Commentary

Security Issues in ETL for the Data Warehouse

The data warehouse and the query and reporting tools that access it represent obvious security risks in a business intelligence infrastructure. Extraction, transformation and loading processes also merit a focus on security.

Enterprises that focus on the security of their business intelligence (BI) environment generally limit their scope to the data warehouse and the data being presented to end users. Sensible enterprises ensure that the data in the data warehouse and any related data marts is secure, and restrict access to query and reporting tools. Although these are certainly the most-critical security areas to address, risks exist on the back end of the data warehouse as well. Extraction, transformation and loading (ETL) processes, which prepare and deliver data to the data warehouse, have security concerns of their own and should be considered a significant part of any BI security improvement efforts.

Source Data and Data in Transit Are Security Risks

ETL processes are about pulling data from operational systems. Whether using ETL tools or writing custom ETL code, enterprises must make source systems accessible to ETL developers to acquire the data needed for the data warehouse. Often, ETL processes are given access to source systems via a system-administration-level user ID. This is a risky approach, because it typically gives ETL developers visibility to far more data than they need. The security risk of unauthorized data extract can be limited by providing the ETL process access to only the data required in the data warehouse. A common alternative involves the application support teams handling the extract themselves, providing the necessary data to the ETL process in the form of flat files or other staging formats.

Additional security concerns arise because ETL processes are rarely single-step flows, where data moves with no stops along the way from the operational source system to the target data warehouse. The complexity inherent in most environments requires data warehouse project teams to implement ETL processes in multiple steps. Data may be extracted from legacy systems and placed in flat files on the same platform as the source system. These files can then be moved, usually via File Transfer Protocol (FTP), to an ETL server for subsequent processing. Source data is often loaded to a relational staging area where initial transformation work is performed to restructure the data so that integration across multiple sources can be achieved. The staging area is often the place where complex transformations are executed prior to loading to the data warehouse.
At the back end of the ETL process, data may be staged again in flat files, ready for input to the bulk-load utilities of the target data warehouse database management system (DBMS). The data at each of these steps — extracted flat files, staging areas and load files — must be secured through access controls so it cannot be viewed or "siphoned off" anywhere along the way. Moreover, the time in which this "in process" data persists should be minimized. Data in staging areas may be useful for troubleshooting data flaws in the data warehouse or for restart and recovery following errors, but should be retained in an easily accessible format only as long as needed for these purposes.

**Fragile ETL Logic**

As well as the security of the data flowing through the ETL process, the security of the ETL logic is critical.

Getting ETL logic right is one of the most challenging and technically complex aspects of a data warehouse implementation and consumes the overwhelming majority of data warehouse project time and resources. Even small flaws in the ETL process can cause data warehouse reliability to suffer. Data will be inaccurate or will not be available in the data warehouse.

Because multiple ETL developers are often involved in an implementation, the security of the ETL logic must be maintained so that changes are not accidentally or maliciously introduced into the environment. Most ETL logic is implemented through custom code (for example, COBOL, C or SQL). Enforcing security on these various components, along with standard version control procedures, is critical to ensure that unauthorized changes to the process are not made.

ETL tools provide some degree of security, typically including functionality to secure the design repository (the ETL metadata) at the project level and, in some cases, down to the object level (for example, an individual transformation). Although this helps restrict access to ETL elements by people not part of the development and maintenance effort, it generally does not provide the granular level of version control on objects that is important in very large multideveloper efforts.

**Impact of Web Services**

Several ETL tools' vendors have defined product road maps that include a vision for Web services support. The concept of making an ETL job easily accessible from a range of applications by way of a simple, callable interface may seem attractive for improving the leverage and value of ETL tools across the enterprise. However, this direction can also introduce significant security risks as enterprises struggle to control access to these interfaces and track where and when they are being used. Enterprises must realize that a tremendous amount of strategic and highly proprietary information flows through ETL processes. It's crucial to consider the security concerns (from a conceptual and technical perspective) inherent in making those processes more accessible via Web services.

**Bottom Line:** ETL processes pose subtle security risks often overlooked in data warehouse implementations. Enterprises must view ETL as a production application, which must be secured to the same degree as any other production application in the environment. ETL "logic" (the transformations and metadata) and data staging areas must be considered in the overall BI security scheme. Explicitly defining roles and access rights for ETL components will minimize the risk of security breaches through this part of the data warehouse architecture.