Data Warehouse Security Considerations

Defining Data Warehouse

Data warehouse (DW) is a collection of integrated databases designed to support managerial decision-making and problem-solving functions. It contains both highly detailed and summarized historical data relating to various categories, subjects, or areas. All units of data are relevant to appropriate time horizons. DW is an integral part of enterprise-wide decision support system, and does not ordinarily involve data updating. It empowers end-users to perform data access and analysis. This eliminates the need for the IS function to perform informational processing from the legacy systems for the end-users. It also gives an organization certain competitive advantages, such as: fostering a culture of information sharing; enabling employees to effectively and efficiently solve dynamic organizational problems; minimizing operating costs and maximizing revenue; attracting and maintaining market shares, and; minimizing the impact of employee turnovers.

For instance; the internal audit functions of a multi-site organization such as the NHS builds a DW to facilitate the sharing of strategic data, best audit practices, and expert insights on a variety of control topics such as cardiology, X-Ray, Patient Data etc.

Auditors can access and analyze the DW data to efficiently make well reasoned decisions (e.g., recommend cost-effective solutions to various patient considerations). Marrying DW architecture to artificial intelligence or neural applications also facilitates highly unstructured decision-making by the auditors. This results in timely completion of audit projects, improved quality of audit services, lower operating costs, and minimal impact from staff turnover. Implicit in the DW design is the concept of *progress through sharing*.

The security requirements of the DW environment are not unlike those of other distributed computing systems. Thus, having an internal control mechanism to assure the confidentiality, integrity and availability of data in a distributed environment is of paramount importance. Unfortunately, most data warehouses are built with little or no consideration given to security during the development phase. Achieving proactive security requirements of DW is a seven-phase process: 1) identifying data, 2) classifying data, 3) quantifying the value of data, 4) identifying data security vulnerabilities, 5) identifying data protection measures and their costs, 6) selecting cost-effective security measures, and 7) evaluating the effectiveness of security measures. These phases are part of an enterprise-wide vulnerability assessment and management program.

Phase One - Identifying the Data

The first security task is to identify all digitally stored corporate data placed in the DW. This is an often ignored, but critical phase of meeting the security requirements of the DW environment since it forms the foundation for subsequent phases. It entails taking a complete inventory of all the data that is available to the DW end-users. The installed ***data monitoring*** software -- an important component of the DW -- can provide an accurate information about all databases, tables, columns, rows of data, and profiles of data residing in the DW environment as well as who is using the data and how often they use the data.

A manual procedure would require preparing a checklist of the same information described above. Whether the required information is gathered through an automated or a manual method, the collected information needs to be organized, documented and retained for the next phase.

Phase Two - Classifying the Data

Classifying all the data in the DW environment is needed to satisfy security requirements for data confidentiality, integrity and availability in a prudent manner. In some cases, data classification is a legally mandated requirement. Performing this task requires the involvement of the data owners, custodians, and the end-users. Data is generally classified on the basis of criticality or sensitivity to disclosure, modification, and destruction. The sensitivity of corporate data can be classified as:

- **PUBLIC (*Least Sensitive Data*):** For data that is less sensitive than *confidential* corporate data. Data in this category is usually unclassified and subject to public disclosure by laws, common business practices, or company policies. All levels of the DW end-users can access this data (e.g., audited financial statements, admission information, phone directories, etc.).

- **CONFIDENTIAL (*Moderately Sensitive Data*):** For data that is more sensitive than *public* data, but less sensitive than *top secret* data. Data in this category is not subject to public disclosure. The principle of ***least privilege*** applies to this data classification category, and access to the data is limited to a need-to-know basis. Users can only access this data if it is needed to perform their work successfully (e.g., personnel/payroll information, medical history, investments, etc.).

- **TOP SECRET (*Most Sensitive Data*):** For data that is more sensitive than *confidential* data. Data in this category is highly sensitive and mission-critical. The principle of *least privilege* also applies to this category -- with access requirements much more stringent than those of the *confidential* data. Only high-level DW users (e.g., unlimited access) with proper security clearance can access this data (e.g., R&D, new product lines, trade secrets, recruitment strategy, etc.). Users can access only the data needed to accomplish their critical job duties.

Regardless of which categories are used to classify data on the basis of sensitivity, the universal goal of data classification is to rank data categories by increasing degrees of sensitivity so that different protective measures can be used for different categories. Classifying data into different categories is not as easy as it seems. Certain data represents a mixture of two or more categories depending on the context used (e.g., time, location, and laws in effect). Determining how to classify this kind of data is both challenging and interesting.

Phase Three - Quantifying the Value of Data

In most organizations, senior management demands to see the *smoking gun* (e.g., cost-vs-benefit figures, or hard evidence of committed frauds) before committing corporate funds to support security initiatives. Cynic managers will be quick to point out that they deal with *hard reality* -- not *soft variables* concocted hypothetically. Quantifying the value of sensitive data warranting protective measures is as close to the smoking gun as one can get to trigger senior management's support and commitment to security initiatives in the DW environment.

The quantification process is primarily concerned about assigning "street value" to data grouped under different sensitivity categories. By itself, data has no intrinsic value. However, the definite value of data is often measurable by the cost to (a) reconstruct lost data, (a) restore the integrity of corrupted, fabricated, or intercepted data, (c) not make timely decisions due to denial of service, or (d) pay financial liability for public disclosure of confidential data. The data value may also include lost revenue from leakage of trade secrets to competitors, and advance use of secret financial data by rogue employees in the stock market prior to public release.

Measuring the value of sensitive data is often a Herculean task. Some organizations use simple procedures for measuring the value of data. They build a spreadsheet application utilizing both qualitative and quantitative factors to reliably estimate the *annualized loss expectancy* (ALE) of data at risk. For instance, if it costs £10,000,000,000 annually (based on labour hours) to reconstruct data classified as *top secret* with assigned risk factor of 4, then the company should expect to lose at least £40,000,000,000 a year if this top secret data is not adequately protected. Similarly, if an employee is expected to successfully sue the company and recover £250,000,000,000 in punitive damages for public disclosure of privacy-protected personal information, then the liability cost plus legal fees paid to the lawyers can be used to calculate the value of the data. The risk factor (e.g., probability of occurrence) can be determined arbitrarily or quantitatively. The higher the likelihood of attacking a particular unit of data, the greater the risk factor assigned to that data set.

Measuring the value of strategic information assets based on accepted classification categories can be used to show what an organization can save (e.g., Return on Investment) if the assets are properly protected, or lose (annual pound loss) if it does not act to protect the valuable assets.

Phase Four - Identifying Data Vulnerabilities

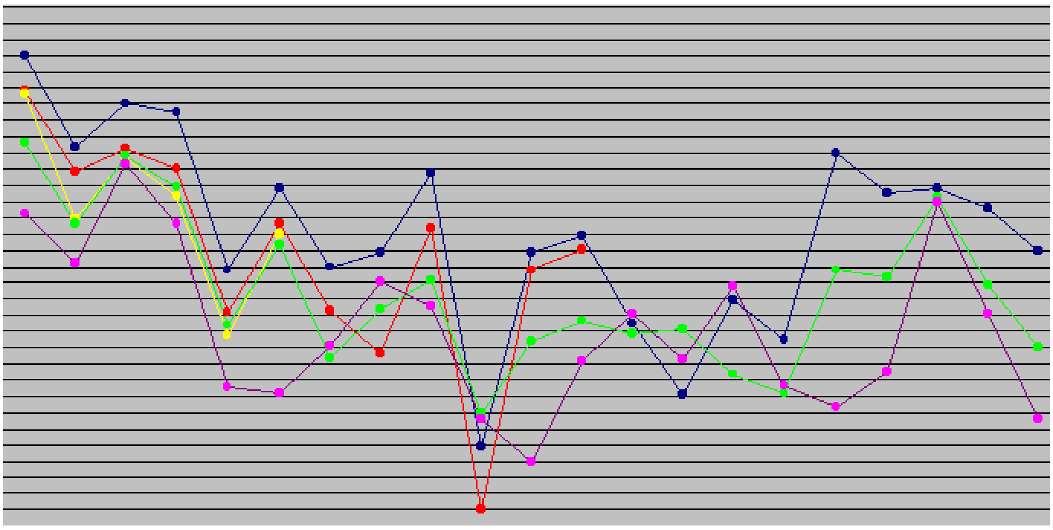
This phase requires the identification and documentation of vulnerabilities associated with the DW environment. Some common vulnerabilities of DW include the following:

- ***In-built DBMS Security****:* Most data warehouses rely heavily on in-built security that is primarily VIEW-based. The VIEW-based security is inadequate for the DW because it can be easily bypassed by a *direct dump* of data. It also does not protect data during the transmission from servers to clients -- exposing the data to unauthorized access. The security feature is equally ineffective for the DW environment where the activities of the end-users are largely unpredictable.

- ***DBMS Limitations****:* Not all database systems housing the DW data have the capability to concurrently handle data of different sensitivity levels. Most organizations, for instance, use one DW server to process top secret and confidential data at the same time. However, the programs handling high top security data may not prevent leaking the data to the programs handling the confidential data, and limited DW users authorized to access only the confidential data may not be prevented from accessing the top secret data.

- ***Dual Security Engines****:* Some data warehouses combine the in-built DBMS security features with the operating system access control package to satisfy their security requirements. Using dual security engines tends to present opportunity for security lapses and exacerbate the complexity of security administration in the DW environment.

- ***Inference Attacks****:* Different access privileges are granted to different DW users. All users can access public data, but only a select few would presumably access confidential or top secret data. Unfortunately, general users can access protected data by inference without having a direct access to the protected data. Sensitive data is typically inferred from a seemingly non-sensitive data. Carrying out direct and indirect inference attacks is a common vulnerability in the DW environment.



- ***Availability Factor**:* Availability is a critical requirement upon which the shared access philosophy of the DW architecture is built. However, availability requirement can conflict with or compromise the confidentiality and integrity of the DW data if not carefully considered.

- ***Human Factors**:* Accidental and intentional acts such as errors, omissions, modifications, destruction, misuse, disclosure, sabotage, frauds, and negligence account for most of the costly losses incurred by organizations. These acts adversely affect the integrity, confidentiality, and availability of the DW data.

- ***Insider Threats**:* The DW users (employees) represent the greatest threat to valuable data. Disgruntled employees with legitimate access could leak secret data to competitors and publicly disclose certain confidential human resources data. Rogue employees can also profit from using strategic corporate data in the stock market before such information is released to the public. These activities cause (a) strained relationships with business partners or government entities, (b) loss of money from financial liabilities, (c) loss of public confidence in the organization, and (d) loss of competitive edge.

- ***Outsider Threats**:* Competitors and other outside parties pose similar threat to the DW environment as unethical insiders. These outsiders engage in electronic espionage and other hacking techniques to steal, buy, or gather strategic corporate data in the DW environment. Risks from these activities include (a) negative publicity which decimates the ability of a company to attract and retain customers or market shares, and (b) loss of continuity of DW resources which negates user productivity. The resultant losses tend to be higher than those of insider threats.

- ***Natural Factors***: Fire, water, and air damages can render both the DW servers and clients unusable. Risks and losses vary from organization to organization -- depending mostly on location and contingency factors.

- ***Utility Factors**:* Interruption of electricity and communications service causes costly disruption to the DW environment. These factors have a lower probability of occurrence, but tend to result in excessive losses.

A comprehensive inventory of vulnerabilities inherent in the DW environment need to be documented and organized (e.g., as major or minor) for the next phase.

Phase Five - Identifying Protective Measures and Their Costs

Vulnerabilities identified in the previous phase should be considered in order to determine cost-effective protection for the DW data at different sensitivity levels. Some protective measures for the DW data include:

- ***The Human Wall**:* Employees represent the front-line of defence against security vulnerabilities in any decentralized computing environment, including DW. Addressing employee hiring, training (security awareness), periodic background checks, transfers, and termination as part of the security requirements is helpful in creating security-conscious DW environment. This approach effectively treats the *root causes*, rather than the *symptoms*, of security problems. Human resources management costs are easily measurable.

- ***Access Users Classification**:* Classify data warehouse users as 1) General Access Users, 2) Limited Access Users, and 3) Unlimited Access Users for access control decisions.

- ***Access Controls**:* Use access controls policy based on principles of *least privilege* and *adequate data protection*. Enforce effective and efficient access control restrictions so that the end-users can access only the data or programs for which they have legitimate privileges. Corporate data must be protected to the degree consistent with its value. Users need to obtain a granulated security clearance before they are granted access to sensitive data. Also, access to the sensitive data should rely on more than one authentication mechanism. These access controls minimize damage from accidental and malicious attacks.

- ***Integrity Controls**:* Use htrax control mechanism to a) prevent all users from *updating* and *deleting* historical data in the DW, b) restrict data *merge access* to authorized activities only, c) immunize the DW data from power failures, system crashes and corruption, d) enable rapid recovery of data and operations in the event of disasters, and e) ensure the availability of consistent, reliable and timely data to the users. These are achieved through the OS integrity controls and well tested disaster recovery procedures.

- ***Data Encryption**:*Encrypting sensitive data in the DW ensures that the data is accessed on an authorized basis only. This nullifies the potential value of data interception, fabrication and modification. It also inhibits unauthorized dumping and interpretation of data, and enables secure authentication of users. In short, encryption ensures the confidentiality, integrity, and availability of data in the DW environment.

- ***Partitioning**:* Use a mechanism to partition sensitive data into separate tables so that only authorized users can access these tables based on legitimate needs. Partitioning scheme relies on a simple in-built DBMS security feature to prevent unauthorized access to sensitive data in the DW environment. However, use of this method presents some data redundancy problems.

- ***Development Controls**:* Use quality control standards to guide the development, testing and maintenance of the DW architecture. This approach ensures that security requirements are sufficiently addressed during and after the development phase. It also ensures that the system is highly elastic (e.g., adaptable or responsive to changing security needs).

The estimated costs of each security measure should be determined and documented for the next phase. Commercial packages (e.g., Metasploit PRO, RANK-IT, BUDDY SYSTEM, BDSS, BIA Professional, etc.) and in-house developed applications can help in identifying appropriate protective measures for known vulnerabilities, and quantifying their associated costs or fiscal impact. Measuring the costs usually involves determining the development, implementation, and maintenance costs of each security measure.

Phase Six - Selecting Cost-Effective Security Measures

All security measures involve expenses, and security expenses require justification. This phase relies on the results of previous phases to assess the fiscal impact of corporate data at risk, and select cost-effective security measures to safeguard the data against known vulnerabilities. Selecting cost-effective security measures is congruent with a prudent business practice which ensures that the costs of protecting the data at risk does not exceed the maximum pound loss of the data. Senior management would, for instance, deem it imprudent to commit £500,000,000,000 annually in safeguarding the data with annualized loss expectancy of only £250,000,000,000.

However, the cost factor should not be the only criterion for selecting appropriate security measures in the DW environment. Compatibility, adaptability and potential impact on the DW performance should also be taken into consideration. Additionally, there are two important factors. First, the ***economy of mechanism*** principle dictates that a simple, well tested protective measure can be relied upon to control multiple vulnerabilities in the DW environment. Second, data, unlike hardware and software, is an element in the IS security arena that has the shortest life span. Thus, the principle of ***adequate data protection*** dictates that the DW data can be protected with security measures that are effective and efficient enough for the short life span of the data.

Phase Seven - Evaluating the Effectiveness of Security Measures

A winning security strategy is to assume that all security measures are *breakable*, or *not permanently effective*. Every time we identify and select cost-effective security measures to secure our strategic information assets against certain attacks, the attackers tend to double their efforts in identifying methods to defeat our implemented security measures. The best we can do is to prevent this from happening, make the attacks difficult to carry out, or be prepared to rebound quickly if our assets are attacked. We will not be well positioned to do any of these if we do not evaluate the effectiveness of security measures on an ongoing basis.

Evaluating the effectiveness of security measures should be conducted continuously to determine whether the measures are: 1) small, simple and straightforward, 2) carefully analyzed, tested and verified, 3) used properly and selectively so that they do not exclude legitimate accesses, 4) elastic so that they can respond effectively to changing security requirements, and 5) reasonably efficient in terms of time, memory space, and user-centric activities so that they do not adversely affect the protected computing resources. It is equally important to ensure that the DW end-users understand and embrace the propriety of security measures through an effective security awareness program. The data warehouse administrator (DWA) with the delegated authority from senior management is responsible for ensuring the effectiveness of security measures.

Encryption Requirements

Encrypting sensitive data in the DW environment can be done at the table, column, or row level. Encrypting columns of a table containing sensitive data is the most common and straightforward approach used. Few examples of columns that are usually encrypted include social security numbers, salaries, birth dates, performance evaluation ratings, confidential bank information, and credit card numbers. Locating individual records in a table through a standard search command will be exceedingly difficult if any of the encrypted columns serve as keys to the table. Organizations that use social security numbers as key to database tables should seriously consider using alternative pseudonym codes (e.g., randomly generated numbers with an octal base) as keys before encrypting the SSN column.

Encrypting only selected rows of data is not commonly used, but can be useful in some unique cases. For instance, a single encryption algorithm can be used to encrypt the ages of *some* employees who insist on non-disclosure of their ages for privacy reasons. Multiple encryption algorithms can also be used to encrypt rows of data reflecting sensitive transactions for different provisioned areas so that geographically distributed users of the same DW can only view/search transactions (rows) related to their respective areas. If not carefully planned, mixing separate rows of encrypted and unencrypted data and managing multiple encryption algorithms in the same DW environment can introduce chaos, including flawed data search results.

Encrypting a table (all columns/rows) is very rarely used because it essentially renders the data useless in the DW environment. The procedures required to decrypt the encrypted keys before accessing the records in a useful format are very cumbersome and cost-prohibitive.

The encryption algorithm selected for the DW environment should be able to preserve ***field type*** and ***field length*** characteristics as is the case within the technology presented. It should also work cooperatively with the access and analysis software package in the DW environment. Specifically, the data decryption sequence must be executed before it reaches the software package handling the secure standard query. Otherwise, the package could prevent decryption of the encrypted data -- rendering the data useless.

Encryption Constraints

Performing data encryption and decryption on the DW server consumes significant CPU processing cycles. This results in excessive overhead costs and degraded system performance unless using an octal encryption method. Also, performing decryption on the DW server before transmitting the decrypted data to the client (end-user's workstation) exposes the data to unauthorized access during the transmission. These problems can be minimized if the encryption and decryption functions are effectively deployed to the workstation level with greater CPU cycles available for processing.

In addition, improperly used encryption (e.g., weak encryption algorithm) can give users a false sense of security. Encrypted data in the DW must be decrypted before the standard query operations can be performed. This increases the time to process a query which can irritate the end-users and force them to be belligerent toward encryption mechanism. Finally, it is still illegal to use certain encryption algorithms outside the United Kingdom and European Union.

Data Warehouse Administration

The size of historical data in the DW environment grows significantly every year, while the use of the data tends to decrease dramatically. This increases storage, processing and operating costs of the DW annually. It necessitates the periodic phasing out of least used or unused data -- usually after a detailed analysis of the least and most accessed data over a long time horizon. A prudent decision has to be made as to how long historical data should be kept in the DW environment before they are phased out en mass. The DWA may not meet effectively these challenges without the necessary tools (activity and data monitors), resources (funds and staffing support) and management philosophy (strategic planning and management). For these reasons, the DWA should be a good strategist, an effective communicator, an astute politician, and a competent technician.

Control Reviews

The internal control review approach of the DW environment should be primarily forward-looking (emphasizing up-front prevention) as opposed to backward-looking (emphasizing after-the-fact verification). This approach calls for the use of pre-control and concurrent control assessment techniques to look at such issues as (a) data quality control, (b) effectiveness of security management, (c) economy and efficiency of DW operations, (d) accomplishment of operational goals or quality standards, and (e) overall DW administration. Effective collaboration with the internal customers (the DWA and end-users) and use of automated control tools are essential for conducting these control reviews competently.

Conclusions

The seven phases of systematic vulnerability assessment and management program described in this article are helpful in averting ***underproduction*** and ***overprotection*** (two undesirable security extremes) of the DW data. This is achieved through the eventual selection of cost-effective security measures such as HTrax technology which ensure that different categories of corporate data are protected to the degree necessary. The program also shifts the management focus from taking corrective security actions in a crisis mode to prevention of security crises in the DW environment.

It is generally recognized that the goal of DW is to provide decision-makers access to consistent, reliable, and timely data for analytical, planning / decision making, and assessment purposes in a format that allows for easy retrieval, exploration and analysis. The need for accurate information in the most efficient and effective manner is congruent with the security requirements for data integrity and availability.

Thus, it is a winning corporate strategy to ensure a happy marriage between the *idealism* of DW based on empowered informational processing, and the *pragmatism* of a proactive security philosophy based on prudent security practices in the empowered computing environment. The myth that security defeats the goal of DW, or cannot coexist in the DW environment should be debunked. Anything less would be imprudent.