tableau SOFTWARE

Tableau Server Security

Version 6.1

AUTHOR:
Marc Rueter, Sr. Director Technology
Consulting & Strategy

DATE:
July 2011

Today’s enterprise class systems need to provide robust security in order to meet the varied and dynamic needs of modern organizations. A system that starts at the departmental level with no data security may suddenly need to be deployed globally with strict and maintainable row- level security. An internal, intranet based analytic application may need to be shared externally with partners or even customers in order to grow or sustain business. For this reason, organizations of any size should seek a Business Intelligence platform vendor that easily meets their security needs with easy to deploy, easy to grow, worry free applications.

Tableau Server

As an enterprise Business Intelligence Platform, Tableau Server provides comprehensive and robust solutions for all aspects of security. Tableau balances a variety of sophisticated security requirements with ease of use and ease of implementation. The result is a solution that is quick and easy to deploy, while requiring zero customization, scripting or coding in Tableau Desktop or Tableau Server.

There are 4 main components to enterprise application security:

- Authentication – Access Security
- Permissions – Object Security
- Data – Data Security
- Network – Transmission Security

Together, these capabilities provide a complete security package that allows a single report or dashboard to serve the needs of a broad and diverse user base, whether internal to your organization or external on the Internet.

Tableau Server has what it takes to fulfill your Business Intelligence needs while delivering the highest security. Tableau Server has passed the stringent security requirements of customers in the financial services, government, and higher education sectors. Banks and investment firms deliver sensitive investment information directly to their clients. Colleges and Universities leverage Tableau Server to deliver personalized reports directly to students and faculty. Tableau Server is deployed by all branches of the military and other government agencies. The rest of this document describes how the n-tier Tableau Server provides comprehensive security.

1. Authentication - Access Security

The first level of security is to establish the user's identity. This is done to prevent unauthorized access and to personalize each user's experience. This process is typically referred to as 'authentication'. It should not be confused with 'authorization' which is covered in the section titled 'Permissions – Object Security'. Tableau Server supports 3 types of authentication : Active Directory, Local, and Trusted in addition to an option to allow anonymous (un-authenticated) access to the system.

For all authentication methods, Tableau Server ensures the security of credentials – even if SSL is not enabled. For more details see the Network – Transmission Security section of this document.

Like most modern systems, Tableau Server provides a personalized experience for users. For example, users can save changes to views, set defaults and add comments. Publishers can construct views that restrict data based on a user's identity. For this reason, Tableau Server must retain information about a user so the personalized experience is repeatable. Tableau does this by creating an account for each named user of the system.

Active Directory

When customers choose to use Active Directory for authentication, all usernames and passwords are managed by Active Directory. Tableau passes credentials to the Active Directory server, but it does not participate in the authentication process. Tableau supports the ability to automatically login users based on their current windows credentials (the credentials they used to login to the machine they are using). Note that this means that the user's credentials are being passed from their local machine, not from another system or portal that they may have logged in to.

For example, if a user logs into their local machine as 'MSmith' and then logs into a SharePoint portal as 'Mary', the credentials passed to the Tableau Server will be for 'MSmith'. In order to use the credentials from the SharePoint site ('Mary') for automatic login, the SharePoint portal must use the Tableau web part with Trusted Authentication.

Even though users and groups are managed by Active Directory, Tableau stores usernames and groups in its repository. Tableau does not store passwords when configured for Active Directory authentication. Users and groups can be synchronized with active directory either manually by an administrator or programmatically using the Tabcommand (Tabcmd) client.

Local Authentication

Tableau Server provides a built-in user management and authentication service called Local Authentication for organizations not using Active Directory or when deploying externally. When using Local Authentication, the Tableau Server is responsible for managing users, groups, passwords and the entire authentication process. User lists can easily be imported to the Tableau Server and most user management functions can be performed programmatically via Tabcmd. This means that provisioning Tableau users can be part of your automated provisioning process. Users can either manually login by entering their credentials when prompted or, when accessing content in a portal, via transparent Trusted Authentication.

Trusted Authentication

Tableau enables a simple and robust mechanism for external authentication that requires zero customizations to Tableau called Trusted Authentication. This method is far easier to implement than the complex and fragile Single Sign On (SSO) integrations. Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. For example, you may have your corporate wiki use Trusted Authentication to show dashboards to employees already signed onto the wiki, without requiring another sign on. When Tableau Server receives requests from these trusted web servers it assumes that the web server has already handled whatever authentication is necessary. The Tableau Server simply receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user's role, permissions and data restrictions.

To prevent invalid requests for tickets, The Tableau Server only accepts requests for tickets from trusted IP addresses. The request simply includes the username that has already been authenticated by the trusted system – this means that tickets will only be issued for licensed users. Each ticket can only be redeemed once by a web browser. And, that ticket must be redeemed within a specified amount of time.   By default tickets are only valid for navigating to views. This prevents users from navigating to other parts of the Tableau Server such as content listings and administration. All errors in this process will receive the same return code value of -1. This makes it even more difficult for untrusted systems to retrieve a ticket.

For deployments behind the firewall these measures are typically enough to provide adequate security. Many customers have found this mechanism to be so secure that they are comfortable disabling the limitation of tickets to only access Views. These customers choose to implement Tableau's unrestricted tickets capability. This allows their users to navigate through all the content they have access to during a session.

For external deployments or where security requirements are high, an additional restriction can be added to tickets: they can be redeemed only by a specified IP address. This means that the request for a ticket from the trusted web server must include the IP address of the client (web browser). The Tableau Server will consider the ticket valid only if it is being redeemed by the correct client IP address.

Setting up Trusted Authentication is straightforward. Customers typically have Trusted Authentication working in their environment within an hour or two. Several examples are provided for common application server frameworks as part of the Tableau Server installation. More details are provided in the Tableau Server Administrators Guide.

Guest User or Anonymous (un-authenticated)

Note: This option is only available with a core-based server license.

Tableau Server can be set up to allow anonymous access to views via a guest account. This is useful when content is being deployed to unknown user communities such as the public web or to communities where the identity of the user is not required such as a large intranet. The 'guest license' is available to allow users without an account on the server see and interact with an embedded view. To prevent accidental anonymous access to sensitive data, the ability to access the Tableau Server as a guest is disabled by default. When enabled, the guest license is assigned to an automatically generated 'Guest' user. Since 'Guest' users are anonymous, meaning there is no way of identifying who they are, Tableau provides a single Guest User. The Guest User can be granted permissions to access and interact with content using the full scope of roles, permissions and data security available in the Tableau Server. Anonymous users can load webpages containing an embedded visualization without logging in. Anonymous users cannot browse the repository; they can only access embedded views (URLs that have the :embed=true parameter set). For simplicity, if an anonymous user requests a view that does not have the embedded flag, Tableau Server will interpret it as a request for an embedded view. This means that URLs shared via email, or links on other web pages will be properly processed for anonymous users.

When a request for an embedded view is received by the Tableau Server, it first checks to see if the user is logged in (i.e. the request is accompanied by a login session cookie for a logon that has not expired). If the user is not actively logged in, then the request is processed as a Guest user.

The Guest user cannot be used when Active Directory authentication is set to 'enable automatic login' due to ambiguity in handling invalid credentials.

Logging Out

An often neglected area of authentication is terminating a session. Tableau Server has automatic login timeouts based on a period of inactivity (idleness). Administrators can change the default idle duration of the login.

When using 'Active Directory Authentication' with automatic login enabled, users are presented with a 'Change User' option rather than a log out button because once they logged out, they would automatically be logged back in. For other authentication scenarios users are presented with a 'Logout' button so they can manually log out when finished with their session.

For integrated environments, for example views embedded in a portal, it is useful to programmatically force a logout on the Tableau Server when the user logs out of the portal. This is easily achieved by calling a logout URL from the client: http://<Tableau Server>/manual/auth/logout

2. Roles and Permissions - Object Security

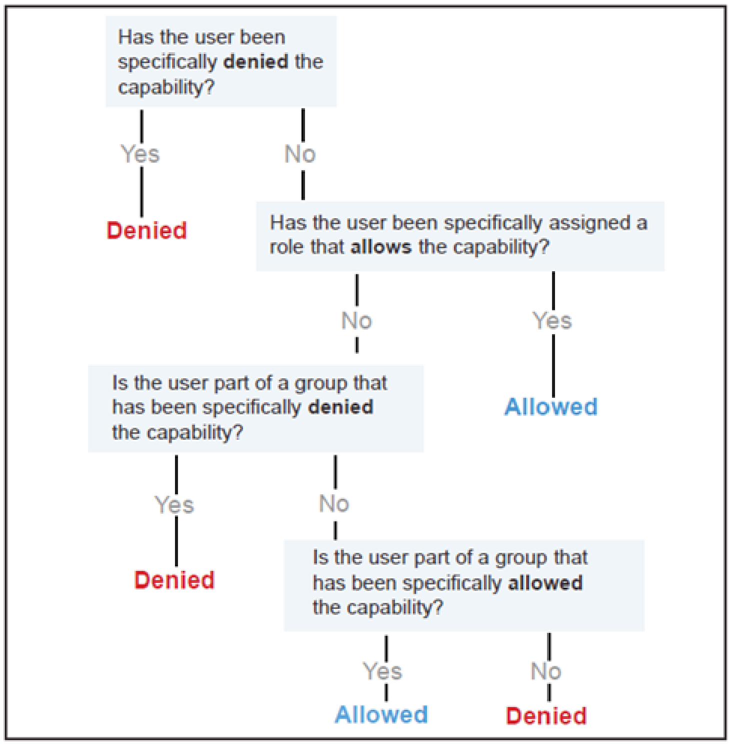
Once a user is properly authenticated and is granted access to the system, Permissions – or an access control list – allow control of what content or objects a user can access what actions a user or group can perform on that content. In Tableau, a role is a default set of permissions that is applied to content for users assigned that role. Roles are assigned to users for specific content and are not universal rights assigned to users for all content in the system. For example, a user is assigned the role of Interactor for a particular view – not for all content.

The maximum set of permissions a user can have is controlled by their license level and rights. For example, a user with a license level of Viewer cannot filter regardless of their role, group membership or being allowed that capability in a view. In this example, granting a user 'Viewer' rights does not grant them access to any content.

Together, these capabilities provide fine grained control of the content – both what users and groups can access as well as what actions they can perform on the content. Again, permissions control what projects, workbooks and views a user has access to and what they can do with those objects. Permissions do not control what data will appear inside of a view. What data users see is covered in the next section: Data – Data Security.

Permissions are controlled and evaluated at the content level. Initial permissions are set via a template mechanism. The initial permissions for a project are copied from the "Default" project. The initial permissions for a workbook are copied from the permissions for its project. The initial permissions for a view are copied from the permissions of its workbook. This is a one-time copy of the parent's permissions. Any changes to the permissions of the parent will not be automatically applied to the children unless the new permissions are actively 'assigned' to the contents. Any item can have permissions that differ from the parent – both granting permission as well as denying. For example, a user or group might not have permission to see a Project, but may have permission to see a View published to the Project. Tableau Server does not support hierarchical object permissions, however, it does provide an inheritance model for users and groups. If a user does not have a permission explicitly set, the setting will be inherited from the group/s that the user belongs to.

As you can see from the Permission Evaluation diagram, if a user or group is not explicitly granted a capability, it will be denied. If you leave the permission set to Inherited, the user or group's access to the view will be inherited by the group or project permissions. Again, if there is no explicit 'Allow' in the inheritance chain, the capability will be denied.


Has the user been specifically **denied** the capability?
Yes
Denied
No
Has the user been specifically assigned a role that **allows** the capability?
No
Yes
Allowed
Is the user part of a group that has been specifically **denied** the capability?
Yes
Denied
No
Is the user part of a group that has been specifically **allowed** the capability?
Yes
Allowed
No
Denied

Permissions with Respect to Projects

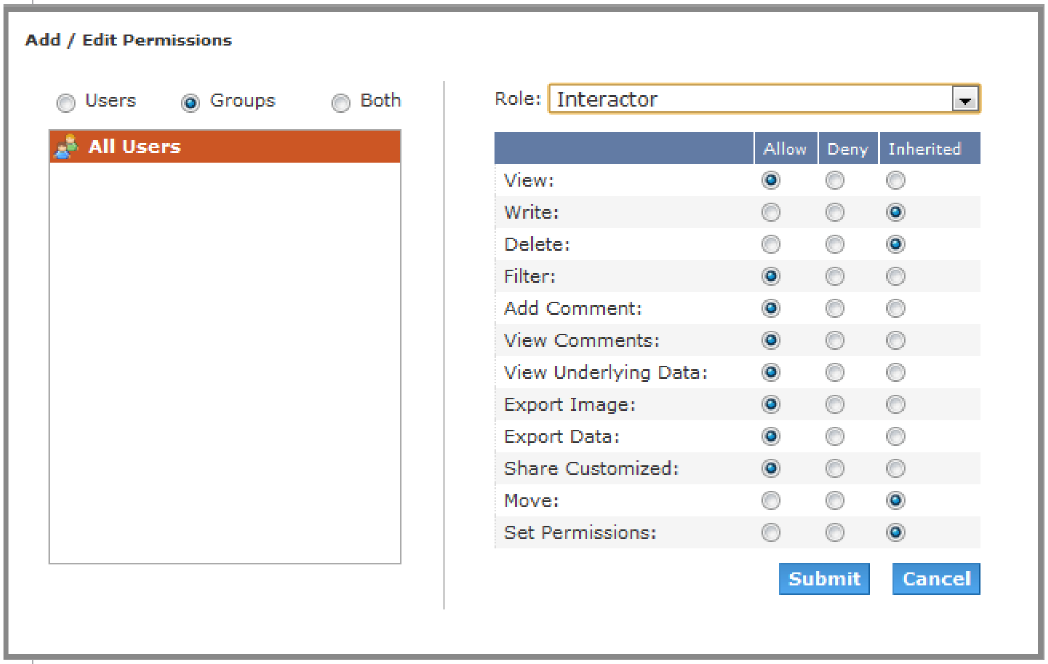
Projects control the default permissions for all workbooks and views published to the project. The default permissions for any content object can be overridden by users with appropriate permissions. For example, Publishers have the ability to fully control the content they publish.

Each new Project begins with a copy of the permissions from the default project. This is a one-time copy – changes to the permissions of the default project do not propagate to any existing projects. For this reason it is worthwhile to choose an appropriate permission strategy prior to creating projects.

Only Administrators (both Content and System Administrators) can create and modify projects and their permissions. Users with the 'Project Leader' permission can fully control all content in the project.

Permissions with Respect to Published Content

Published content includes Data sources, Workbooks and Views. Permissions include the typical content management actions such as view, create, modify and delete, but also what interactions a user can have inside of a view. This is an example of the permissions available for a single view:



Add / Edit Permissions

○ Users ● Groups ○ Both

All Users

Role: Interactor

	Allow	Deny	Inherited
View:	●	○	○
Write:	○	○	●
Delete:	○	○	●
Filter:	●	○	○
Add Comment:	●	○	○
View Comments:	●	○	○
View Underlying Data:	●	○	○
Export Image:	●	○	○
Export Data:	●	○	○
Share Customized:	●	○	○
Move:	○	○	●
Set Permissions:	○	○	●

Permissions do not control what data within a View a user can see. Data security is covered in the next section. Permissions are applied when a user searches for content and navigates through the Tableau Server UI.

Data source permissions are only applied when browsing the repository via search, directly navigating to the 'Data Sources' section of the Tableau Server or when importing Data Sources from Tableau Desktop. It is important to note that at runtime, Views published to Tableau Server do not maintain a relationship to published Data Sources – they use server managed Connections instead. This has several implications, one of which is that permissions on a Data Source have no impact on permissions on a View or Workbook. This means that if a user is denied access to a Data Source, but has permission to access a view, the permissions on the Data Source are ignored and the user can still run and interact with the view.

A Word About Connections:

Connections are automatically created by Tableau Server during the Publish process. This allows administrators to control connection attributes such as updating database credentials or migration to a new database server without having to manually edit each workbook that was published to the Tableau Server. Furthermore, multiple workbooks and data sources can share a Connection. This means that caches can be shared across workbooks to further reduce the load on your database server.

Permissions and Administrators

There are two types of administrators: Content Administrators and System Administrators. Content Administrators can manage users, groups, projects, workbooks and data connections. System Administrators have all the rights of a content administrator but they can also administer the server itself including maintenance, settings, schedules, and the search index. Administrator rights can only be assigned to users with the Interactor license level. The Publish right is automatically granted to all Administrators.

The 'Default' Project

If the Tableau Server is deployed in an open environment where knowledge and information sharing is key, then the permissions for the Default project should include 'All Users' group with its role set to 'Interactor'. Then users will automatically be able to publish to and consume content from new projects unless the default permissions are overridden.

If the Tableau Server is deployed in a restrictive environment where data security and access control is key, then the permissions for the Default project should be empty: Delete the permissions for all users and groups. Then, users and groups will need to be explicitly granted permission to publish and consume content in new projects.

Multi-Tenant Deployments

Many Tableau customers use Groups and Projects to support multiple external parties (tenants) on a single Tableau Server. Tableau Server's security is robust enough to meet the demands for deployments in Finance, Health-care and other situations where one client cannot see another client's data under any circumstances.

However, it must be noted that users with Administrator or Publisher rights on the Tableau server will be able to see all users of the Tableau Server. Additionally, Administrators can see all content published to the Tableau Server, but this does not mean they will have access to all data used by the Tableau Server. Since data access is separate from permissions, it will be covered in the next section.

3. Data – Data Security

Data security is an increasingly important consideration in any enterprise - especially those organizations that need to meet regulatory requirements or those that are deploying Tableau Server externally. It is critical that Tableau provide a robust set of capabilities that allow customers to build upon existing data security implementations or augment deficient systems. The goal is to have a single place to enforce data security.

There are three main options for data security:

1. Implement the security solely in the database(database authentication)
2. Implement security solely in Tableau
3. Create a hybrid approach where user information in Tableau Server has corresponding data elements in the database.

Tableau Server supports all three types of security, but typically the hybrid approach is used. When leveraging database security it is important to note that the method chosen for authentication to the database is key. This level of authentication is different from the Tableau Server authentication discussed above. I.e. when a user logs into the Tableau Server, they are not logging into the database. This means that Tableau Server users will need to have credentials to log in to the database in order for the database level security to be applied for them. When leveraging Tableau's User Filters, the Tableau Server users will not need to have additional credentials to access their secure data.

In all cases, the result is that a single view or dashboard is published that provides secure, personalized data and analysis to a wide range of users.

To further protect your data, the credentials used to access the database only need read access. This prevents publishers from accidentally changing data and also improves query performance in many cases. Tableau recommends granting the database users permissions to create temporary tables.

Database Authentication

If the data is extracted using Tableau's fast Data Engine, then no options for database authentication will be available for end users. When automatically refreshing or incrementing extracts, a single set of credentials will be used to generate a single extract for each data source (either the "Run as User"[1] or credentials embedded in the workbook).

Views published to the Tableau Server are dynamic in accessing the database to retrieve current data. Whenever a view is opened, if the data source for it is a database that requires a login (as opposed to something like an Excel workbook or a text file) the Tableau Server needs to know what database username and password to connect as to retrieve the data. Tableau Server has several options and settings that work together to specify what database username and password will be used for accessing the data. It is important to keep the distinction clear between the Tableau Server login technique which is used to gain access to the Tableau Server itself, and the database login that may be required for each view that is published to your Tableau Server. The table below summarizes the alternatives. The column headings refer to the technique used when

	Windows Integrated Security (NT Authentication)	Username & Password Prompt	Embedded Username & Password at Publish
Tableau Server logs into the database as:	"Run as User" of Tableau Server.	Each user is prompted for their database credentials, which they can choose to have saved.	The database credentials specified by the author when the view originally published. The Tableau Server user is not prompted for any credentials.
Tableau Server leverages the existing user-based data security built into my database:	No. All users share the same database login.	Yes, the individual user identity is known to the database.	No. All users share the same database login,.

[1] This is the 'Server Account' as defined in the Server Administrator Guide

If the 'Saved Passwords' option is turned on in the Tableau Server Administration panel, then a user only needs to enter their credentials one time per data source. These data source credentials are then stored in the Tableau Server and re-used for that user's next connection to the same data source. Note that these credentials are separate from those used to log in to the Tableau Server. Tableau always encrypts any password that is stored in the Tableau Server repository.

Details on Authentication Options

All discussions below are with respect to the database security itself and do not impact the ability to use 'User Filters' in Tableau Server. When publishing, authentication options can be set for each data source in the workbook.

Windows Authentication

The Tableau Server uses the 'Run as User' credentials to connect to the database. All users of the Tableau server will share this same connection information for the database. This does not use the credentials of the publisher or the credentials of the user logged in to Tableau Server. This option requires the database to leverage Windows Integrated Security. This is very common for SQL Server or SQL Server Analysis Services implementations. The default 'Run as User' for the Tableau Server is the Network Authority user. By definition, this Network Authority account does not have rights to connect to a database.

User Name and Password (not embedded)

Each user of the Tableau Server will be prompted to log in to the database with their database specific user name and password. If you already have database security set up, this is a good option to make sure that the security is honored by the Tableau Server. Rather than being prompted for credentials each time a view is run, there is an optional setting to allow Tableau Server to remember the user's credentials. The passwords are encrypted and stored in the Tableau Server repository.

Embedded Credentials (not for use with Windows Authentication)

When Embedded Credentials are enabled, Tableau Server will remember the username and password that were used by the original author of each workbook. At publish time, the author simply enters a set of credentials for the database – a username and password. All users of the Tableau server will share this same connection information for the database. The Tableau Server uses the published credentials to connect to the database to retrieve data.

Impersonation

For SQL Server data sources, Tableau Server supports impersonation of users when running queries. This allows Tableau to leverage security that you have already implemented in SQL Server. Tableau will either connect to the database using the 'Run as User' or with embedded credentials. But, all queries will be executed as though another user had connected. Tableau impersonation is designed to work in conjunction with SQL Server Implementations that adhere to Microsoft's best practices for Context Switching using Database Impersonation.

Query Banding

For Teradata data sources, Tableau Server supports inserting user information into the Query Band. This can enable data to be restricted based on database rules or a variety of other Teradata Workflow rules to be processed.

Data Security – User Filters

User Filters are a Tableau Server capability that enable row-level data security. Tableau uses dynamic data filtering based on the username, group membership and other attributes of the current user. When executing the view, Tableau Server will append all queries to the database with an appropriate 'where' clause to properly restrict the data for the current user's request. In general, user filters reduce the ability for the Tableau Server to re-use caches across multiple users. User Filters can be used with all data sources. They can also be used against data that has been extracted into Tableau's Fast Data Engine.

Tableau views can be constructed to include a calculated field using a variety of variables based on the user name or group membership of the user that is logged in to the Tableau Server.

For example, an Order table may contain customer information (CustomerID), sales person information (employeeID) and details about the order. A single calculated field can be added to the view to enable user filtering: username()=CustomerID OR username()=employeeID. Now a single view or dashboard can be published to the Tableau Server that will securely deliver data externally to customers and internally to sales people. Customers will only see orders they have placed while sales people will only see orders they have sold.

The benefit of this approach is that no maintenance needs to be done to the Views as new users and data are added to the system. The filtering rules are built into the Views and the database provides the keys for those rules to process.

If there is no suitable content in the database to programmatically identify which data to provide to which user, then a manual user filter can be created. This type of user filter is processed the same as calculated user filters, but does not dynamically adapt to new users and data elements. Therefore, additional maintenance to the Views is required.

Extract Security

When optional data extracts are used, the Tableau Server is responsible for storing and processing data used in Views and Workbooks. The data is stored on the file system of the Tableau Server in Tableau's proprietary Fast Data Engine format. The data is in an encoded, compressed, binary format. The metadata that describes the extracts are stored in plain text. This means that the data is not human readable, however some descriptions of the data such as data types, field names etc. can be discerned. To protect these files, they are stored in the 'Program Data' directory with access controls restricted to the "run as user" of the Tableau Server. The extract data files themselves are not encrypted on disc.

Just like the other databases that Tableau connects to, the Data Engine extracts are not directly queryable from the Tableau Server user interface. This means that users cannot compose SQL, MDX or any other syntax to interact directly with the database. This helps to prevent unauthorized access, SQL injection and other malicious attacks.

Repository Security

Tableau Server has an internal repository database that stores information about the system (usage statistics, users, groups, permissions etc.) as well as content (workbooks, views, comments, tags etc.). The repository does not store the raw data or extracted data used in Tableau views and workbooks.

By default, the repository database does not allow external connections. This means that access to the information stored in the repository is restricted to the components of the Tableau Server. However, for customers that would like direct access to this information, the repository can be configured to allow external connections. External connections are restricted to read only views of the data to prevent malicious use and accidental changes to the Tableau Server content or configuration.

4. Network – Transmission Security

For many internal deployments, network security is provided by preventing access to the network as a whole. However, even in these cases it is important to securely transmit credentials across the network. For external deployments, transmission security is often critical to protect sensitive data, credentials and to prevent malicious use of the Tableau Server. Regardless of your situation, Tableau Server has robust capabilities.

There are 3 main network interfaces to the Tableau Server: Client to Tableau Server, Tableau Server to Database and Communication between Tableau Server Components. Each one of these interfaces is described below. In addition to these broad security capabilities, Tableau has pays special attention to the storage and transmissions of passwords at all layers and interfaces. A variety of encryption techniques are used to ensure security from browser to server tier to repository and back, even when SSL is not enabled. For more details on the encryption technologies in use, please see the Tableau Knowledge base.

Client to Tableau Server

Client in this case means a web browser, Tableau Desktop or Tabcmd (Tab Command). By default these communications use standard HTTP requests and responses which are suitable for most internal deployments. For external or other sensitive deployments, Tableau Server can be configured for HTTPS (SSL) with customer supplied security certificates. When the Tableau Server is configured for SSL, all content and communications between clients are encrypted and use the HTTPS protocol.

When the Tableau Server is configured for SSL, the browser and SSL library on the server negotiate a common encryption level. Tableau uses OpenSSL as the server-side SSL library and it is pre-configured to use currently accepted standards. Each web browser accessing Tableau Server via SSL uses the standard SSL implementation provided by that browser. This works even in embedded situations and results in a seamless experience for the end user with no security warnings, pop-ups or exceptions.

Passwords are communicated from browsers and from Tabcmd to the Tableau Server using public/private key encryption. The Tableau Server sends a public key to the browser, which uses the key to encrypt the password for transmission. Each encrypted transmission uses a key one time before it is discarded. This means that passwords are always secured regardless of the use of SSL.

Tableau Desktop uses Microsoft's WinINet API for communication with the Tableau Server using either HTTP or HTTPS. For communicating passwords, strong encryption is used to encrypt passwords sent back to the server.

Communication Between Tableau Server and the Database

Tableau Server makes dynamic connections to databases to process result sets and/or to refresh extracts. Tableau uses native drivers to connect to databases whenever possible. Tableau relies on a generic ODBC adapter when native drivers are not available. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau. However, since Tableau Server to Database communication is typically behind a firewall, most customers choose to not encrypt this communication.

Furthermore, the Tableau Server can be deployed in an n-tier configuration with the web server in the DMZ and the VizQL Server (responsible for communication with the database) behind the firewall. This further protects your data and further reduces the need for encryption.

Tableau only recommends encrypting the database transport layer when you have chosen to implement Tableau with database communications passing through the public internet.

Communication Between Tableau Server Components

This section only applies to distributed deployments of the Tableau Server.

There are two aspects to communication between Tableau Server components: trust and transmission. Each server in a Tableau cluster uses use a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. The primary server is the only machine in the cluster that accepts requests from 3rd parties (clients), all other machines in the cluster only accept requests from other trusted members of the cluster. Trust is established by a whitelist of IP address, port and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other. With the exception of license validation and accessing the repository, transmission of all internal communication is performed via HTTP.

When passwords are transmitted within the cluster, a key is used to encrypt the passwords transmitted between Tableau Server components (e.g., between Application Server and VizQL Server processes). Each encrypted transmission uses a key one time before it is discarded.

Other Topics

Platform Robustness

Due to the outward facing nature of extranets, Tableau Server has many built in safeguards to maintain integrity in an exposed environment. Among these are: a single port is needed for all client communication; a proxy server can be placed in front of the Tableau Server; Tableau Server employs sophisticated anti-spoofing and hi-jacking mechanisms; the multi-tier architecture of Tableau Server helps prevent SQL injection attacks to the raw data; distributed components of Tableau Server employ a stringent trust requirement. Tableau actively tests for vulnerabilities and quickly responds to new threats with monthly updates to address issues.

Encryption

Tableau Server makes use of a number of encryption technologies to keep sensitive information secure. Specifically, Tableau Server encrypts passwords when they are transmitted between different server components and when they are stored in the Tableau Server repository. These encryption strategies are used for all Tableau Server installations and the private keys are compiled into the Tableau Server executables. In addition, Tableau Server can optionally be configured to use SSL encryption (HTTPS) for all communication between the browser and the server. Refer to the Tableau Server Administrator Guide for more information about configuring the Tableau Server.

Summary

The Tableau Server provides a comprehensive set of security capabilities to suit your deployment needs. Tableau has proven public facing deployments at numerous customer sites and even more internal deployments on secure networks. Tableau uses modern industry standards as a baseline and is responsive to threats and known issues. From row level security, to secure websites and every security detail in between, Tableau has the answers to your security questions built in to our products.

Tableau Software
837 North 34th Street, Suite 400
Seattle, WA 98103 U.S.A.

Produced in the United States of America
July 2011