Tableau Server platform security

Implementing the four tenets of enterprise security

Contents

Introduction

Tableau is a modern enterprise analytics platform that enables self-service analytics at scale through governance. Security is the first and most critical part of a data and content governance strategy. Tableau Server provides the comprehensive features and deep integration to address all aspects of enterprise security. Tableau helps organizations promote trusted data sources so all users have access to the right data to make the right decisions quickly. As the promise of a single central EDW wanes and data proliferation continues to accelerate powered by the cloud, managing consistent security across all the various platforms becomes crucial for your enterprises.

Overview

There are four general components to enterprise application security, and this paper will cover them in greater depth for Tableau Server:

1. Authentication

2. Authorization

3. Data security

4. Network–transmission security

When properly implemented, these four components address all enterprise security requirements and enable a broad user base to access trusted data and build reports, dashboards, and collaborative analysis. Business users trust the information provided by a secure data and analytics platform, encouraging widespread usage and realizing more value from data. External access to the same analytics platform can be made available to clients and contractors while still fulfilling enterprise security requirements.

Tableau Server has passed the stringent security requirements of customers in the financial services, government, healthcare, and higher-education sectors. Banks and investment firms deliver sensitive confidential investment information directly to their clients. Colleges and universities use Tableau Server to deliver personalized reports directly to students and faculty. Tableau Server is deployed by all branches of the military and many state and federal government agencies. This document describes how Tableau Server provides comprehensive security at enterprise scale.

1 Authentication

Tableau Server supports several forms of industry standard authentication including Active Directory, LDAP, Kerberos, OpenID Connect, SAML, Trusted Tickets, and certificates. Tableau Server also has its own built-in user identity service called Local Authentication.

Once a user signs in, Tableau Server provides a customizable experience including language and locale, a personalized start page, and an overview of personally-authored content. Tableau Server retains user information across sessions for a consistent personalized experience. Tableau does this by creating and maintaining an account for each named user on the system. In addition, authors and publishers can use server-wide identity information to control the level of authorization other users have to the underlying data for views they publish.

User identity

As mentioned above, you can manage user identities with Active Directory or by storing them within the server using Local Authentication. We describe the difference between these two methods of managing user authentication below.

Active directory

When customers choose to integrate Tableau Server with Active Directory as the identity store, Active Directory manages all usernames and passwords.

Even though users and groups are centrally managed by Active Directory, Tableau Server stores a copy of the usernames and groups in its own 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 tabcmd command-line utility or REST API.

Local authentication

Tableau Server also contains a built-in user management and authentication service called Local Authentication. This method is used by organizations who choose not to use Active Directory or who are deploying to clients external to AD. When using Local Authentication, Tableau Server is responsible for managing users, groups, and the entire authentication process. The administrator has the option of storing passwords on Tableau Server. However, the option to delegate passwords and user information to an external service, such as OpenID or SAML, is also an option. User lists can easily be imported to Tableau Server, and most user management functions can be performed programmatically via tabcmd or REST API. This makes it easy to provision Tableau users as part of your automated provisioning process.

LDAP

Tableau Server on Linux introduces support for authenticating to any LDAP provider with Windows support coming soon. All of the same authentication and user management features available with an Active Directory server are available for any Directory Service that supports the LDAP protocol and any of the following authentication mechanisms: GSSAPI, simple bind, simple bind with Kerberos. Work with your IT department to determine what works for you.

Single sign-on and integration with external authentication services

Tableau Server supports several types of single sign-on (SSO) solutions as well as mutual SSL (client certificate authentication).

Mutual SSL provides a secure automatic sign-in experience with Tableau across all devices. With mutual SSL, when a client (Tableau Desktop on Windows, a web browser, or tabcmd.exe) with a valid certificate connects to Tableau Server, Tableau Server confirms the existence of a valid client certificate and automatically signs in the user with the username it finds in the certificate.

With SSO, users don't have to explicitly sign in to Tableau Server. Instead, the credentials they use to authenticate with other external authentication services (for example, signing in to their corporate network) can be used to seamlessly authenticate them onto Tableau Server without prompting a log in screen. SSO establishes the user's identity externally and maps it to a user identity defined in the Tableau Server identity store.

When you configure Tableau Server for use with an external authentication service for SSO, the external authentication service handles all authentication. However, Tableau Server will manage user access to Tableau resources based on the site roles stored in the identity store. See the authorization section below for more details.

Tableau Server supports integration with the following external authentication services:

- **SAML:** You can configure Tableau Server to use SAML (security assertion markup language) for SSO. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity. You can use SAML to access Tableau Server regardless of your Active Directory or local authentication configuration. You can also configure Tableau Server to use a different SAML IdP for each site, known as Site-Specific SAML.

- **Kerberos:** If Kerberos is enabled in your environment and Tableau Server is configured to use Active Directory authentication, you can provide users with access to Tableau Server based on their Windows identity. You cannot use Kerberos if your Tableau Server is configured for local authentication.

- **Integrated Windows authentication:** If you have Tableau Server configured with Active Directory authentication, you can enable automatic logon. Automatic logon uses Microsoft SSPI to sign in users based on their Windows username and password. Users are not prompted for credentials, which creates an experience similar to single sign-on (SSO) and Kerberos.

- **OpenID:** OpenID Connect is a standard authentication protocol that lets users sign in through a compatible identity provider. After they've successfully signed in to their identity provider, they are automatically signed in to Tableau Server. To use OpenID Connect with Tableau Server, the server must be configured to use local authentication; Active Directory authentication is not supported.
- **Trusted authentication:** Trusted authentication (also known as trusted tickets) lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from a trusted web server, it assumes that the web server has already handled the necessary authentication. Tableau Server 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 and permissions.

Guest user or anonymous access

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

Tableau Server can be set up to allow anonymous access to views via a guest account. This is useful for deploying content to large user communities such as the public web or to communities where the identity of the user is not required, such as a corporate intranet. The guest license allows users without an account on Tableau Server to see and interact with embedded views.

To prevent accidental anonymous access to sensitive data, the ability to access 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 to identify who they are, Tableau provides only a single guest user because they are universal.

Anonymous users can load webpages containing embedded visualizations without ever having to log in to Tableau Server, but you can choose to require credentials to access the intranet or the page hosting the view. 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 linked from other web pages will be properly processed for anonymous users and made accessible. Note that only guest accessible views (as defined in permissions) will be rendered for anonymous users; any view restricted from guest users will not render regardless of the "embed" flag.

Guest user permission to content can be controlled with the full scope of roles, permissions, and data security available to all other user types on Tableau Server. When Tableau Server receives a request for an embedded view, 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, if enabled.

Guest user access will not work 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 session timeouts based on length of inactivity. Administrators can change the default length of the idle duration timeout. Tableau Server also allows an absolute session timeout to be configured.

When using Active Directory authentication with automatic login enabled, users have a "switch user" option rather than a "sign out" option. This is because they would be automatically logged back in if they initiated a log out. For all other authentication scenarios, users get a "sign out" option so they can manually log out when finished with their session.

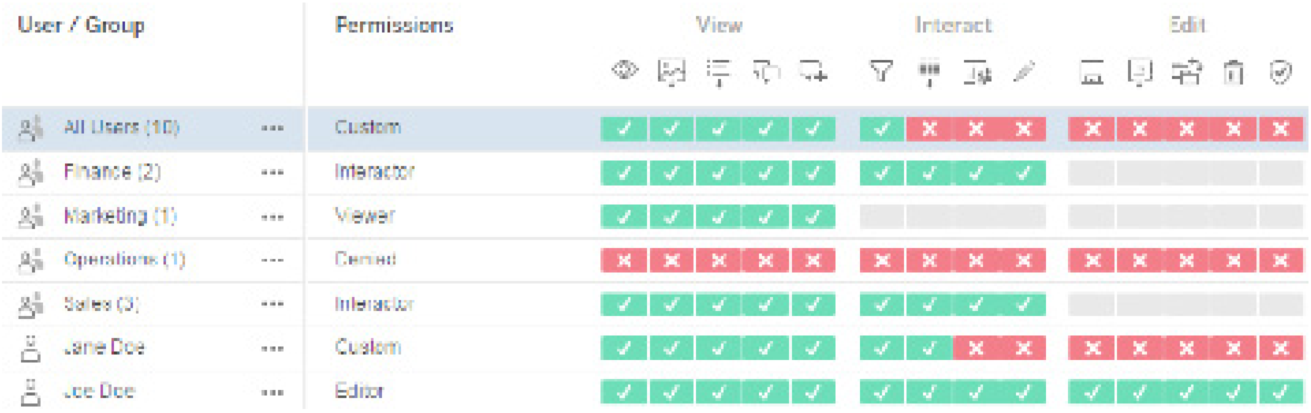
For integrated environments, such as views embedded in a portal, it is useful to programmatically force a logout on Tableau Server in addition to the logout for the portal. You can easily do this by calling a logout URL from the client: `https://<Tableau Server>/ manual/auth/logout.`

2 Authorization

Once you properly authenticate a user and grant them access to the system, the next step is to authorize what content and server permissions they have. In Tableau Server, site role and permissions provide administrators with fine-grained control over what data, content, or objects a user can access and -- what actions a user or group can perform on that content. These actions are often referred to as capabilities and include the ability to view and interact, add comments, save workbooks, and connect to data sources, among others.

You can also group users to apply permissions in batches more easily. Tableau Server provides you the flexibility to set permissions (allow, deny, or unspecified/inherited) on each piece of content (project, data source, workbook, and individual views within workbooks) and for specified users/ groups. When permissions are not explicitly set on a piece of content, Tableau will apply a default set of permissions. These default permissions will depend on the default settings at the time the content was created and are inherited from the parent of that content. Permissions do not control what data will appear inside of a view. Controlling what data users see is covered later in the section on Data Access Security.

In the example below, members of the operations group have been explicitly denied all capabilities for the sample view. On the other hand, Joe Doe has all capabilities allowed on this particularly view. Members of the marketing team have been given permissions to view the content, but the capabilities around interacting and editing the content are left unspecified. This means that Tableau Server will check up the chain, first to the permissions of the workbook and then the project, to see if these permissions have been allowed for this group. If not, these permissions will be implicitly denied.



User / Group		Permissions	View					Interact				Edit				
All Users (10)	•••	Custom	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Finance (2)	•••	Interactor	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Marketing (1)	•••	Viewer	✓	✓	✓	✓	✓									
Operations (1)	•••	Denied	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Sales (3)	•••	Interactor	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Jane Doe	•••	Custom	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Joe Doe	•••	Editor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 1. Setting customized permissions for groups and users based on content.

Default permissions and inheritance

Tableau sets initial permissions for content via a template mechanism. It copies the initial permissions for a project from the default project. It is important that you set the permission on the default project to be appropriate for your organization's security model. If you deploy Tableau Server in a self-service environment where knowledge and information sharing is encouraged, also known as an open permissions model, then the default project permissions should include the "All Users" group and be set to the Interactor permission role template. Users can then by default browse the server and interact with published views, only being limited in their access to workbooks that have custom permissions defined. If you are deploying Tableau Server in a closed permissions model where data security and access control is required, then the permissions for the "All Users" group in the default project should be none. This will remove all permissions for users and groups by default. Users and groups will then need explicit permission to publish and consume content in newly-created projects.

Content permissions model

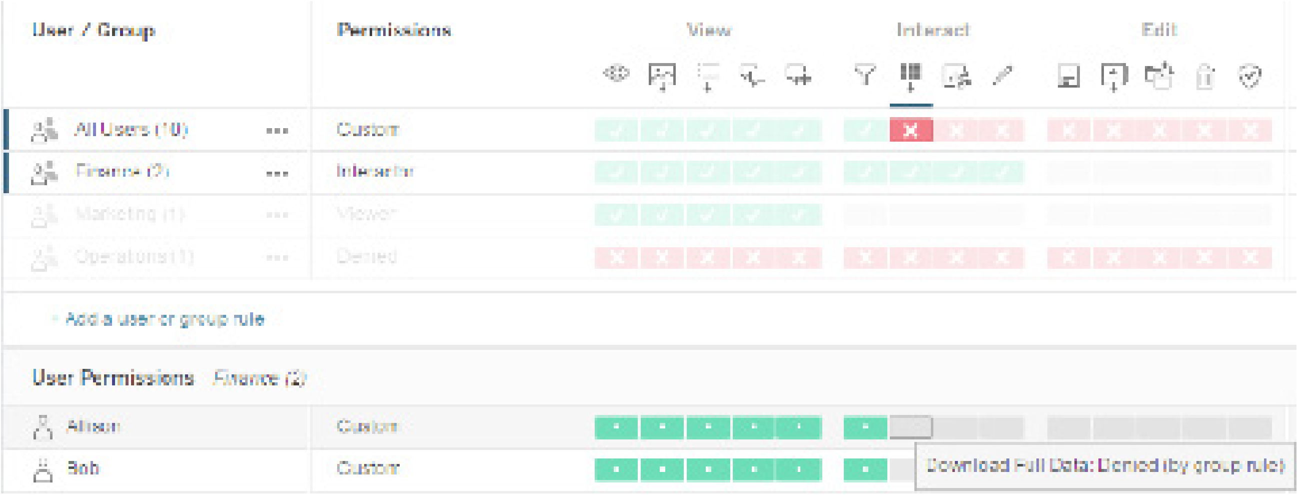
Published content includes data sources, workbooks, and views. Content permissions include the typical content management actions such as view, create, modify, and delete. They also include the interactions a user can have inside of a view. Permissions are also applied when a user searches for content and navigates through the Tableau Server UI.

Content permissions do not maintain hierarchy; rather, the initial permissions are copied from the parent's permissions at the time the item is first generated. Tableau Server also copies the initial permissions for a view from the permissions of its parent workbook. Any changes to the permissions of the parent will not automatically reapply to the children, unless the content is manually refreshed and permissions are redefined. Content may have different permissions than its parent. These can be more stringent or more lax, as defined by the author.

User permissions model

Unlike the permissions model for content, Tableau Server provides an inheritance model for permissions on users and groups. If a user does not have a particular permission explicitly set, the setting will be inherited from the group(s) that the user belongs to. In the Tableau Server Permissions Manager view, this appears as unspecified permissions or gray boxes (see fig. 1 & 2). If a user and group are not explicitly granted a capability in the inheritance chain, the capability will be denied. Changes to group permissions will propagate to all individual users automatically.

One useful tip for seeing the resulting permissions for a user or group is to select the group or user from the permissions page and look at the user permissions area at the bottom. This lets you see the actual permissions for each individual user after applying the group's inheritance settings. Hovering over a specific capability also provides information about the capability's name, the resulting setting, and how the results are determined.



User / Group	Permissions	View					Interact				Edit				
All Users (10)	Custom	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Finance (2)	Interactor	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Marketing (1)	Viewer	✓	✓	✓	✓	✓									
Operations (1)	Denied	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Add a user or group rule

User Permissions Finance (2)

Allison	Custom	🟩	🟩	🟩	🟩	🟩	🟩								
Bob	Custom	🟩	🟩	🟩	🟩	🟩	🟩								

Download Full Data: Denied (by group rule)

Figure 2. Viewing the resulting permissions of an individual user.

Tableau Server Permissions

Projects

Projects control the default permissions for all workbooks, views, and data sources published to the project. Only site and server administrators can create and modify projects and their permissions, while users with the "project leader" permission can fully control all content and permissions within their projects. Users with appropriate permissions can override the default permissions for any piece of content. For example, publishers have the ability to fully control access permissions to the content they publish. When admins require more control over the permissions within a specific project, they have the ability to define and restrict permissions of that project. Locking permissions within the project means all content published to that project uses the default permissions that the administrator set on the project. Content owners are then unable to change permissions, either on the server or during the workbook publishing process.

Permissions Template	Descriptions
Viewer	Allows the user or group to view the workbooks and views in the project.
Publisher	Allows the user or group to publish workbooks and data sources to the server.
Project Leader	Allows the user or group to set permissions for all items in a project.
None	Sets all capabilities for the permission rule to **Unspecified**.
Denied	Sets all capabilities for the permission rule to **Denied**.
Data Source Connector	Allows the user or group to connect to data sources in the project.
Data Source Editor	Allows the user or group to connect to, edit, download, delete, and set permissions for a data source in the projects. They can also publish data sources. Owners of published data sources can update connection information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.

Whether you lock permissions or allow content owners to manage permissions themselves, it is up to the administrator and the requirements of the project itself. Some projects may have locked permissions while others are left open. Permissions can easily be modified in the future as needs change.itself. Consider that it might make sense to lock permissions in some projects, but leave others open. Permissions can easily be modified in the future as needs change.

Workbooks & Views

The list of capabilities and the available permission role templates vary depending on whether you are setting permissions for a workbook or a view. For information about capability definitions, see Permissions Reference.

Permissions Template	Descriptions
Viewer	Allows the user or group to view the workbook or view on the server.
Interactor	Allows the user or group to view the workbook or view on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other permissions are inherited from the user's or group's project permissions.
Editor	Sets all capabilities for the rule to **Allowed**.
None	Sets all capabilities for the rule to **Unspecified**.
Denied	Sets all capabilities for the rule to **Denied**.
Custom	Administrator-defined rule for the selected combination of capabilities

Data Sources

Data source permissions provide another layer of security for both Tableau Desktop and Tableau Server users.

A user granted the "connect" permission for a data source can use Tableau Desktop to run queries to that data source through the Data Server component of Tableau Server. The user can either provide his or her own credentials or, if included, the saved credentials of the original author. This means Tableau Desktop users do not need to install database drivers on their machines, download data, or even have individual database credentials to run live queries against a data warehouse or a Tableau data extract. Data Server acts as a proxy without the need for direct connectivity to the database.

Permissions Template	Descriptions
Connector	Allows the user or group to connect to the data source on the server.
Editor	Allows the user or group to connect to, download, delete, and set permissions of data sources on the server. They can also publish data sources, and as long as they are the owner of a data source they publish, they can update connection information and extract refresh schedules. (The latter two capabilities are no longer available if an administrator or project leader changes data source ownership.)
None	Sets all capabilities for the permission rule to **Unspecified**.
Denied	Sets all capabilities for the permission rule to **Denied**.

Additionally, views using published data sources on Tableau Server can only be accessed by users that have both permission to the view and the underlying data source (either "view" or "connect" permissions for the data and view). However, if the publisher of the view has chosen to embed their credentials in the data source, users with permission to see the view can also connect to the data source on behalf of the publisher. To learn more about Data Server, please watch our Data Server video.

A word about connections

Tableau Server automatically creates data connections during the publishing process for both workbooks and data sources. This allows administrators and data source owners to control connection attributes separate from the view. This allows updates to credentials or migration to new database servers without requiring to manually edit each individual workbook. Furthermore, multiple workbooks and data sources can leverage a single connection, increasing performance and reducing duplication. This also means cached data is shared across workbooks to further reduce the load on your database server.

Permissions and administrators

There are two types of administrators: server administrators and site administrators. Server Administrators have full access to all server and site functionality, all content on the server, and all users. They can also configure the entire server cluster, including managing sites, users, maintenance, settings, schedules, and the search index. Site administrators can manage users, groups, projects, workbooks, and data connections within a site. Optionally, site administrators can add users to the site for delegated administrative scenarios.

All administrators automatically have the publishing privilege. Administrators can also create additional administrators at their same level.

Multi-tenant deployments

While the use of groups and projects is a common way for administrators to organize and permission content within an organization, the most common practice for supporting multiple external parties (tenants) on a single Tableau Server is through the use of sites. In fact, this is how Tableau Online, Tableau's hosted software-as-a-server (SAAS) offering is implemented. The content (workbooks, data sources, users, etc.) within each site are siloed from all other content on that instance of Tableau Server. Another way to say this is that Tableau Server supports multitenancy by allowing server administrators to create multiple sites on the server for different sets of users and content. All server content is published, accessed, managed, and controlled on a per-site basis. This means data sources and connections cannot be shared across sites. This functionality makes Tableau Server's security robust enough to meet the demands for deployments in finance, healthcare, education, and other institutions where a company's clients cannot see other client's data under any circumstances.

However, it must be noted that users with administrator or publisher rights on Tableau Server will be able to see a list of all users of Tableau Server (since they set role permissions for new content). Additionally, server administrators can see all content published to Tableau Server, but this does not mean they will have access to all the data used by Tableau Server since data access is separate from content permissions. This will be covered in more depth in the next section.

For more information about permissions on Tableau Server, see Tableau Server: Everybody's Install Guide.

3 Data Access Security

Data access security is of utmost importance in every enterprise, but particularly so for organizations with federal regulatory requirements and those that are deploying Tableau Server to external clients. It is critical that Tableau provides the robust capabilities to allow customers to build upon their existing data security implementations and augment any pre-existing deficient systems. The goal is to have a single place to enforce data security regardless of whether users are accessing the data from published views on the web and mobile devices, or accessing the data through Tableau Desktop.

There are three main approaches to data security:

1. Implement the security solely within 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 approaches, but customers often favor the hybrid approach for its simplicity and flexibility, especially when using multiple disparate data sources.

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 separate from the Tableau Server authentication discussed above (i.e. when a user logs into Tableau Server, he or she is not yet logging into the database). This means that Tableau Server users will also need to have credentials to log into the database for the database-level security to apply. To further protect your data, Tableau only needs read access credentials to the database, letting you limit users' access to read-only. This prevents publishers from accidentally changing the underlying data and can lead to improved query performance in many cases. Alternatively, in some cases it is useful to give the database user permission to create temporary tables. This can have both performance and security advantages because the temporary data is stored in the database rather than in Tableau. There is a tradeoff between granting limited write access to Tableau users for creating temporary tables and storing more data locally in Tableau Server.

You can also limit what users see what data by setting user filters in workbooks and data sources to better control what data users see in a published view based on their Tableau Server login account. By combining these techniques, you can publish a single view or dashboard in a way that provides secure personalized data and analysis to a wide range of users on Tableau Server.

Database Authentication

If data is extracted using Tableau's fast Data Engine, then database security permissions will not be propagated to the end users. When automatically refreshing or incrementing extracts, Tableau Server will use a single set of saved credentials to generate extracts for each data source (either as the "run as user" or the credentials embedded in the workbook). It will enforce that user's security privileges to the database.

Views published with live data connections on Tableau Server are dynamic in that they query the database each time to retrieve current data. Whenever a user opens a view and the data source is a database that requires a login (as opposed to something like an Excel workbook or a text file), then Tableau Server needs to know the database username and password to connect and 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 a clear distinction between Tableau Server's login techniques, which are used to gain access to Tableau Server itself, and the database login that may be required for the data source. The table below summarizes the options when creating and publishing views to Tableau Server:

Authentication Type	Tableau Server Response	Tableau Server leverages user-based data security built into database?
Username & password prompt	Tableau prompts each viewer to enter their own database credentials	Yes, the individual user identity is known to the database
Embedded password	The author specifies the database credentials when publishing the view. Viewers are not prompted for any credentials	No, all users share the same database login, that of the author
Viewer/publisher credentials	The user's domain user name and password are used to authenticate through SSO via Kerberos or SAML	Yes, the individual user identity is known to the database
Windows-integrated security (NT authentication)	"Run as user" of Tableau Server	No, all users share the same database login
Linux integrated security (ad/kerberos delegation)	"Run as user" of Tableau Server	Yes, the individual user identity is known to the database

Windows Authentication

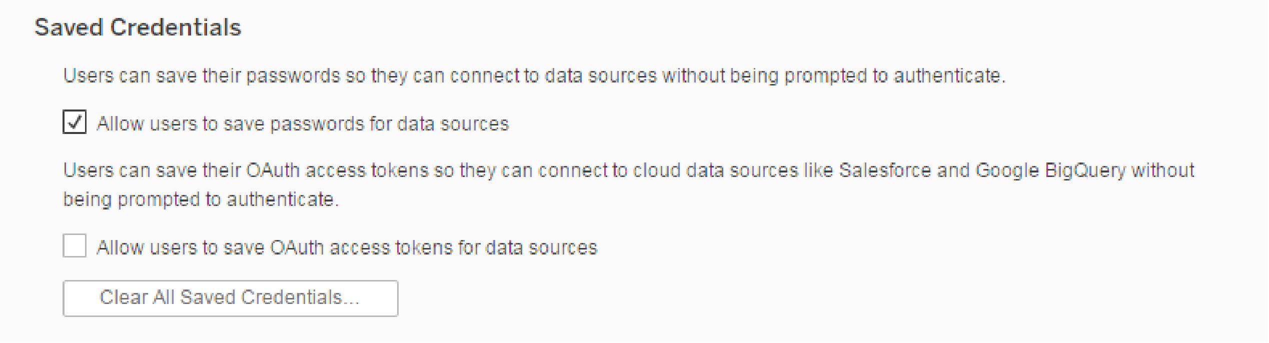
Tableau Server uses the "run as user" credentials to connect to the database with Windows. All users of Tableau Server will share this profile's connection information for the database. This does not use the credentials of the publisher or the credentials of the user logged in to Tableau Server. This option requires the database to leverage Windows-integrated security. This is very common for SQL Server or SQL Server Analysis Services implementations. Upon installation, the default "run as user" for Tableau Server is the Network Authority user. By definition, this network authority account does not have rights to connect to databases. To use an account that will accommodate NT authentication with data sources, specify a username and password including the domain name.

Linux Authentication

Tableau Server on Linux uses the "run as user" credentials as well, however, this is done in a slightly different manner. On Linux, you must provide a key tab file for the user that you want to use as the "run as user." This means that you'll need to establish a different "run as user" for a given task. For example, to connect to a given database, the data source must use a data source "run as principal" or "run as user." The data source "run as users" must be domain users, not just local users.

User Name and Password (Not Embedded)

Each user of Tableau Server will be prompted to log in to the database with his or her database-specific username and password. If you already have preexisting database security set up, this is a good option to leverage that security through Tableau Server. If you turn on the "saved credentials" option on the Tableau Server Settings page, then a Tableau Server user only needs to enter credentials one time per data source. Tableau Server then stores the user's data-source credentials and re-uses them just for that user's next connection to the same data source. Note that these credentials are typically separate from those used to log onto Tableau Server. Tableau always encrypts all passwords that are stored in the Tableau Server Repository. Database passwords are encrypted with a strong key. Fresh asset keys should be generated for each deployment using the `tabadmin assetkeys` command.



Saved Credentials

Users can save their passwords so they can connect to data sources without being prompted to authenticate.

☑ Allow users to save passwords for data sources

Users can save their OAuth access tokens so they can connect to cloud data sources like Salesforce and Google BigQuery without being prompted to authenticate.

□ Allow users to save OAuth access tokens for data sources

Figure 3. Saved credentials settings on Tableau Server Settings page.

Embedded Credentials (Not for Use with Windows Authentication)

When you enable embedded credentials, Tableau Server can remember the username and password of the original author of each workbook. At publish time, the author simply enters a set of credentials for the database—his or her username and password—and selects "embed credentials." All users of Tableau Server will then use these same connection credentials when retrieving data from that data source. Tableau Server will use the same encryption mechanism described earlier to secure the embedded credentials in the Repository. One caveat to consider when choosing this method is that passwords can expire, thus preventing users from accessing the data.

Additional Database-Specific Options

Impersonation

For Microsoft SQL Server data sources, Tableau Server supports impersonation of users when running queries. This allows Tableau to leverage security that you may already have implemented in Microsoft SQL Server. Tableau will either connect to the database using the "run as user" option 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.

Kerberos Delegation

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook to execute a query instead of the author. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).

- Your data source has row-level security, where different users have access to different cells.

For this to work, the database must support Kerberos delegation. Tableau Server requires constrained delegation, with the "run as user" account specifically granted delegation rights to the target database Service Principal Names (SPNs). Delegation is not enabled by default in Active Directory.

Row-Level Security and Impersonation with Initial SQL

When connecting to some databases, you can specify an initial SQL command to run when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. This initial SQL is different from a custom SQL connection, which defines a relation (table) to issue queries against.

You can use this command to:

- Set up temporary tables to use during the session
- Set up a custom data environment

You can pass parameters to your data source in an initial SQL statement.

There are several reasons why this is useful: You can configure impersonation using the **TableauServerUser** or **TableauServerUserFull** parameters. If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they are authorized to see.

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. Additionally, using a Query Band can increase performance. In order for query banding to work in Tableau Server, you must configure it appropriately.

User Filters

User filters are Tableau Server's approach to row level security. Tableau uses dynamic data filtering based on the username, group membership, and other attributes of the logged-in 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. User filters can be used with all data sources, including data extracts.

Published data sources can be built with calculated fields to control a variety of dimensions or measures based on the username or group membership of logged in users. This field is then added as a data source filter before publishing. By denying download capability, this makes the user filter immutable for both Tableau Desktop and Tableau Server users connecting to the data source for ad hoc analysis.

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. This allows a single workbook published to Tableau Server to securely deliver the appropriate 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, all based on their credentials.

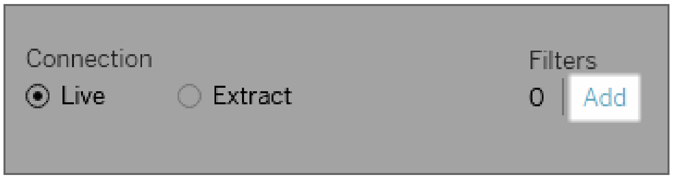
The benefit of this approach is that no additional maintenance is needed for the views when new users and data are added to the system. The filtering rules are built into the views and the database dynamically provides the keys for those rules to process.

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 a calculated user filters, but does not dynamically adapt to new users and data elements. Therefore, additional maintenance to the views is required.

Data Source Filters

Tableau Server supports creating filters directly on a data source, thereby reducing the amount of data returned from the data source. For example, your database may include data from the last 5-10 years, however, you only want your users to have access to the last three years of data. Adding a data source filter makes it easy to only show that timeframe.

If you create an extract from a data source that already has data source filters in place, those filters are automatically recommended as extract filters, and will appear in the extract dialog. Those recommended filters are not required to be part of the extract filter list, and can be removed independently from the existing set of data source filters.



Connection

● Live ○ Extract

Filters

0 Add

Figure 4. Adding filters to Tableau data sources from Tableau Desktop.

Data source filters can be useful for restricting the data users can see when you publish a workbook or data source. When you publish a data source to Tableau Server, the data source and any associated files or extracts are transported in their entirety to the server. As you publish a data source, you can define access permissions for downloading or modifying the data source, and you can also choose the users and groups that can remotely issue queries through Tableau Server against that data source. When users have query permission and no download permission, you can share a rich data model with calculated fields, aliases, groups, sets, and more—but limited to just querying.

Furthermore, users who query published data sources will never be able to see or modify any data source filters present on the underlying published data source, and all of the users' queries will be subject to that data source's filters. This is a great way to offer a restricted subset of your data, for example by filtering dimensions for specific users and groups, or by defining data source filters based on a fixed or relative date range. This is useful for data security but it also allows you to manage performance of the remote database, which Tableau Server will ultimately query on a user's behalf. For systems that rely heavily on partitions or indexing, data source filters can yield tremendous control over the performance of queries issued by Tableau.

Extract Security

When data extracts are used, Tableau Server is responsible for storing and processing data used in views and workbooks. The data gets stored on the file system as a Tableau data extract (TDE) in an encoded, compressed, binary format. The metadata that describes the extracts is stored in plain text. This means the data is not human-readable; however, we can discern some descriptions of the data such as data types, field names, and so on. To protect these files, Tableau Server stores them in the "Program Data" directory with access controls restricted to the Tableau Server "run as user" and local administrators of the machine. The extract data files themselves are not encrypted on disk.

Just like other databases that Tableau connects to, Data Engine extracts cannot be queried directly from the Tableau Server user interface. Although users can perform drag-and-drop analysis, users cannot compose SQL, MDX, or any other syntax to interact directly with the Data Engine database. This helps prevent unauthorized access, SQL injection, and other malicious attacks on extracts.

It is possible to integrate with third-party and OS solutions for disk-level encryption (e.g. BitLocker) or file- and/or directory-level encryption (e.g. Encrypting File System or EFS) to further enhance the security of the data extract files. But these solutions generally target all data on the disk, so encryption will not be limited to Tableau Server data files. In addition, there may be a performance impact when enabling these solutions.

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 does not allow external connections. This means access to the information stored in the Repository is by default restricted to just Tableau Server components. However, customers who'd like direct access to this information can configure the Repository using the tabadmin dbpass command to allow external connections. External connections are restricted to read-only views of the data to prevent malicious use and accidental changes to Tableau Server's content or configuration. You can also configure the Repository to only allow SSL connections using the Tableau Server configuration utility.

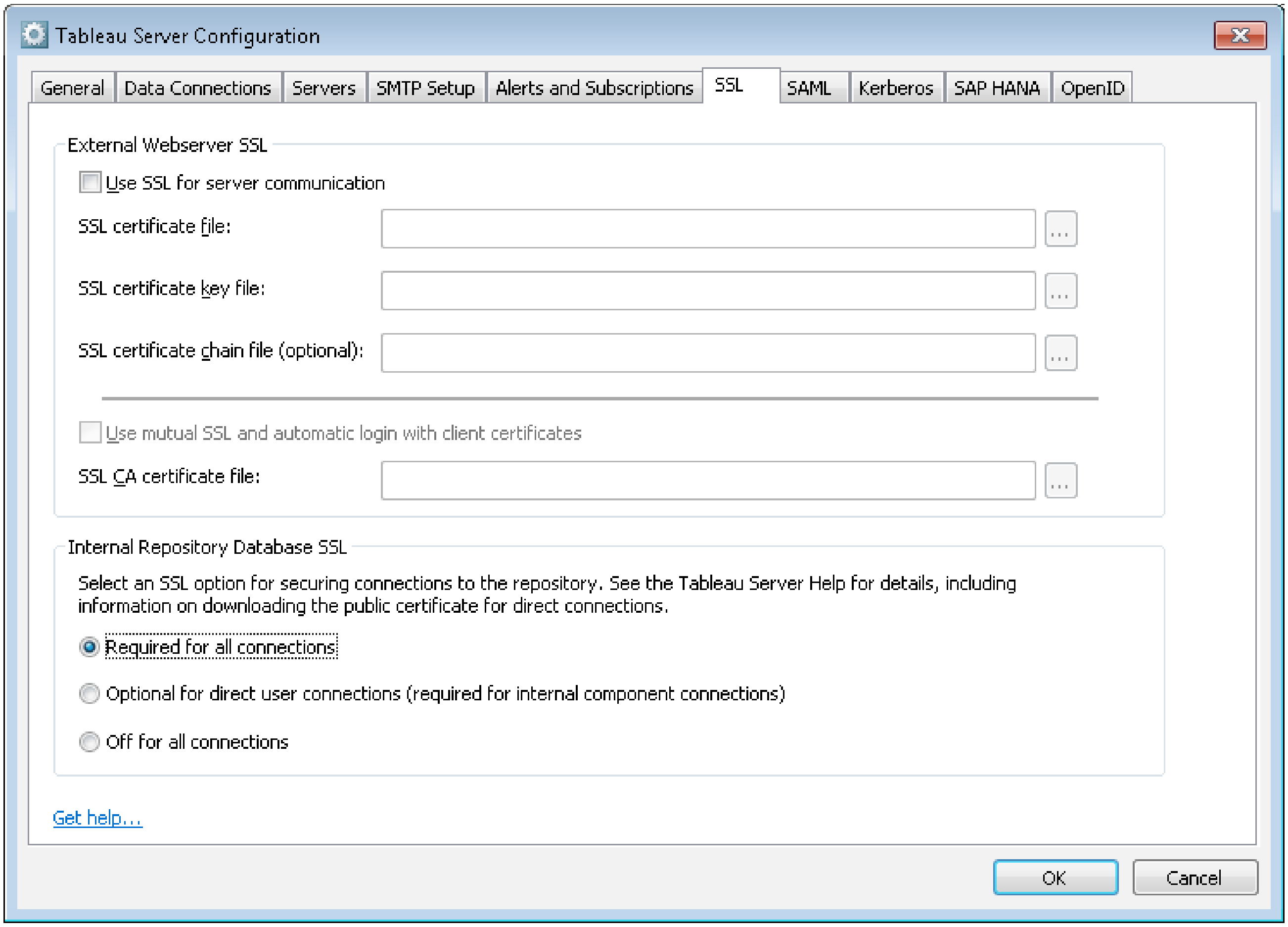


Tableau Server Configuration

External Webserver SSL

□ Use SSL for server communication

SSL certificate file:

SSL certificate key file:

SSL certificate chain file (optional):

□ Use mutual SSL and automatic login with client certificates

SSL CA certificate file:

Internal Repository Database SSL

Select an SSL option for securing connections to the repository. See the Tableau Server Help for details, including information on downloading the public certificate for direct connections.

● Required for all connections

○ Optional for direct user connections (required for internal component connections)

○ Off for all connections

Figure 5. Configuring internal Repository database SSL

4 Network – Transmission Security

Administrators often use network security devices to protect access to Tableau Server deployed on-premises from untrusted networks and the Internet. However, even in these cases, credentials still need to be securely transmitted across the network. When access to Tableau Server is not restricted, transmission security becomes even more critical to protecting sensitive data and credentials, and preventing malicious use of Tableau Server. Regardless of your situation, Tableau Server has robust transmission security capabilities.

There are three main network interfaces to 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 pays special attention to the storage and transmissions of passwords at all layers and interfaces.

Client to Tableau Server

In this case, “client” means a web browser, Tableau Desktop, tabcmd, or REST API applications. 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/TLS) with customer supplied security certificates. When Tableau Server is configured for HTTPS, all content and communications between clients is encrypted and uses the HTTPS protocol. SSL/TLS should be enabled for all deployments where security is a concern.

When Tableau Server is configured for HTTPS, the browser and HTTPS library on the server negotiate a common encryption level. Tableau uses OpenSSL as the server-side HTTPS library and it is preconfigured to use currently accepted standards. Each web browser accessing Tableau Server via SSL uses the standard HTTPS 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.

Tableau Desktop communicates with Tableau Server using either HTTP or HTTPS. Protecting the transmission of passwords securely requires HTTPS to be enabled.

Communication between Tableau Server and the Database

Tableau Server makes dynamic connections to databases to process result sets and 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, and this type of configuration is transparent to Tableau.

Communication between Tableau Server Components

This section only applies to distributed deployments of Tableau Server. There are two aspects to communication between Tableau Server components: trust and transmission. Each server node in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from the other nodes in 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. It is recommended that Tableau Server be firewalled off from unsecure servers.

5 Other Considerations

Due to the outward facing nature of extranets, Tableau Server has many built-in safeguards to maintain integrity in an exposed environment. For example, we require that all client communications go through a single port. In addition, we provide support for configuring forward and reverse proxies so that communications between your network and the internet can be mediated using proxy servers.

Tableau has invested in an internal security team that actively tests for vulnerabilities and quickly addresses new threats with monthly updates. To get the latest information, please visit our Security page and review our whitepaper on secure development. Finally, we strongly recommend that you also review the Security Hardening Checklist which provides additional suggestions for securing your deployment of Tableau Server.

Summary

Tableau Server provides a comprehensive set of security capabilities to suit your deployment needs. Tableau has proven successful public-facing deployments on countless customers' sites and internal deployments on secure networks. Tableau uses modern industry standards as a baseline and is responsive to future threats and issues. From row-level security, to secure websites, and every security detail in between, Tableau has considerations to your security questions built directly into our platform.

About Tableau

Tableau helps people transform data into actionable insights that make an impact. Easily connect to data stored anywhere, in any format. Quickly perform ad hoc analyses that reveal hidden opportunities. Drag and drop to create interactive dashboards with advanced visual analytics. Then share across your organization and empower teammates to explore their perspective on data. From global enterprises to early-stage startups and small businesses, people everywhere use Tableau's analytics platform to see and understand their data.

Resources

Tableau Server Hardening guide

Tableau Server Administrator Guide

Tableau Server High Availability: Delivering mission-critical analytics at scale

Tableau Server Scalability - A Technical Deployment Guide for Server Administrators