Security in Data Warehousing

Harsh Sorathiya, Apurva Patel, Harsh Jain, Aahana Khajanchi, Jayesh Surana
Information Technology, SVITS
Gram barol, Sawer road, Indore, MP, India

***Abstract*— As Data Warehouse store huge amount of data, the security of this huge information base is crucial for the sustainability and reliability of data warehouse. Since its advent the data warehouse has gone through various technological changes, which has prompted us to change the security strategies as well. Thus, taking a deep look at the various changes in the security mechanisms of data warehouse, along with the changes in the strategies for the data warehouse development it helps in understanding the various security aspects which are related to the data warehouse.**

I. INTRODUCTION

Providing access to multiple, distributed, heterogeneous databases and other information sources has become one of the leading issues in database research and industry. In the research works, most approaches to the data integration problem are based on the following very general two-step process, they are:

- Accept a query, determine the appropriate set of information to answer the query, and generate the appropriate commands for each information source.
- Obtain results from the information sources, perform appropriate translation, filtering, merging and other things of the information, and return the final answer to the user or application.

We usually refer to this process as a lazy or on-demand approach to data integration, since information is extracted only from the sources only when different queries are posed.
The natural alternative to a lazy approach is an eager or in-advance approach to data integration. Apart from these, In an eager approach:

- Information from each source that may interest us is extracted in advance before moving further, translated and filtered as appropriate, merged with relevant information from other sources, and stored in a (logically) centralized repository.
- When a query is posed, the query is evaluated directly at the repository, without accessing the original information sources.

This approach is commonly referred to as data warehousing, as the repository that serves as a warehouse, stores the data of interest.

A rather lazy approach to integration is appropriate for information that changes quickly with respect to time, for clients with unpredictable needs, and for queries that operate over huge amounts of data from a large numbers of information sources. However, the lazy approach may incur inefficiency and delay in query processing, especially when queries are issued multiple times, when information sources are slow, expensive, or periodically unavailable, and when significant processing is required for the translation, filtering, and merging steps. In various cases where information sources do not permit the use of ad-hoc queries, the lazy approach is simply not feasible.
In the data warehousing approach, the integrated information is available for querying and analysis by clients. Thus, the warehousing approach is appropriate for these:

- Clients requiring specific, predictable portions of the available information.
- Clients requiring high query performance and efficiency but not necessarily requiring the most recent state of the information.
- Environments in which native applications at the information sources require high performance.
- Clients wanting access to private copies of the information. so that they can be modified, annotated, and so on, or clients wanting to save information that is not maintained at the sources.

II. DESCRIPTION

A. Architecture

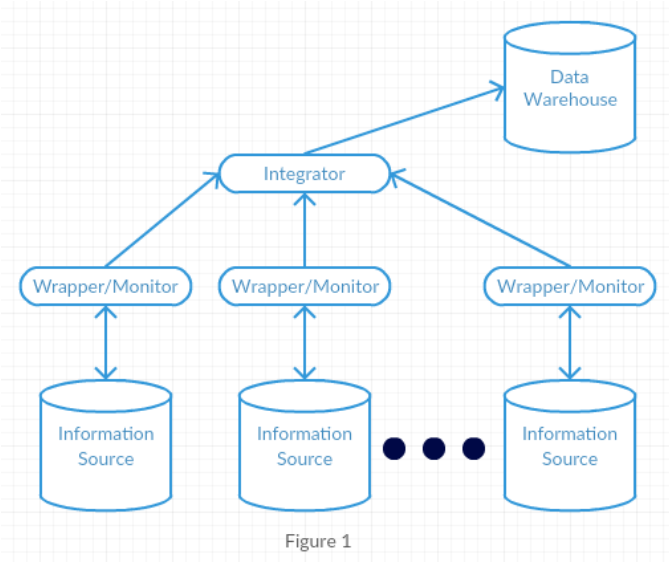

Figure 1

Figure 1 illustrates the basic architecture of a data warehousing system. The bottom of the diagram shows the information sources. Although the traditional disk shapes connote conventional database systems, in the general case these sources may include non-traditional

Data such as flat files, newswires, HTML and SGML documents, knowledge bases, legacy systems, and so on. Connected to each information source is a wrapper/monitor. The wrapper component is responsible for translating information from the format in which the source is present and data model used by the warehousing system, while the monitor component is responsible for automatically detecting changes of interest in the source data and reporting them to the integrator. When a new information source is attached to the warehousing system, or when relevant information at a source changes, the new or modified data is propagated to the integrator. The integrator is an important component and is responsible for installing the information in the warehouse, which includes faltering the information, summarizing it or merging it with information from other sources. In order to properly integrate the new information into the warehouse, it may be necessary for the integrator to obtain further information from the same or different information sources. This behaviour is illustrated by the downward dashed arrows in Figure 1. The data warehouse itself can use an off-the-shelf or special purpose database management system. Although in Figure 1 we illustrate a single, centralized warehouse, the warehouse certainly may be implemented as a distributed database system, and in fact data parallelism or distribution may be necessary to provide the desired performance. The architecture and basic functionality we have described is more general than that provided by most commercial data warehousing systems. In particular, current systems usually

Assume that the sources and the warehouse subscribe to a single data model (normally relational), that propagation of information from the sources to the warehouse is performed as a batch process (perhaps off-line), and that queries from the integrator to the information sources are never needed.

B. Data Marts

The requirements are necessary to be specified which users have access to data and any data aggregation security specifications.

C. Other Data Warehouse Security Considerations

Should consider requirements for other sensitive corporate data e.g. human resource data;

- Should consider test data requirements e.g. a need to “black-out” sensitive data during testing;
- Should consider security concerns for project documentation e.g. are detail design specifications of value to competitors and what are the requirements to prevent unauthorized access;
- Should ensure that the requirements specifications refer to the information management data security policy to ensure common understanding of all data warehouse security requirements.

III. DATA WAREHOUSE SECURITY GUIDELINES

A. Authentication

- Access to the network and to each system should be controlled through use of individually owned user accounts and associated confidential authentication key or password;
- A formal record should be maintained of all access rights, including complete user or account names and group descriptions;
- Accounts that become inactive or unused should be suspended after sixty (60) days, and, if they remain inactive, deleted after ninety (90) days;
- Temporary user accounts may have to be set up, e.g., for test purposes. Such accounts should have an expiration date.
- All the users shall be provided, initially and on a reset, a temporary password that they are required to change immediately. In order to ensure calls for password reset are valid, user's identities should be verified using information

which is present about the user that only the user would know. Temporary passwords should be conveyed to users in a secure manner;

- Unneeded or unsecured special accounts should be restricted or removed e.g. Guest, Anonymous, Null, and non-user accounts;
- The logon procedure should not request or display information during the logon procedure that would aid an unauthorized user;
- Logon information shall be validated only on completion of all input data. If an error condition comes into play during the login, the system shall not indicate which part of the data is correct or incorrect;
- The number of grace logins should be set to a maximum of 5-6 notices (preferably 6). A grace login allows the user to delay changing their password, and to log in for a fixed number of times, after which they are required to change their password, and cannot proceed to log in until they do;
- The number of unsuccessful logon attempts should be limited to four logins. After 4 consecutive unsuccessful logon attempts: 1) record the unsuccessful attempt; and 2) inactivate the account for an automated timeout/reset period of 20-25 minutes or greater, or in a manner that requires a manual reset by the system administrator; and
- User IDs should not give any indication of the user's privilege level e.g., manager nor the application system to which they have access.

B. Data Warehouse Security Passwords

- Passwords should have a minimum length of eight characters;
- Passwords should have a mix of alphabets, symbols and numeric characters;
- Passwords should not contain more than 3 consecutive identical characters;
- Passwords should not contain any control characters e.g., Ctrl-C or blank spaces
- Passwords should not be reused for at least five (5) generations pf consecutive changes;
- Passwords should be changed when the system prompts, or at least every 120 days if the system does not prompt for a change; Applications that utilize two-factor authorization are not required to expire on a pre-defined schedule;
- Passwords should not be easy to be guessed by others or through use of automated tools;
- Passwords should be stored in encrypted and hashed forms or with access controls;
- Password files should be stored separately from the main applications and other system data;
- Default passwords should be changed following installation of software and patches;
- An effective password management system should be applied or equivalent password management methodology should be used to authenticate users; and
- All passwords generated on behalf of an individual user must conform to the password management standards.

C. Applications

- Users of a system should not have unauthorized access to other user's data;
- Passwords should not be stored unencrypted on disk, in computer memory, or in any system-based data repository, e.g., the NT Registry;
- Passwords should not be embedded in macros, scripts, job control language, programs, or files, unless they have been stored encrypted, hashed, or with access controls;
- Passwords should not be displayed in clear text on the screen when being entered;
- Application and system output that contains confidential or proprietary data should be routed only to authorized terminals and locations;
- Initiation scripts and aliases, commands should be executed using fully-qualified command names, for example, full path names;
- Using the full path name for a command can prevent the execution of malicious code residing in a local directory;
- Screen savers should be set to activate after a period of fifteen (15) minutes of user inactivity, and should be password protected;
- Active application sessions should be terminated, unless they can be secured by a screen lock or other protection;
- Logon credentials should not be cached on the system;
- Any screen/web page requiring a password entry should be configured to prevent the caching of the entered password;
- Information should not be transmitted to cell phones in clear text;
- Applications should have controls to validate the integrity of data prior to be used as input to the application;
- Applications should have controls to validate the correct processing of data to detect data integrity errors caused by processing errors or malicious acts;
- Applications should have controls to validate the integrity of electronically transmitted data to detect corruption or unauthorized changes;
- Applications should have controls to validate the integrity of processed or stored data; Passwords should not be sent to the user via email unless the password is encrypted using an approved encryption tool such as Entrust;
- Passwords should be encrypted during transmission; Browser based applications should use 128-bit encryption if it exchanges confidential data with the browser; and

- Browser based applications that require SSL should restrict access to browsers capable of supporting 128-bit, or higher, encryption.

There are several key considerations when implementing a data warehouse.
The first consideration, which is hardly surprising, is that the data warehouse team must consider end-to-end encryption techniques to provide security. A data warehouse environment consists of much more than just a database. The entire environment ranges from the extraction of data from operational system, transportation of this data to the data warehouse, the possible distribution of these data to data marts and other analytic servers, and finally the dissemination of this data to end-users. The environment spans multiple servers and multiple software products ... and of course every component needs to be secure.

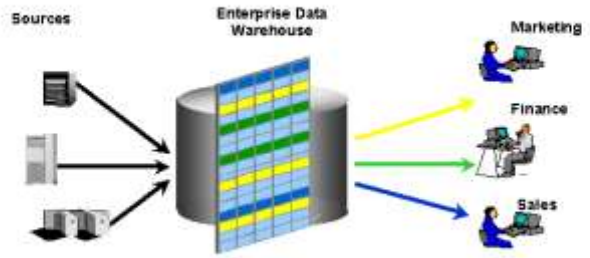

Figure 2: Consolidation in an enterprise data warehouse with built-in security.

There are undoubtedly many data warehouses today in which the database itself has little risk of a security breech, but at the same time the flat files which are used to populate the data warehouse are stored in an unsecured location. This is a solid example of the security loopholes that can emerge when the entire data-warehouse process is not designed with various security techniques in mind.
The second consideration is related to the interaction of security and the data warehouse architecture. A consolidated data warehouse is much simpler to secure than dozens of heterogeneous data marts. Indeed, many industry analysts and customers agree that an enterprise data warehouse is the preferred implementation model, and among that model's many virtues is the fact that a centralized data warehouse's security is simpler and less expensive to manage, while providing higher levels of security.

The final consideration is the recognition the core of a data warehouse is the data. Although end-to-end security is crucial, the ability to provide a flexible multi-layer security model on the data in the data warehouse is nevertheless the primary requirement for data warehouse security

IV. CONCLUSION

The area of data warehousing and the various measures related to the security in this field for integrating multiple, distributed, heterogeneous information sources, is a viable and in some cases superior alternative to traditional research solutions. Traditional approaches are slow, and work in a specific manner i.e. request, process, and merge information from sources when queries are posed. In the data warehousing approach, information is requested, processed, and merged continuously in a single flow, so the information is readily available for direct querying and analysis at the warehouse. Although the concept of data warehousing already is prominent in the database industry, we believe there are a number of important open research problems, described above, that need to be solved to realize the flexible, powerful and efficient data warehousing systems of the future which, apart from being powerful and flexible will also be secure.

V. REFERENCES

[1] Hayen R., Rutashobya C., Vetter D., (2007)
[2] Keith L., Mark F., "Current Issues in Data Warehousing.", (2002).
[3] Hongjiang X., and Mark H., "The Effect of Implementation Factors on Data Warehousing Success: An Exploratory Study.",*Journal of Information, Information Technology, and Organizations*" (2007).
[4] Wixom B., and Hugh W., "An Empirical Investigation of The Factors Affecting Data Warehousing Success." MIS quarterly (2001): 17-41.
[5] Amin, MdRuhul, and MdTaslimArefin. "The Empirical Study on the Factors Affecting Data Warehousing Success."International Journal of Latest Trends in Computing.Vol (1) pp 138- 142.
[6] Chenoweth T., Karen C., and Haluk D., "Seven key interventions for data warehouse success." Communications of the ACM 49, no. 1 (2006): 114-119.