movement of large images across the network, which may warrant further, detailed investigation. It's also wise to have -- and actively enforce -- a security policy that clearly outlines acceptable usage, what data types can and can't be sent across the network and how it should be protected. Also, restrict unauthorized programs, ban the use of unauthorized encryption and steganography in the workplace and consider limiting the size of mailboxes. Finally, consider determining whether employees who deal with confidential information should have access to large media files, particularly image, video or audio files that are to be posted on your Web site. Malicious parties could use steganography to pass information via such files to a third party with access to your site. Why not consider using steganography to your advantage by using digital watermarks, a form of steganography, to copyright your Web-accessible media files? You can even use it to hide system passwords or keys within other files to provide a more secure storage location.

3. Negative Database Technology

A negative database can be defined as a database that contains huge amount of data which consists of counterfeit data along with the actual data. A few approaches that describe this concept have been proposed but have not yet been implemented to work for real world databases. Our main objective is the validation of a benign user and rejection of a malicious user for a particular database. There have been cases where an attack to a database can take place by writing a query in the username and password field of the login page. For a normal database, this may work and the malicious user can get access to the database of the system. On the other hand, our framework prevents a malicious user form doing so. [6]

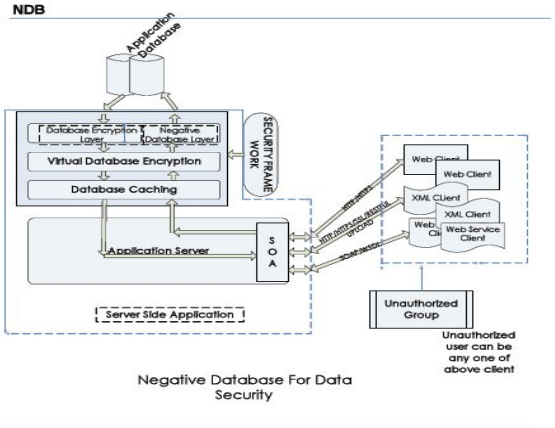

Figure:1.2

The database is always populated by data that is entered by the administrator and updated by a benign user, which could be a online banking or credit card company customer. A problematic situation may arise when a malicious user tries to update or modify the database. The example of such updates could be a SELECT query inside an INSERT query. The purpose of the encryption module along with other three modules is to provide utmost security to the data. In our framework
we use the public key encryption algorithm RSA. Strong encryption makes the database more secure and reliable. The encryption algorithm is preceded by two modules which are Database caching and Virtual Database Encryption algorithm. It takes the input from the previous module and applies RSA on the data. The encrypted data is passed through the Negative Database conversion algorithm as shown in Figure 3 to generate encrypted multi sets of data for a single true set of data, making the database hard to query. The data encrypted in this layer is passed to the next layer called the Negative Database Conversion layer.[3]

The concept of a negative database took shape a couple of years ago, while Dr Esponda was working at the University of New Mexico with Paul Helman, another computer scientist, and Stephanie Forrest, an expert on modelling the human immune system. The important qualification concerns that word "everything". In practice, that means everything in a particular set of things.[5]

4. Web-based Database Security

Various high-profile hacking attacks have proven that web security remains the most critical issue to any business that conducts its operations online. Web servers are one of the most targeted public faces of an organization, because of the sensitive data they usually host. Securing a web server is as important as securing the website or web application itself and the network around it. If you have a secure web application and an insecure web server, or vice versa, it still puts your business at a huge risk

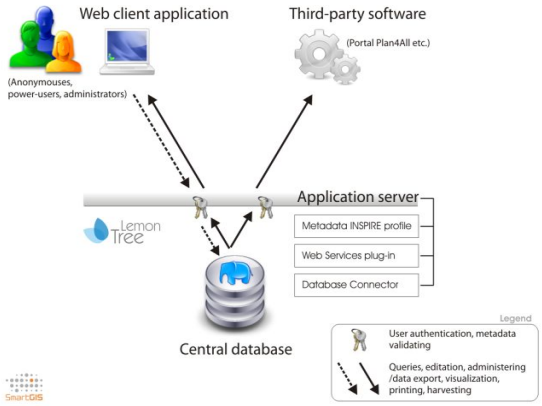

Figure:1.3

4.1 Remove Unnecessary Services

Default operating system installations and configurations, are not secure. In a typical default installation, many network services which won't be used in a web server configuration are installed, such as remote registry services, print server service, RAS etc

4.2 Remote access

Using security tokens and other single sign on equipment and software, is a very good security practice. Remote access should also be restricted to a specific number of IP's and to specific accounts only. It is also very important not to use public computers or public networks to access corporate servers remotely, such as in internet café's or public wireless networks.

4.3 Separate development production environment

Since it is easier and faster for a developer to develop a newer version of a web application on a production server, it is quite common that development and testing of web applications are done directly on the production servers itself. Such applications could easily be discovered and exploited by a malicious user, by using free available tools on the internet.

4.4 Web application content and server-side scripting

The web application or website files and scripts should always be on a separate partition or drive other than that of the operating system, logs and any other system files. Through experience we've learnt that hackers who gained access to the web root directory, were able to exploit other vulnerabilities, and were able to go a step further and escalate their privileges to gain access to the data on the whole disc, including the operating system and other system files. From there onwards, the malicious users have access to execute any operating system command, resulting in complete control of the web server.

4.5 Privileges and Permissions:

File and network services permissions play a vital role in web server security. If a web server engine is compromised via network service software, the malicious user can use the account on which the network service is running to carry out tasks, such as execute specific files. Therefore it is very important to always assign the least privileges needed for a specific network service to run, such as web server software. It is also very important to assign minimum privileges to the anonymous user which is needed to access the website, web application files and also backend data and databases.

4.6 Install all security patches on time

Although having fully patched software does not necessarily mean your server is fully secure, it is still very important to update your operating system and any other software running on it with the latest security patches. Up untill this day, hacking incidents still occur because hackers took advantage and exploited un-patched servers and software.

4.7 Monitor and audit the server

All the logs present in a web server, should ideally be stored in a segregated area. All network services logs, website access logs, database server logs (e.g. Microsoft SQL Server, MySQL, Oracle) and operating system logs should be monitored and checked frequently. If one notices strange activity from the logs, this should immediately be escalated so the issue can be investigated to see what is happening.

4.8 User accounts

Unused default user accounts created during an operating system install should be disabled. There is also a long list of software that when installed, user accounts are created on the operating system. Such accounts should also be checked properly and permissions need to be changed required It is also a good security practice not to share each others' user accounts.

4.9 Remove all unused modules and application extensions

A default Apache installation has a number of pre-defined modules enabled, which in a typical web server scenario are not used, unless they are specifically needed. Turn off such modules to prevent targeted attacks against such modules.

The same applies for Microsoft's web server; Internet Information Services. By default, IIS is configured to serve a large number of application types, e.g. ASP, ASP.NET and more. The list of application extensions should only contain a list of extensions the website or web application will be using. Every application extension should also be restricted to use specific HTTP verbs only, where possible.

4.10 Use security tools for web servers

Microsoft released a number of tools to help administrators secure IIS web server installations, such as URL scan. There is also a module called mod security for Apache. Although configuring such tools is a tedious process and can be time consuming, especially with custom web applications, they do add an extra bit of security and piece of mind.

5. Security in Real-time Database Systems

Real-time database systems [13,14], such as military command control, communication, avionics, radar tracking, and managing automated factories, have timing constraints. These timing constraints are typically in the form of deadlines that require a transaction to be completed by a specified time. Hence, it is necessary to provide database security for real-time databases. Conventional database security models are not adequate for time-critical applications [15]. A model for database security in real- time databases must address satisfying the timing constraints and security constraints. Major efforts to design secure databases have focused on databases without timing constraints. It is only recently that work [12,15,16] has been reported in the area of database security for real-time databases. The objective of this work is to study the factors involved in secure concurrency control, develop a suitable concurrency control algorithm, implement and study the performance of the algorithms with a real-time database system simulation. Many real-time database systems are contained in such environments, and, hence, are considered for this study. Mandatory access control is applicable on large amounts of information that requires firm protection in an environment where the subjects are grouped into clearances and objects are grouped by their classifications. In a multilevel secure database model objects have security levels and subjects have clearance levels. According to the Bell-LaPadula simple property [17], a subject can read a certain data object if the subject's clearance level dominates the object's security level. According to the Bell-LaPadula star-property [17], a subject can write to an object if the subject's clearance level is dominated by the object's security level However, the concurrent execution of transactions results in contention for data objects, and as a result, it is possible to have an indirect flow of information from objects at higher levels to subjects at lower levels due to a covert channel [13].

A secure real-time database system has to simultaneously satisfy two goals, provide security and ensure that the deadline miss percentage is minimized. In many occasions, these two goals conflict and to achieve one goal is to sacrifice the other. For example, suppose we have a high priority transaction at a high security level and a low priority transaction at a low

security level, and there is a data conflict between them. (Priority is based on deadline, where a higher priority means an earlier deadline.) If the high priority transaction gets the data and blocks the low security transaction, then although the priority is maintained, a covert channel is introduced. The lower security level can determine there are transactions at higher levels and may even be able to receive information from the length of the delay. If the reverse is allowed to happen, security will be maintained but the priority will be violated. Whether to maintain security or priority is dependent upon the system. If the system requires that security be maintained regardless of the deadline of the conflicting transactions, then conflict must be resolved in favor of security. If the system allows a compromise between security and priority, then the goal is to maintain as much security as possible without raising the deadline miss percentage significantly.

6. Detecting SQL Injection

In order to protect a Web application from SQL Injection attacks, there are two major concerns. Firstly, there is a great need of a mechanism to detect and exactly identify SQL Injection attacks. Secondly, knowledge of SQL Injection Vulnerabilities (SQLIVs) is a must for securing a Web application. So far, many frameworks have been used and/or suggested to detect SQLIVs in Web applications. Here, we mention the prominent solutions and their working methods in brief to let the readers know about the core ideas behind each work.

6.1 SAFELI

Fu et al., in [18] propose a Static Analysis Framework in order to detect SQL Injection Vulnerabilities. SAFELI framework aims at identifying the SQL Injection attacks during the compile-time. This static analysis tool has two main advantages. Firstly, it does a White-box Static Analysis and secondly, it uses a Hybrid-Constraint Solver. For the White- box Static Analysis, the proposed approach considers the byte-code and deals mainly with strings. For the Hybrid- Constraint Solver, the method implements an efficient string analysis tool which is able to deal with Boolean, integer and string variables.

The methodology is an efficient approximation mechanism to deal with string constraints. However, the approach is only dedicated to ASP.NET vulnerabilities.

6.2 Thomas et al.'s Scheme

Thomas et al., in [19] suggest an automated prepared statement generation algorithm to remove SQL Injection Vulnerabilities (SQLIVs). They implement their research work using four open source projects namely: (i) Net-trust, (ii) ITrust, (iii) WebGoat, and (iv) Roller. Based on the experimental results, their prepared statement code was able to successfully replace 94% of the SQLIVs in four open source projects. However, the experiment was conducted using only Java with a limited number of projects. Hence, the wide application of the same approach and tool for different settings still remains an open research issue to investigate.

6.3 Ruse et al.'s Approach

Claimed and apparent efficiency of the technique, the major drawback of the work is that it was not tested with real queries on a real-life existing database.

6.4 Haixia and Zhihong's Database Security Testing Scheme

In [22], Haixia and Zhihong propose a secure database testing design for Web applications. They suggest a few things; firstly, detection of potential input points of SQL Injection; secondly, generation of test cases automatically, then finally finding the database vulnerability by running the test cases to make a simulation attack to an application. The proposed methodology is shown to be efficient as it was able to detect the input points of SQL Injection exactly and on time as the authors expected. However, after analyzing the scheme, we find that the approach is not a complete solution but rather it needs additional improvements in two main aspects: the detection capability and the development of the attack rule library.

6.5 Roichman and Gudes's Fine-grained Access ControlScheme

In [20], Ruse et al. propose a technique that uses automatic test case generation to detect SQL Injection Vulnerabilities. The main idea behind this framework is based on creating a specific model that deals with SQL queries automatically. In addition, the approach identifies the relationship (dependency) between sub-queries. Based on the results, the methodology is shown to be able to specifically identify the causal set and obtain 85% and 69% reduction respectively while experimenting on few sample examples. Moreover, it does not produce any false positive or false negative and it is able to detect the real cause of the injection. In spite of the

In [23], Roichman and Gudes, in order to secure Web application databases, suggest using a fine-grained access control to Web databases. They develop a new method based on fine-grained access control mechanism. The access to the database is supervised

and monitored by the built-in database access control. This approach is efficient in the fact that the security and access control of the database is transferred from the application layer to the database layer.

References

[1] Korth Henry F., Silberschatz Avi, Sudarshan, Database System Concepts 5th Edition.

[2] Stallings William Cryptography and Network security.

[3] Cryptographic Checksum for Multilevel Database Security.US Army of Airforce 1987.

[4] Popa Raluca, Catherine M. S., Zeldovic Nickolai, Balakrishnan Hari 2011. CryptDB: A practical encrypted relational DBMS.

[5] Popa Raluca, Catherine M. S, Zeldovich Nickolai, Balakrishnan. October 2011 CryptDB:Protecting Confidentiality with Encrypted Query Processing.

[6] Marten van Dijk, Gentry Craig, HaleviShai ,Vaikun tanathanVinod December 11, 2009 Fully Homomorphic Encryption over the integer .

[7] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, C-35:677– 691, 1986 .

[8] Patel, A., Sharma, N. (2008). "Negative Database for Data Security", *Master 's Project Report* , Computer Engineering Department, San Jose State university.

[9] Robert Vinson – IT Security Analyst The University of Iowa .

[10] Esponda,F.(2005).Negative Representations of Information, *Ph.D.* Dissertation,The University of New Mexico.

[11] Hasan Kadhem, Toshiyuki Amagasa, HiroyukiKi tagawa , "A Novel Framework for Database Security based on Mixed Cryptography",Fourth Int ernational Conference on Internet. 24-28 May 2009 , pp.163-170

[12] George, B. and J. Haritsa, "Secure Transaction Processing in Firm Real-Time Database Systems,"Proceedings SIGMOD, Phoenix, AZ, 1997, pp. 462-473.

[13] Ozsoyoglu, Gultekin, and Richard T. Snodgrass, "Temporal and Real-Time Databases: A Survey," IEEE Transactions on Knowledge and Data Engineering, vol. 7 no. 4, Aug. 1995, pp. 513-532.

[14] Ramamritham, Krithi, "Real-Time Databases," International Journal of Distributed and Parallel Databases, 1993, pp. 199-226.

[15] Rasikan, David, Sang H. Son and Ravi Mukkamala, "Supporting Security requirements in Multilevel Real- Time Databases," Proceedings IEEE Symposium on Security and Privacy, Oakland, CA, May 1995, pp. 199- 210.

[16] Son, Sang H., Rasikan David and Bhavani Thuraisingham, "An Adaptive Policy for Improved Timeliness in Secure Database Systems," Proceedings of Annual IFIP WG 11.3 Conference of Database Security, Aug. 1995.

[17] Bell D. E., and L. J. LaPadula, " Secure Computer Systems: Mathematical Foundations and Model," Technical Report, MITRE Corporation, 1974.

[18] Fu, X., Lu, X., Peltsverger, B., Chen, S., Qian, K., and Tao, L., A Static Analysis Framework for Detecting SQL Injection Vulnerabilities. Proc. 31st Annual International Computer Software and Applications Conference 2007 (COMPSAC 2007), 24-27 July (2007), pp. 87-96.

[19] Thomas, S., Williams, L., and Xie, T., On automated prepared statement generation to remove SQL injection vulnerabilities. Information and Software Technology, Volume 51 Issue 3, March 2009, pp. 589–598.

[20] Ruse, M., Sarkar, T., and Basu. S., Analysis & Detection of SQL Injection Vulnerabilities via Automatic Test Case Generation of Programs. Proc. 10th Annual International Symposium on Applications and the Internet, 2010, pp. 31-37.

[21] Junjin, M., An Approach for SQL Injection Vulnerability Detection. Proc. of the 6th International Conference on Information Technology: New Generations, Las Vegas, Nevada, April 2009, pp. 1411- 1414.

[22] Haixia, Y. and Zhihong, N., A database security testing scheme of web application. Proc. of 4th International Conference on Computer Science & Education 2009 (ICCSE '09), 25-28 July 2009, pp. 953-955.

[23] Roichman, A., Gudes, E., Fine-grained Access Control to Web Databases. Proceedings of 12th SACMAT Symposium, France 2007.