JReport®
JINFONET SOFTWARE

JReport Security

Business Intelligence and Security

The Importance of Security

Business Intelligence (BI) isn't just about accessing information. It is about ensuring that the right people in an organization have the information they need to make confident decisions; and that people can only access the data for which they are properly authorized.

In Software as a Service (SaaS) applications where multiple companies' data may be stored in the same DBMS, it is critical to the integrity of the SaaS provider that each company's data is secure, and protected from being viewed by other companies' employees.

This is also the case when reports are prepared for customers, such as invoices, order history reports, credit information, and other vital personal information. It is essential that companies ensure such personal reports are not accidently shared with people who should not have access. For example, Visa's small business accounts use JReport to allow their customers to see their charges at any time. Customers must have confidence that JReport and Visa are handling their confidential information with the utmost care.

Business Intelligence plus Security is a vital combination to ensure the right information is delivered only to authorized people.

When reviewing a BI Vendor's offerings, it is essential to evaluate security as a key component. In most cases, security is not offered by the Free Open Source Software (FOSS) vendors. Only the commercial versions of the open source products (COSS) offer security and even that is much less comprehensive than full commercial offerings. This is a clear indicator that without a strong security model, a BI tool has little value.

Setting up Security

Authentication technology is used to ensure that a user is who he claims to be. There are many ways to do this, which this paper will not detail. However, JReport provides a built in authorization system as well as supporting all common infrastructure methods such as LDAP, Active Directory and Single Sign On using external authentication systems.

Authorization is also sometimes called provisioning or permissions. Based on the authentication we know who the user is, and the role that he plays in the organization, and can thus know what reports he can run and what data in the report he can view. That is, given the authentication, we know what reports and data the user is authorized to access.

Organizations need to define users and roles, and then provide a way to match users to the specific role they play in the organization. Security rules then will be placed on the roles, allowing people mobility within the organization, while keeping information secure.

Data Source Security

The first level is physical security. This can be based on having different databases, different virtual machines, or entirely separate computers for each customer. It is physically impossible for one company's information to be mixed with another's if these precautions are taken.

The next level is DBMS user/password security. This is the underlying security of the DBMS, used to ensure that reports can access only the data for a particular customer, or role, that is mapped to a DBMS user. Generally, this is a high level role, so that one user/password in the DBMS has access to all data for the customer. This is usually not at the level of granularity needed for BI tools, but does provide a degree of separation between applications.

JReport has a catalog (metadata layer) which all of the reports use to gain access to data. This is a big advantage over many competing products which embed the connection information directly into the report template. In JReport, an administrator simply changes one file (called datasource.xml), to reconfigure all of the report templates that use that data source to change the DBMS host, JDBC URL, user and password. When adding a new company or new database instance, no changes are required to the individual report templates. This also provides a level of internal security so developers can easily build and test reports using non-production data and the application administrator can modify datasource.xml at deployment time without having to give the access information to developers to modify code or report templates.

Resource Access Security

The physical and DBMS security are usually not granular enough for a BI environment. For example, consider reports for a sales organization. Monthly sales reports need to be generated either on a schedule, on-demand or via an ad hoc interface based on the hierarchical roles in the sales organization. Individual account managers have access to only the accounts in their territory, regional managers see data for all the account managers in their region and the VP sales can see data for the entire company. Typically the DBMS would have one login/password for access to all of the sales related data.

The first level of security within JReport is permission to run the report, schedule the report or view the report results. Each resource, which includes folders, report templates and report results, can be restricted based on permissions. Available permissions are: Visible, Read, Write, Delete, Execute, Schedule, Grant and Update Status. For example if a particular role has Visible, Read and Execute permission on a folder, the user can see the objects within the folder, read the object (such as a report result), and run reports in the folder. He cannot, however, update the reports (write permission), schedule the reports, delete the reports or results, or give his access rights to someone else. All objects inside the folder, including other folders, inherit the permissions of the parent folder unless they are explicitly changed individually. All of these permissions can be set based on any

combination of Roles, Groups and Users, in an easy-to-use browser based administration console.

Parameter Based Security

One common method used by JReport to access data for a report is via parameters. Here, the application making the request to JReport automatically applies a parameter to all of the queries, such as providing the individual territory or region to the query based on the requestor's role. This is transparent to the user running the report but does require that the calling application pass the appropriate parameters. The advantage to this method is that the same easily created and maintained report templates can be used by all levels of the organization. The disadvantage is that each user must run their own individual query to the DBMS. If there are hundreds of account managers, dozens of regional managers and a VP, the same data must be retrieved many times from the DBMS. This can make the load on the computer systems and DBMS quite heavy.

Metadata Based Security

The metadata based methods Row Level Security and Column Level Security are similar to using parameters for accessing data. These are built entirely within a JReport catalog and so do not rely on the calling application. Instead of parameters being passed into JReport to limit the data retrieved by the queries, the metadata layer limits the query dynamically based on role permissions built into the catalog. The end result is identical to the parameter method. Looking from the DBMS perspective, only the data the user is authorized to see is returned from the DBMS to the report. Additionally, the Column Level Security feature controls the actual columns of data to be visible or invisible on the report based on the user's role. Thus private data, such as a Social Security Number or Salary, could be invisible except to the authorized person running the report.

Running reports on-demand using parameters or metadata, rather than scheduling them, offers the advantage of potentially reducing the actual number of reports created, since not every account manager may request the report each month. Further, the reports may be simply viewed and deleted, or exported to the user's local disk. IT's responsibility is reduced, only needing to manage reports stored in JReport Version System by setting auto-purging policies.

Load on the systems can, however, be hard to manage. If many requests come through simultaneously immediately before a sales forecast meeting, performance for the entire organization could be impacted, or service could even be interrupted entirely.

Both of the above methods, using parameters and metadata, require that unique queries for each role be executed by the DBMS. There are alternatives in JReport which use only one query to retrieve all of the required data from the DBMS. The user still only sees the data just as with

parameter and row level security methods, but the system uses far fewer resources on the DBMS and reporting servers.

Report Bursting

The first alternative is report bursting. Bursting allows JReport to take the data from a single query with all of the sales data and create unique reports for each level of management with a single run. The obvious benefit of this is that one query can be executed which retrieves all of the data, and the JReport server only passes through the data once to create all of the reports. Passing the data and formatting only once results in large performance savings on the DBMS server, the network load, and the JReport server.

Some other BI tools support one hierarchy for bursting, such as bursting a unique copy of the report out to each account manager. JReport, however, supports defining multiple hierarchies. With a single pass each account manager receives his report, each regional manager receives his report, and the VP receives his report with all of the sales with one query and a single pass of the data though JReport.

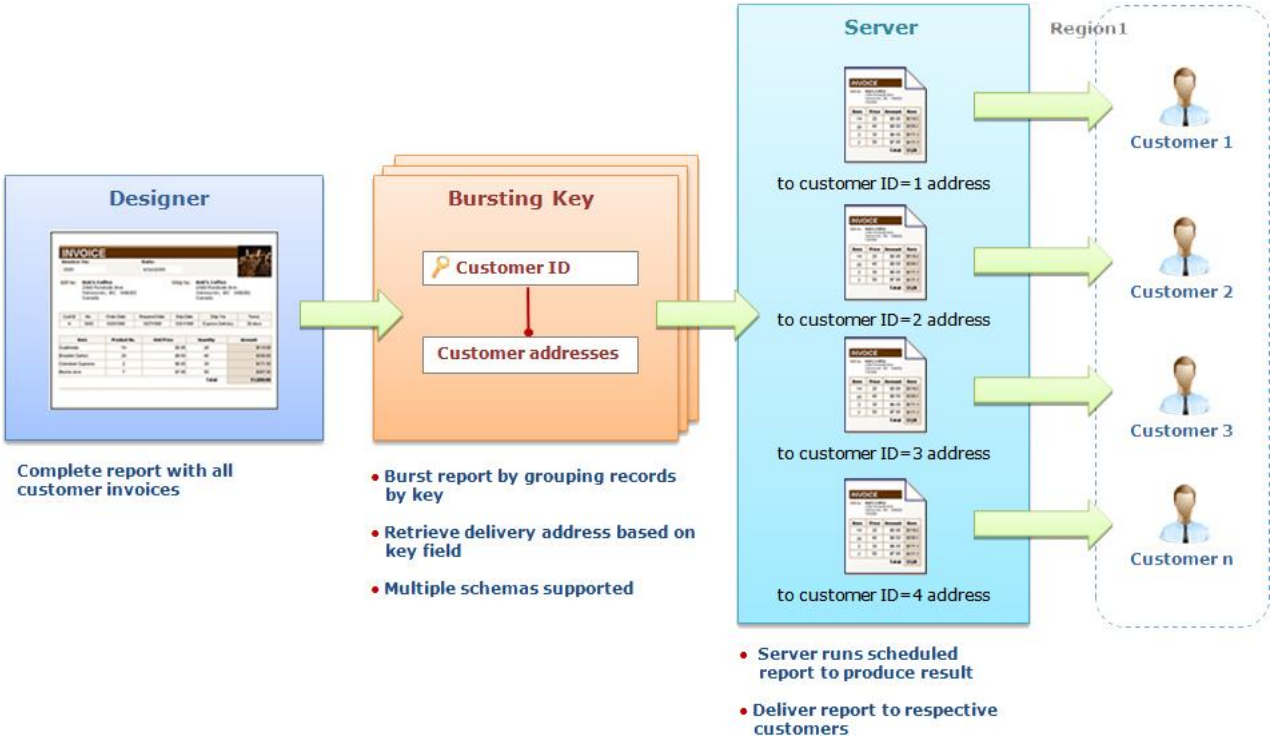

Illustration of Bursting Level Security

The obvious advantages to report bursting are that the load on the systems is much less, and the actual process of scheduling the reports to be run, or requesting the reports on-demand, is very easy. Simply schedule or request one report and everything is completed automatically, rather than scheduling or requesting hundreds of reports with different parameters; or, the case of

Row Level Security, by different actual users, where each user would have to schedule or run his own private report. Another advantage is that each user can choose the format in which they want to view their report, such as HTML, PDF or Excel, and the delivery method, such as JReport's version system, email, ftp or physical disk location. They can even choose to have the report delivered in multiple formats in multiple locations, still with just one run of the report. JReport bursting may be applied to any report output that results in a physical file, including PDF, HTML, Excel, and printed reports.

The security on all of the individual bursting files must be maintained to ensure security of the data. JReport provides multiple ways to handle file security such as permissions in the version system, use of directory permissions within the file system, secured email, and secured FTP.

A caution for bursting is that the proliferation of files must be managed and cleaned up when the value of the data expires. When using JReport's version system it makes this easy to do automatically but using the file system it must be done externally to JReport.

Group Level Security

Group Level Security is another combination of metadata security and bursting. As in Row Level Security, security information regarding which roles can view certain groups of data is built into the report definition. The data must be organized into groups, such as grouping by territory or region. The report is then divided into pages using the grouping as criteria. As in bursting, a single query is used to retrieve the data for all of the users of the report. Unlike bursting though, a single report file is created which contains the security rules that specify which groups and pages of data an individual user can see based on his roles.

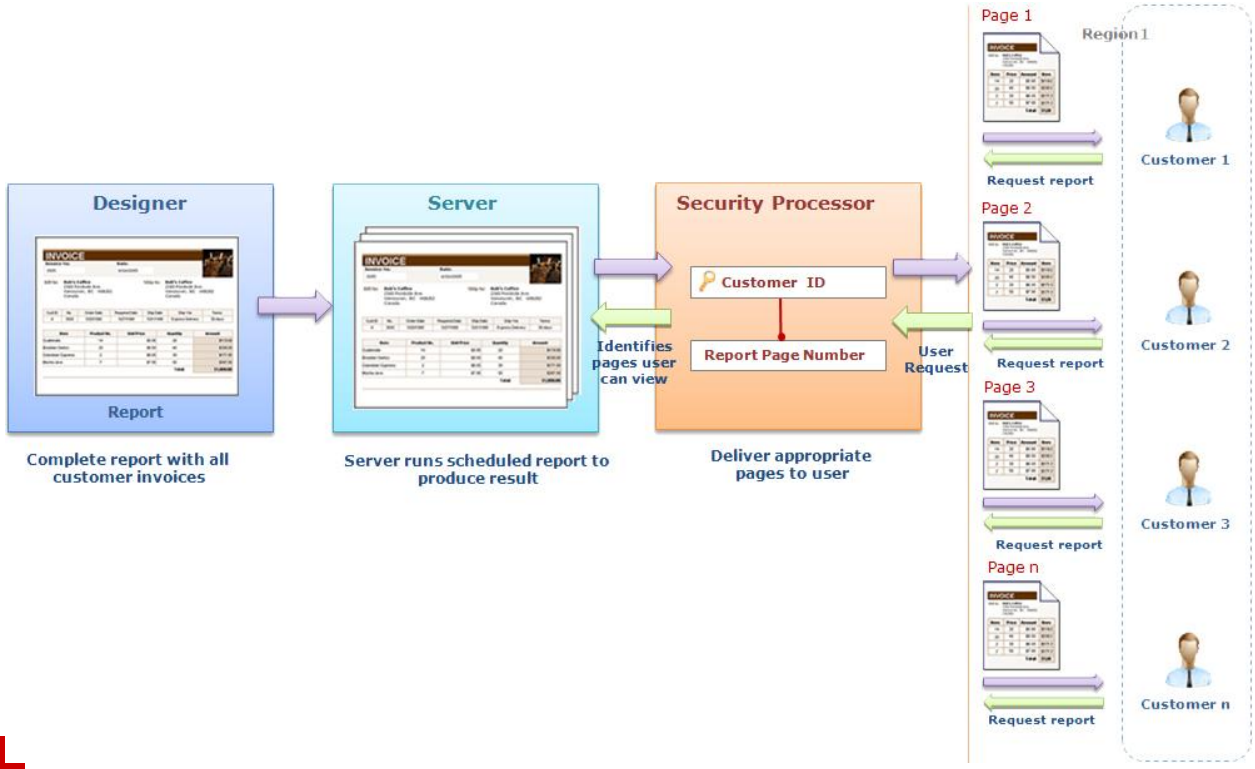

Illustration of Page Level Security

From the resource load perspective only one query is needed, only one set of data is transferred over the intranet, and only one template is needed to format the report. For IT the advantages are clear: a single report needs to be scheduled and then just one file needs to be managed, archived and purged. Permissions on the file are simplified since only a single file exists and only the JReport server needs to have access to read it.

With group level security, even if many users request their report at the same time JReport Server can quickly access the single report result file without any impact to the DBMS server, and displays the pages of data for the accessible groups to the users. Another advantage to page level security is that the reports can be delivered in HTML5 format to a browser, so the user can continue to filter, sort, drill down and interact with the report in ways not available with static formats in a bursting report. Still, if the user wants to export the report to PDF or other static formats and save it to his local computer he can do so, without any overhead to the IT department for managing resources and permissions.

JReport also supports a number of additional security features such as encryption and password access to PDF files, secure FTP to transfer reports using encryption to remote sites, and SSL encryption for HTML and HTML5 access to reports from a browser.

Summary

Security cannot be overlooked when comparing features of BI tools. It is smart to ensure that the BI tool that you choose has all of the options you need to not only manage the security of the report data, but also to manage the resources, file permissions and disk resources. When calculating the total costs over time, make certain the calculations include the subscription costs of purchasing the commercial versions from open source vendors, and the additional resource costs needed for tools which do not include multi-level bursting and page level security. JReport is the only product which provides all of these features out of the box without any additional costs: they are all included in the base product.

More Information

Schedule a Free Evaluation to Review Your Specific Security Needs
Request a Live Demo to See How JReport Security Can Work for You
Test Drive JReport

Contact Us

Tel:	+1 240-477-1000
Fax:	+1 240-465-0355
Website:	http://www.jinfonet.com
General Info:	info@jinfonet.com
Sales:	sales@jinfonet.com
Support:	support@jinfonet.com
Marketing:	marketing@jinfonet.com

About Jinfonet

Jinfonet Software is a company committed to delivering flexible, timely, and actionable information to all users across an enterprise via advanced visualization. Headquartered in Rockville, Maryland in the heart of the I-270 Technology Corridor, and equipped with a team of more than 160 expert Java developers, Jinfonet is the provider of the leading embedded Java reporting solution. Founded in 1998, and experiencing year-on-year growth since, Jinfonet is currently in its 11th release cycle of JReport.

About JReport

The JReport Product Suite is comprised of JReport Designer, JReport Server Live, and JDashboard. This comprehensive reporting software is the leading embedded reporting solution offering intuitive reporting from directly within existing applications. Featuring 100% Java architecture, The JReport Product Suite reaches millions of end users worldwide on a daily basis via the more than 25,000 downloads completed since the product's inception.

Jinfonet Software, 2275 Research Blvd, Rockville, MD 20850
Tel: +1 240-477-1000 | Web: www.jinfonet.com