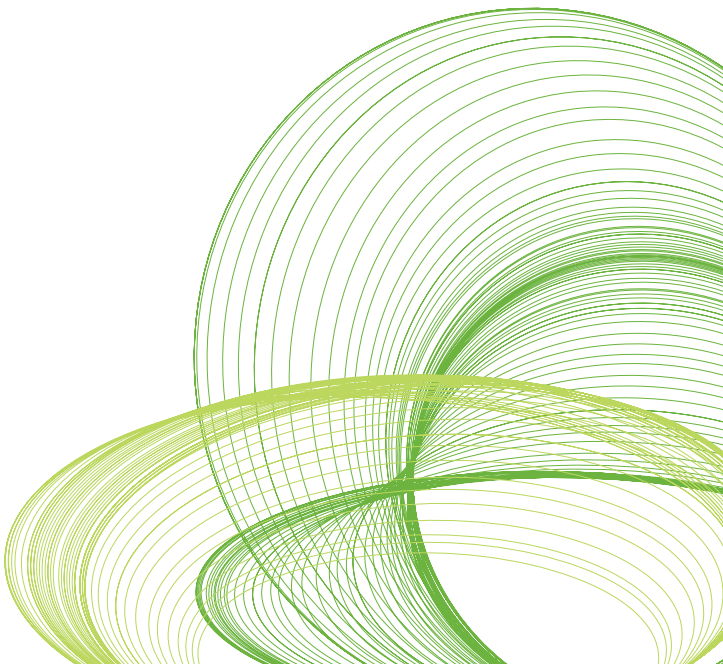
QlikView

QLIKVIEW SECURITY OVERVIEW

A QlikView Technology White Paper

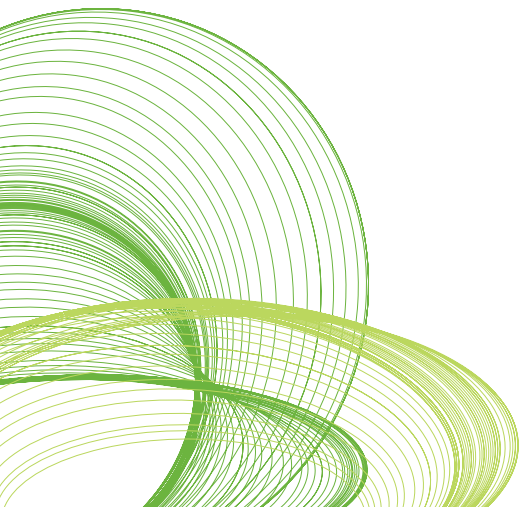


Published: February, 2011

qlikview.com

QlikView

Table of Contents



SECURITY COMPRISES THREE MAIN COMPONENTS

- **Authentication.** Who you are and how do you prove it? QlikView uses standard authentication protocols such as Integrated Windows Authentication, HTTP headers and ticketing to authenticate every user requesting access to data.
- **Document-level Authorization.** Do you have access to the document or not? QlikView utilizes server-side capabilities such as Windows NTFS or QlikView's own Document Metadata Services (DMS) to determine access privileges at the file level.
- **Data-level Authorization.** Are you allowed to see all of the data or just part of it? QlikView implements row and field level data security using a combination of document-level capabilities (Section Access) and server-side data reduction capabilities using QlikView Publisher.

Overview

Security in any context is critically important. It becomes especially important when applied to the enterprise software solutions that organizations rely upon to make decisions based on sensitive information. Ensuring that the right people have access to the right information at the right time and no one else is a critical part of being able to support decision making. Company performance, employee payroll, sales data, personally identifiable information and forecast information are just some examples of company information that are commonly represented within QlikView applications.

Security in QlikView is multi-faceted and powerful but is also easy to implement and familiar to IT professionals. This paper will discuss the following topics as applied to QlikView:

- **Authentication.** Who are you? How did you prove it?
- **Authorization.** What are you allowed to see? What are you allowed to do?
- **Implementation.** How do you implement security in QlikView?

Out of scope of this document:

- Firewalls and network security
- Encryption of traffic, credentials, or data

Common Security Challenges

Organizations increasingly face challenges related to implementing a secure environment so that unauthorized access to data is prohibited. The most common security challenges are:

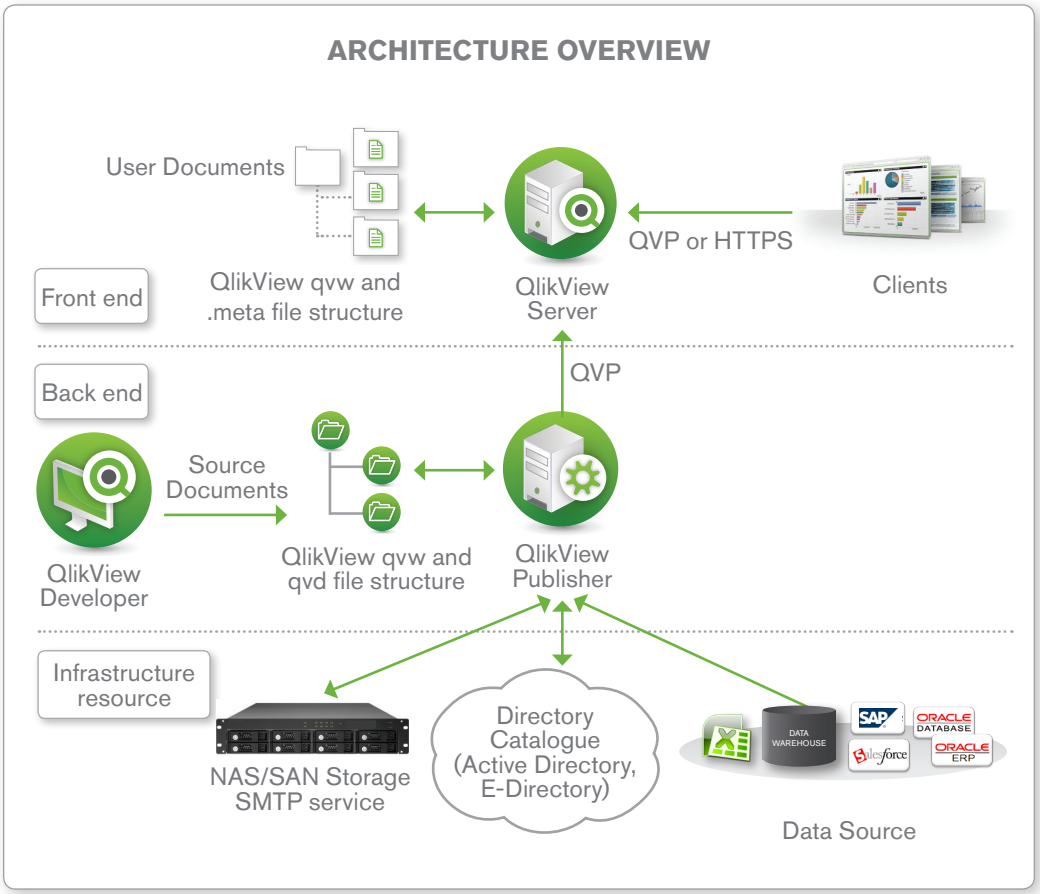
- **Trust.** With any enterprise software solution, and especially those that deal with analyzing, presenting and distributing sensitive information, IT professionals need to know precisely what safeguards are in place and how to mitigate the risk associated with providing access to sensitive information. They also need firm evidence that the vendor's solution can comply with their corporate IT security standards.
- **Complexity.** To obtain satisfactory security in a solution, the vendor product needs the flexibility to be able to cope with the most common architectures used by the customer. Not understanding the approach that any particular vendor takes can cause confusion that can lead to an unsatisfactory security level.

QlikView Architecture

In order to frame the discussion on security contained in this paper, it's important to first understand the roles of the various products that comprise a QlikView deployment and to understand the how a tiered approach to application and data security forms a best practices basis for deploying a secure QlikView environment.

Figure 1 depicts a simplified view of a standard QlikView deployment containing the location of the various QlikView products as well as both data and application locations.

Figure 1: Architecture Overview



BACK END (including Infrastructure resources):

This is where QlikView source documents, created using the QlikView Developer, reside. These source files contain either a) scripts within QVW files to extract data from the various data sources (e.g. data warehouses, Excel files, SAP, Salesforce.com) or b) the actual binary data extracts themselves within QVD files. The main QlikView product component that resides on the Back End is the QlikView Publisher: the Publisher is responsible for data loads and distribution. Within the Back End, the Windows file system is always in charge of authorization (i.e. QlikView is not responsible for access privileges).
The Back End depicted here is suitable for both development, testing and deployment environments.

FRONT END

The Front End is where end users interact with the documents and data that they are authorized to see via the QlikView Server. It contains the QlikView user documents that have been created via the QlikView Publisher on the back end. The file types seen on the Front End are QVW, .meta and .shared documents. All communication between the client and server occurs here and is handled either via HTTPS (in the case of the AJAX client) or via the QlikView proprietary QVP protocol (in the case of the plugin or Windows client). Within the Front End, the QVS is responsible for client security.

From a security standpoint, it's important to understand that the Front End does not have any open ports to the Back End. It does not send any queries to data sources on the back end, nor do any of the user documents (QVW's) contain any connection strings to data sources located on the back end. End users can only access QlikView documents that exist on the Front End, and never in the Back End.

This paper will now examine the means by which users and services are authenticated within a typical QlikView deployment and the methods by which users and groups are authorized to access QlikView applications and data.

At its most fundamental level, security consists of correctly indentifying who the user is (Authentication), restricting their access to resources (Authorization) based on their authenticated identity, and the ability to audit that the security system has given the right people access to the right information.

Authentication (Who are you? How do you prove it?)

All computer systems require some form of authentication before the user can begin interacting with them. There are times that, after a user has authenticated themselves to an individual system, they need a resource on a second system. In many medium and large organizations, this is accomplished via a sophisticated Single Sign-On (SSO) mechanism.

Although QlikView can be configured to allow anonymous access, the majority of implementations require that users be authenticated. In these environments, QlikView always requires that the user is authenticated when establishing a session via the QlikView Server (either through a browser or when downloading and opening the document via the desktop client).

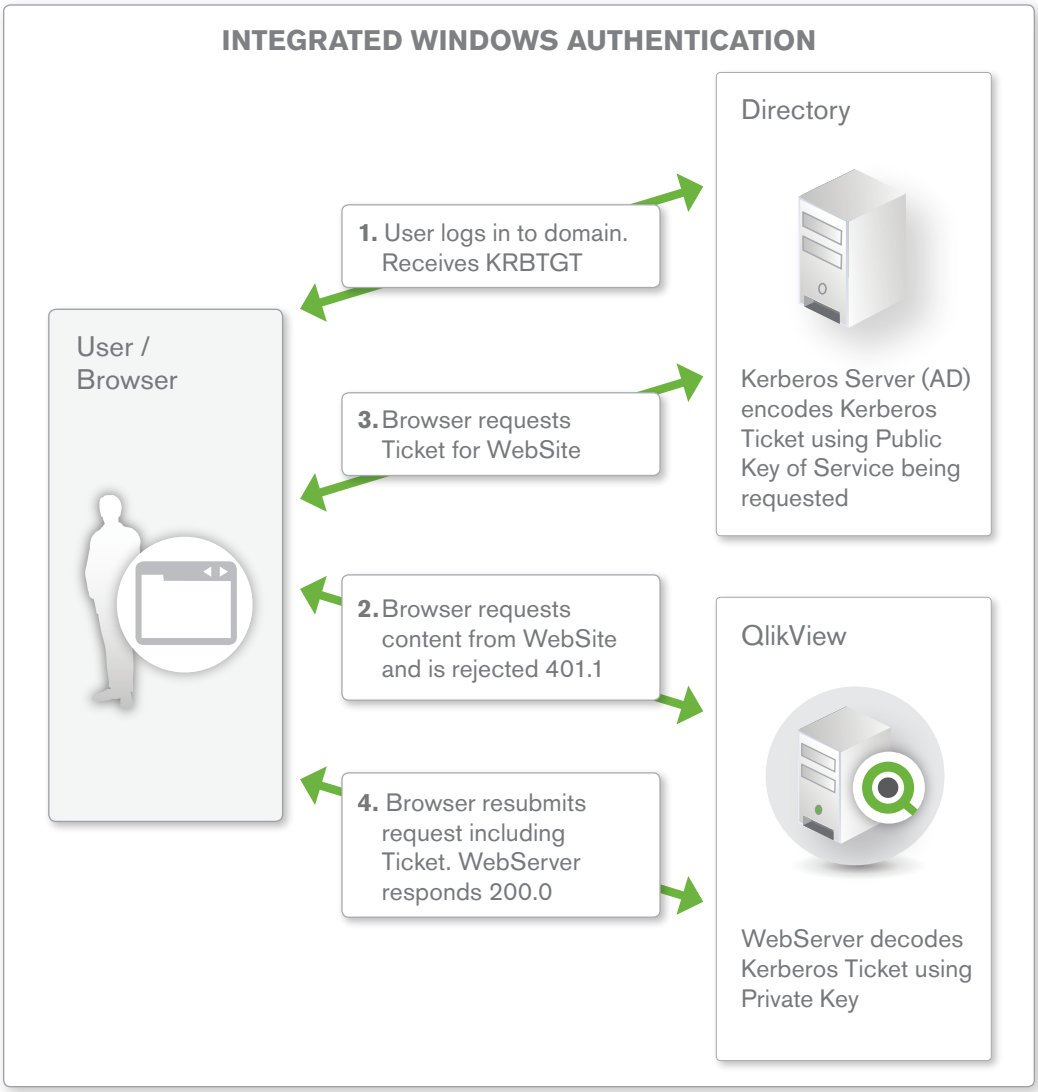
In the QlikView context, the authentication of a user is almost always done against an external entity that is then used to pass the externally authenticated user identity to the QlikView Server. In these scenarios, QlikView relies on the authentication to be performed prior to accessing QlikView, and some token of iIdentity is transmitted to, and trusted by, QlikView.

AUTHENTICATION WHEN USING QLIKVIEW SERVER IN A WINDOWS USER ENVIRONMENT

Authentication to a QlikView Server when in an environment based on Windows users (e.g. incorporating Active Directory) is straightforward and is a very common implementation scenario. The process is as follows (see Figure 2):

1. The User's credentials are validated when they sign into their client computer's Windows operating system.
2. Later when they want to establish a session with a QlikView Server (QVS) (e.g. via a browser on their desktop), the QVS can utilize the built-in Integrated Windows Authentication (IWA).
3. The logged-in user's identity is communicated to QlikView Server using either the Kerberos or NTLM security solution. This solution will provide single sign-on capabilities right out of the box. In case the authentication exchange fails to identify the user, the browser will prompt the user for a Windows user account name and password.

Figure 2: Using QlikView with Integrated Windows Authentication



The authentication process differs based on whether the environment is:

- On a local area network: Integrated Windows Authentication is most common, and most suitable for recognized Windows users on a LAN. The act of authentication is performed when logging into the workstation, and this identity is leveraged by QlikView.
- In a multi-domain environment: The internal company network IWA should be avoided in architectures where a multi-domain environment exists with no trust relationship between the domain of the workstation and the domain of the server, or when used across a reverse proxy. In such an environment the QlikView deployment should be configured to use either an existing external SSO service or a QlikView custom ticket exchange to expose an authenticated identity to QlikView.

AUTHENTICATION WITH A QLIKVIEW SERVER USING AN EXISTING SINGLE SIGN-ON SOFTWARE PACKAGE

In environments where SSO infrastructure already exists (e.g. CA SiteMinder®, IBM WebSeal®, or Oracle Oblix®), QlikView can utilize the HTTP header injection method of single sign-on provided by these software packages. Once again this provides single sign-on right out of the box. These software packages can be configured as follows:

- Repeat users get access: In short what happens is that these software packages can be configured to protect a resource. When a user requests access to QlikView, the SSO package will grant access if they have previously signed into the SSO's authentication page.
- New users log in: If the user does not have an existing session with the SSO package they will be redirected to the SSO package's login page. After logging in they will be redirected to the original URL they requested.

In both cases, if the user has properly authenticated to the SSO software, the user's username gets injected into an HTTP header and the value in that header is what the QlikView server accepts as the user's authenticated identity.

Use caution with the HTTP header method of authentication. Note, that unless you have SSO software in place the HTTP header method of authenticating to a QlikView Server should not be used. HTTP Headers can be easily spoofed. All of the SSO software packages named above offer protection against this type of spoofing attack if they are the only path that users can use to access the content.

QlikView does not recommend or endorse any specific tool or product for providing identity in an HTTP header. The approach is highly suited to extranet deployments wherein the users may not exist in the internal Active Directory. The act of authentication is performed by the reverse proxy or ISAPI filter that intercepts the end user's attempt to interact with QlikView content.

AUTHENTICATION USING NEITHER IWA NOR SSO SOFTWARE

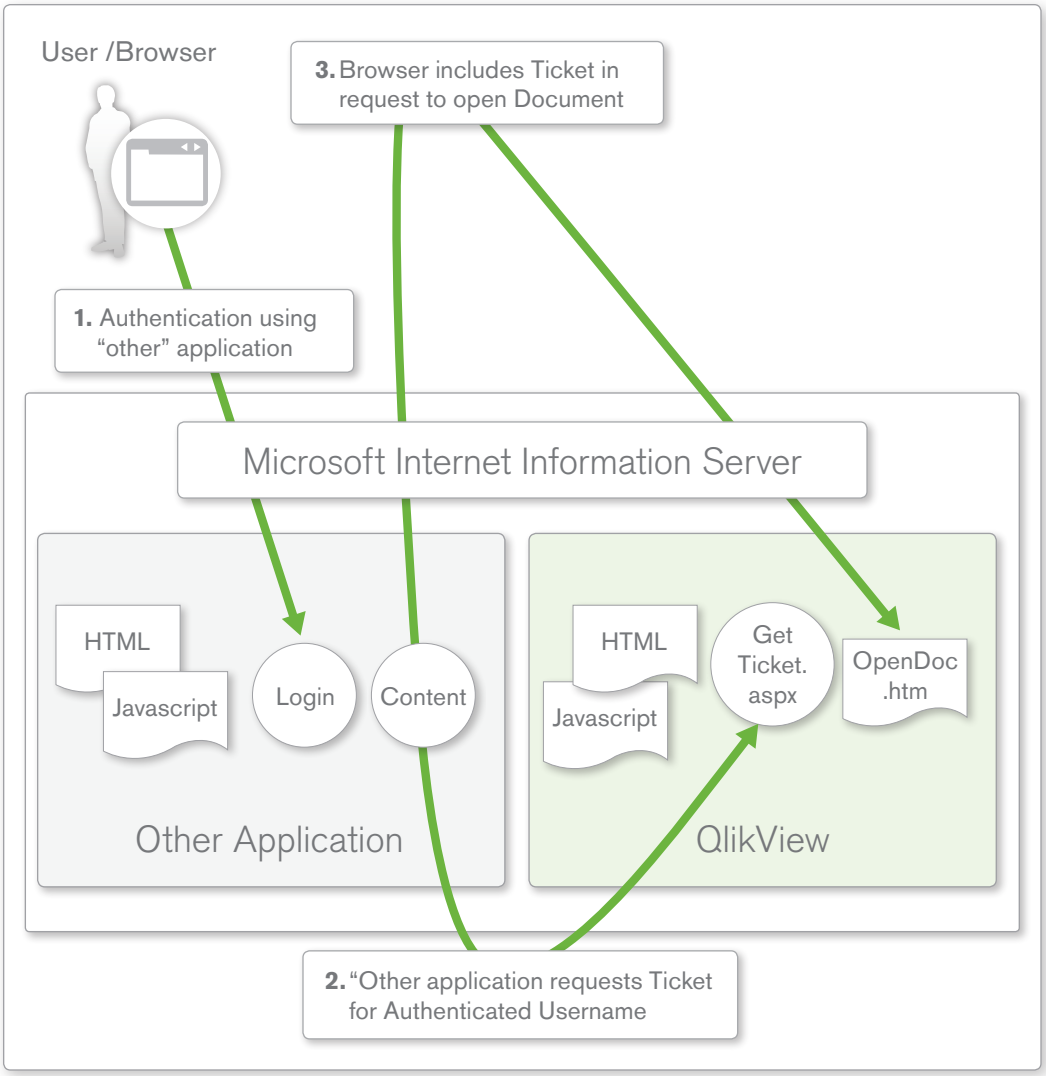
QlikView offers a third method of single sign-on when neither of the above methods is suitable. The third method is called Custom Ticket Exchange (CTE).

CTE relies on the user having authenticated previously to some other system.
It works like this:

1. The third-party system is granted the privilege and responsibility to request an authentication token (called a 'ticket' in QlikView) from the QVS on behalf of that system's authenticated user. It is that third-party system's responsibility to only request a ticket for a user who has properly been authenticated (e.g. QVS has no knowledge of the user's authentication status). This system then passes the authentication token to the user who then utilizes it in their request to open a session with the QVS. The QVS validates that the ticket is valid and then opens a session for the authenticated user.
2. This system then passes the authentication token to the user, who then utilizes it in their request to open a session with QVS.
3. QVS validates that the ticket is valid and then opens a session for the authenticated user.

Ticketed authentication is most applicable to embedding QlikView content in third-party applications and portals, and rarely used for providing general access to QlikView. Typically a small amount of custom development is needed to implement the request and passing of this ticket for the Custom Ticket Exchange method to work.

Figure 3: Authentication using ticketing

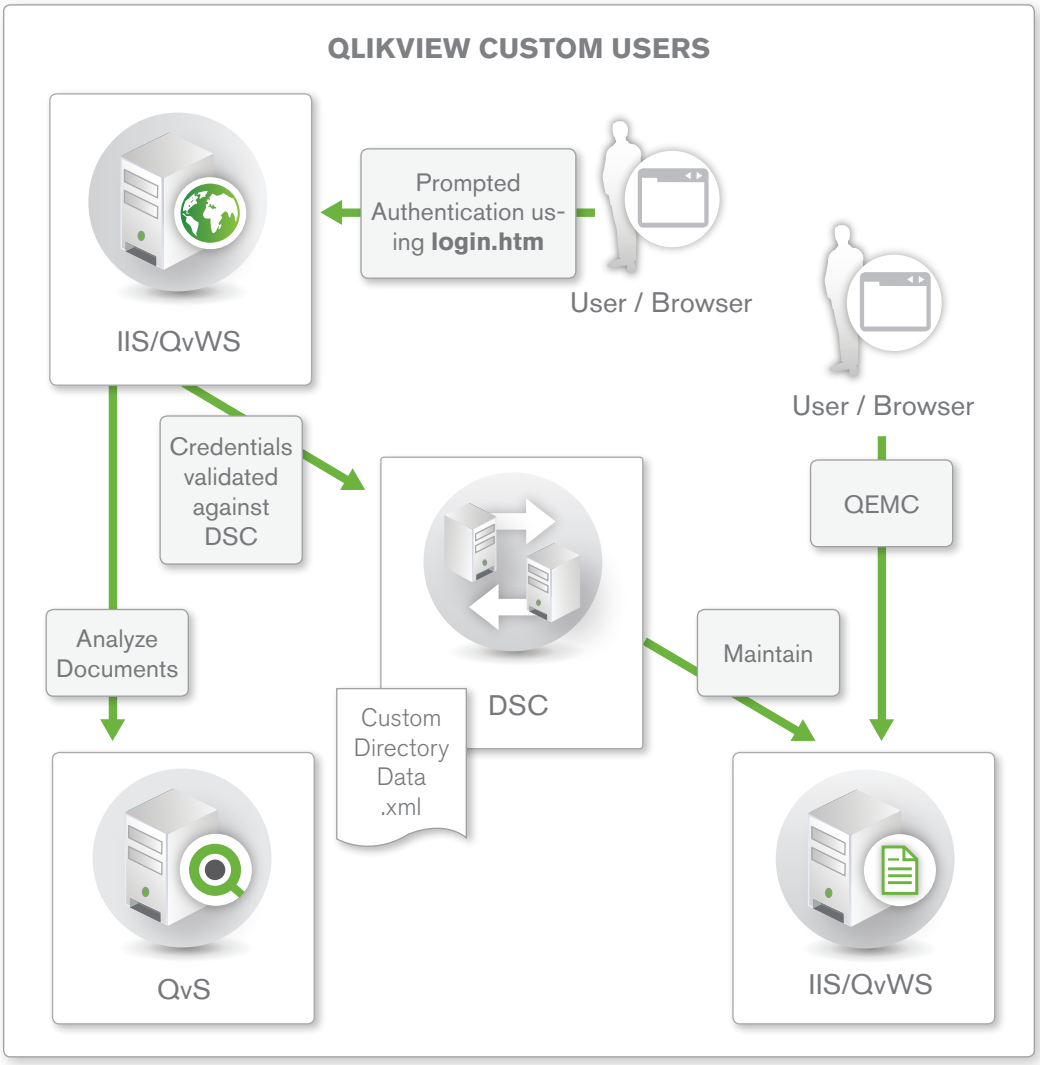


QLIKVIEW SERVER AUTHENTICATION USING "CUSTOM USERS"

The three methods described above all use a single sign-on principal where the userid and password are stored externally to the QVS and an external entity is responsible for the authentication. Less common, although available, is the ability to store the user credentials inside of the QlikView Server environment itself by using the QlikView Publisher Custom Users functionality. In this case the users and their passwords are defined and stored within the QlikView environment and the web tier of the QlikView deployment is responsible for forms authentication. This solution is suitable for smaller, standalone QVS deployments, and should not be used in environments where the user definitions need to be available to multiple systems. In environments where definitions should be available on multiple systems it is highly recommended to use one of the three single sign-on solutions described.

Each coexistent form of authentication may require a distinct WebServer instance. Several WebServers can forward User Requests to the same QVS instance(s).

Figure 4: QlikView Custom Users



A NOTE ON ENCRYPTION

While a broad treatment of the topic of encryption is out of scope for this document, it's worth mentioning here that encryption is a fundamental aspect of security in a distributed QlikView environment. QlikView automatically encrypts all data between the QlikView Server & QlikView Client (i.e. desktop thick client, IE Plugin or iPhone/iPad) with 128-bit encryption. Encryption between QlikView Server and the AJAX ZFC is possible using standard public/private key cryptography (employing HTTPS). In order to guarantee secure communication, an administrator must install a certificate (self-signed or otherwise) on the QlikView Web Server. QlikView recommend that HTTPS encryption is used if sensitive information or authentication tokens are transferred over HTTP.

Authorization (What are you allowed to see? What are you allowed to do?)

Once a user has been authenticated (i.e. the system knows who they are), the first step in assigning their security privileges has been completed. The second step is to understand what authority or rights they have to applications, data or both. This step is called Authorization. At a fundamental level, an administrator will populate an Access Control List (ACL) with a list of users and/or groups and what they should have access to. When the time comes for a user to request access, the system looks up the user's authenticated identity in the ACL and verifies if the administrator has granted the user enough privileges to do that.

Direct access to the QlikView Document using QlikView Desktop is always governed by Windows NTFS File Security. Access to the web-based QlikView Enterprise Management Console is restricted to Windows Users who are a member of a particular local Windows Group.

DOCUMENT-LEVEL AUTHORIZATION

Once a user has been authenticated, the QVS typically handles authorization on its own. The QVS offers the choice between storing the ACL information as Windows NTFS privileges (applicable only when the user authenticated using a Windows user identity) or by storing the ACL information in QlikView's own internal repository (called DMS or Document Metadata Service). The choice of NTFS or DMS affects access to all documents on the QVS.

NTFS VS DMS

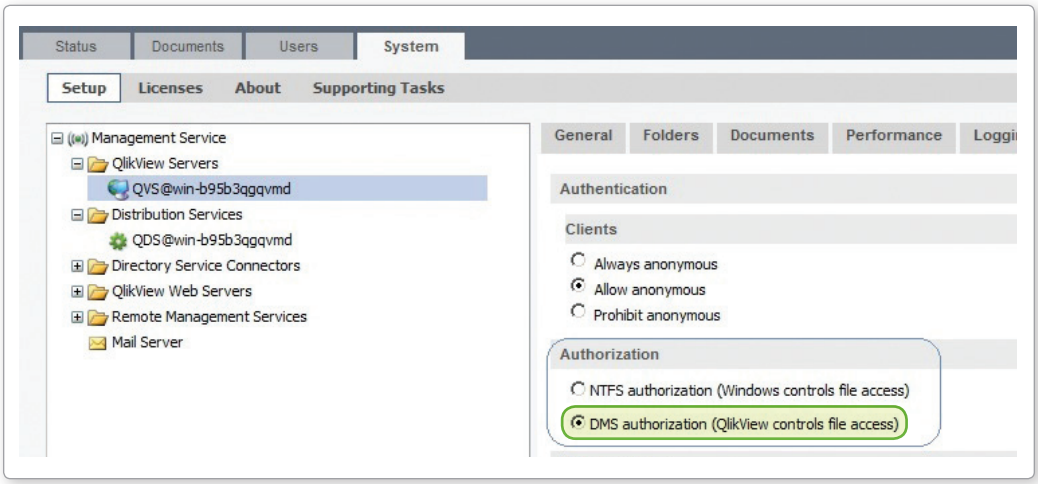
The QVS can use the Windows file system's own NTFS privileges to store authorization information. When in NTFS authorization mode, the QVS controls access to a given QlikView document by determining if the authenticated user has NTFS privileges to the underlying QlikView Document file (.QVW file). This is based on the operating system privileges and Windows NTFS is used for the ACL. The privileges of the authenticated user are configured by a server admin using standard Windows Explorer functionality via directory properties options.

As an alternative to Windows NTFS, QlikView can use its own ACL called the Document Metadata Service (DMS). Unlike NTFS, this allows non-windows users and groups to be authorized to access applications and data. DMS integrates fully with the existing Directory Service Provider (e.g. Active Directory, other LDAP) where Group Membership has been recorded, this is a mechanism by which the QVS can re-use existing enterprise account and group structures. The permitted Users or Groups are recorded in a meta file that resides beside the QlikView Document, and is managed using the QlikView Enterprise Management Console.

NTFS is the default document authorisation model, and suitable when all users and groups are identified in Active Directory or locally on the QVS host. The NTFS permissions may be inherited from the directory that the QlikView documents are in, or may be assigned using QlikView Publisher distribution tasks.

DMS is required if the authenticated user identity is not a Windows User Account. The DMS permissions are explicitly assigned using the QlikView Enterprise Management Console, or may be assigned using QlikView Publisher distribution tasks.

Figure 5: Enabling DMS authorization in the QlikView Enterprise Management Console



DIRECTORY SERVICES

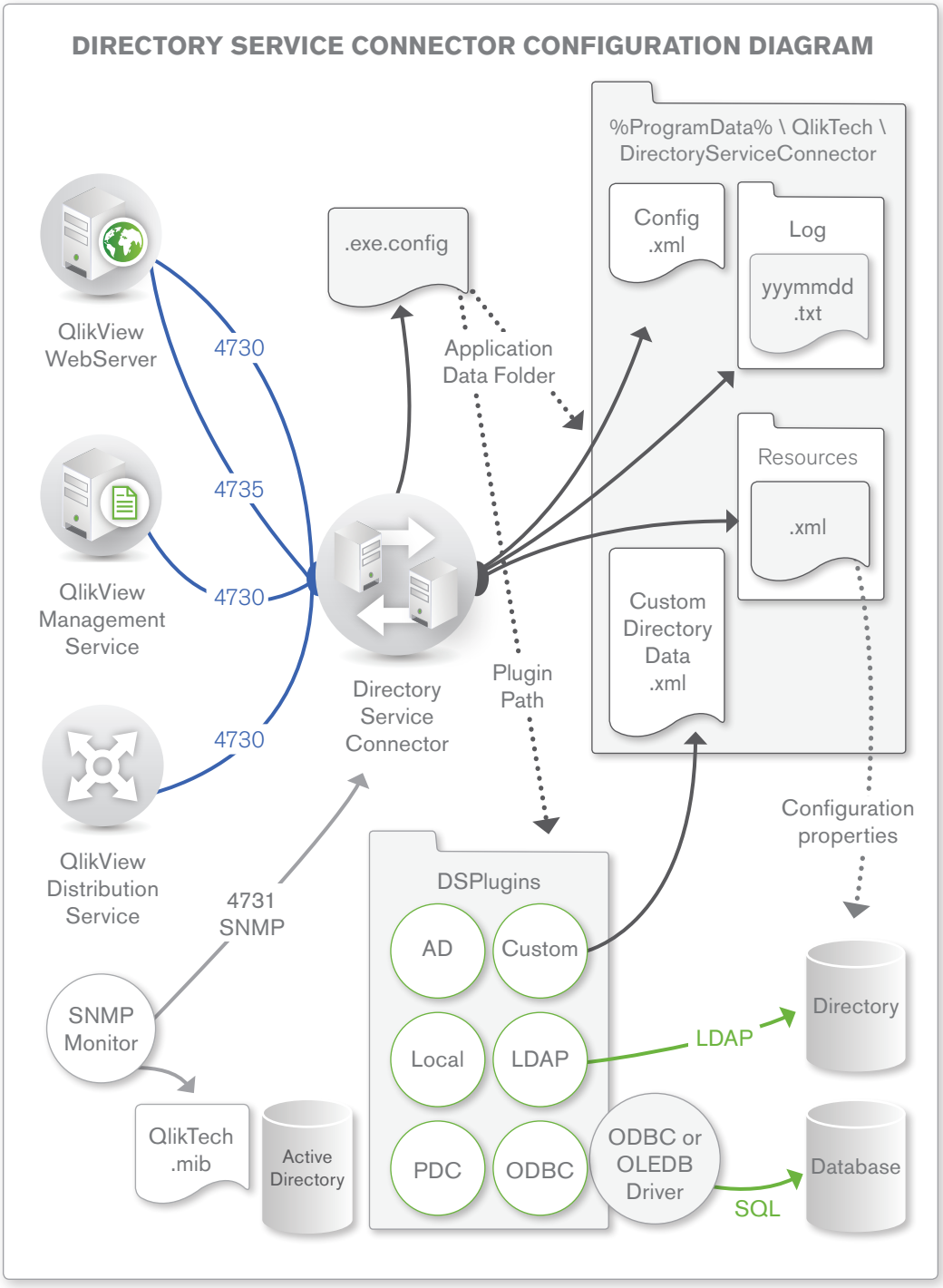
The QlikView 'Directory Service Connector' (DSC) component is used for interaction with whatever repository contains the Group Membership. QlikView provides several configurable 'Directory Service Providers' (DSP) and also an API for development of a custom DSP.

The following list describes the types of directory services QlikView supports.

- **Active Directory** — QlikView provides built-in support for users in the same domain as the QlikView Server. It also supports users in any domain trusted by the QlikView Domain.
- **Local Windows Users** — QlikView supports Local Windows users for environments where no Domain controller is available.
- **LDAP** — QlikView allows you to configure a connection to directories such as Novell, OpenLDAP, ADAM, SunOne LDAP.

- **ODBC** — QlikView Users and Groups may have been created in a database. This DSP is expected to be most relevant when QlikView has been embedded in an OEM product, and rarely used otherwise.
- **Custom (QlikView) Users** — QlikView allows administrators to create users in QlikView, side-stepping the need to set up users in Windows or Active Directory.
- **Custom Directory Service Connector** — QlikView provides the ability to write a custom connector to authorize against other user repositories (e.g. SalesForce.com, SAP)

Figure 6: Directory Service Connector configuration diagram



DATA-LEVEL AUTHORIZATION

It's very common to limit access to certain data within a QlikView application, beyond that which is possible using file-level restrictions. For example an organization might have a single application covering worldwide sales transaction data, however it may want to limit country managers to view data only pertaining to their own country. This data-level security can be further refined to both row-level and field (or column)-level data access. QlikView provides data-level security using Section Access within the QlikView document, or by using Reduction/Distribution performed by the QlikView Publisher or some combination of the two.

SECTION ACCESS (aka Dynamic Data Reduction)

'Section Access' is a QlikView technology that allows QlikView to control which users have access to specific data (or 'sections') in a QlikView application. Because the document contains all data, and the rules that restrict data access are enforced when an authenticated user interacts with the document it is also referred to as Dynamic Data Reduction. Section Access uses a security table to determine what data users can access, and applies security based on an association between users and data. User accounts can either be created internally within the application, or be integrated with user logins.

When using a Section Access security table to store usernames and passwords, these can be stored either within the QlikView document itself (behind a password-protected script) or in a separate database or file that resides in the QlikView Server.

Whenever the user's identity is supplied to QlikView (Integrated Windows Authentication, Header, Ticketing, Custom) this is exposed to the QlikView Document in a field called NTNAME. If the NTNAME is associated with data restriction rules in the document then there is no need to record or validate any further password within the QlikView Document.

User-Role-Data associations or restrictions can be re-used amongst many QlikView documents if these have been stored in a commonly accessible location such as database tables or using the User Management feature of the QlikView Enterprise Management Console.

At a basic level, an administrator needs to supply only 3 parameters for each user intended to use the application:

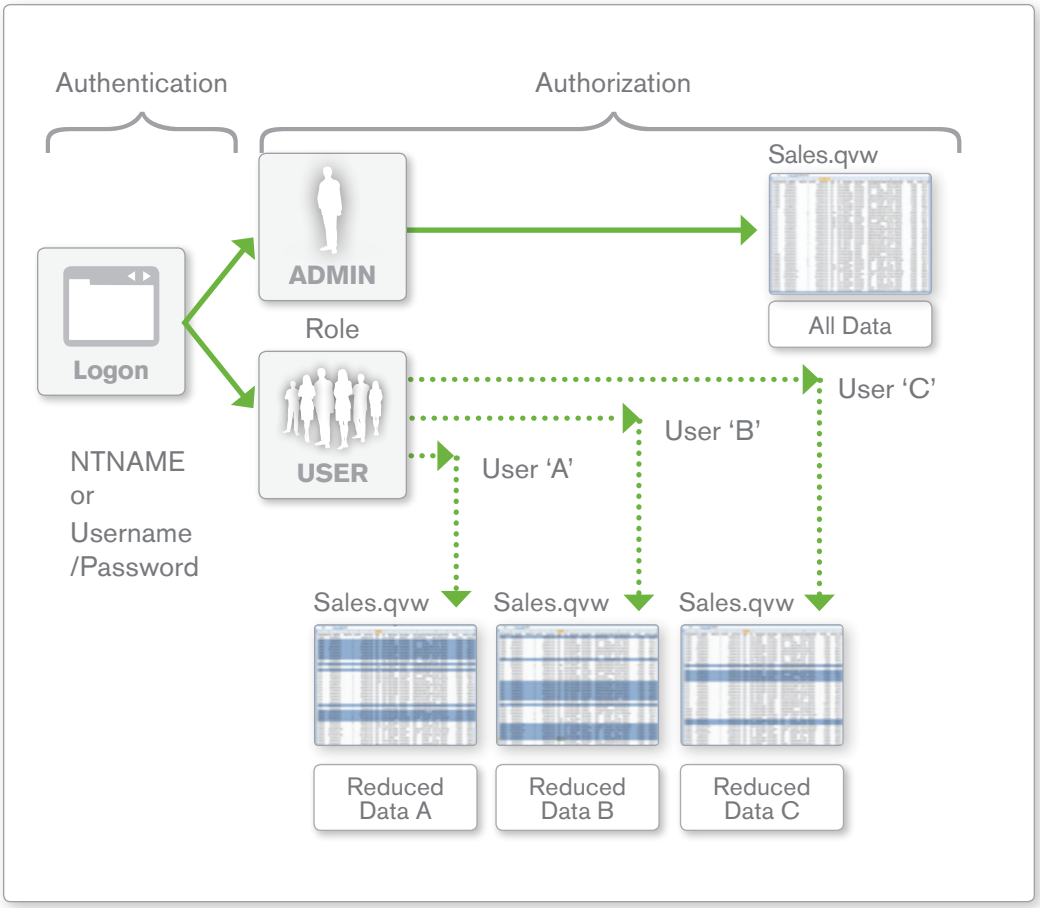
1. Role (User or Administrator) - that affects the Users capabilities when opening the QlikView Document file using QlikView Desktop.
2. Identity
   a. NTNAME, or
   b. Username and password
3. Reduction field (additional reduction fields optional)

Row Level Security. The reduction field is a field that determines which data a user can access. For example, if users should see only data associated with a specific department, the reduction field could be "Department", and users who have been granted access to the data associated with a specific department will be limited to access this data only.

Field Level Security: In addition, field (or column) level security can be applied using Section Access. Individual users or groups can be denied or permitted access to entire fields by invoking the 'OMIT' keyword when calling a Section Access script. The OMIT keyword is a reserved keyword in a QlikView script to determine which users will be granted access to specific field-level data.

Even if the document authorization (NTFS/DMS) permits a user to see that a document exists, the Section Access rules may exclude them from seeing any data, and so they will be unable to open the document.

Figure 7: shows an example of row-level security applied to a document using Section Access.



REDUCTION/DISTRIBUTION (AKA STATIC DATA REDUCTION)

For larger deployments and/or those needing centralized control of authorization capabilities, the QlikView Server/Publisher products are used. Often, a department or function has a 'master' application containing all relevant data covering all their analysis needs, and this master document needs to be separated ('reduced') according to the intended audience's needs and access privileges. The QlikView Publisher reloads the QlikView document with available data, refreshes the Section Access tables and splits the large QlikView document into many smaller documents based on values of a particular field.

This “Reduction and Distribution” allows for a file containing many data fields to be broken up by the contents of a field and distributed to authorized users or groups according to their access privileges.

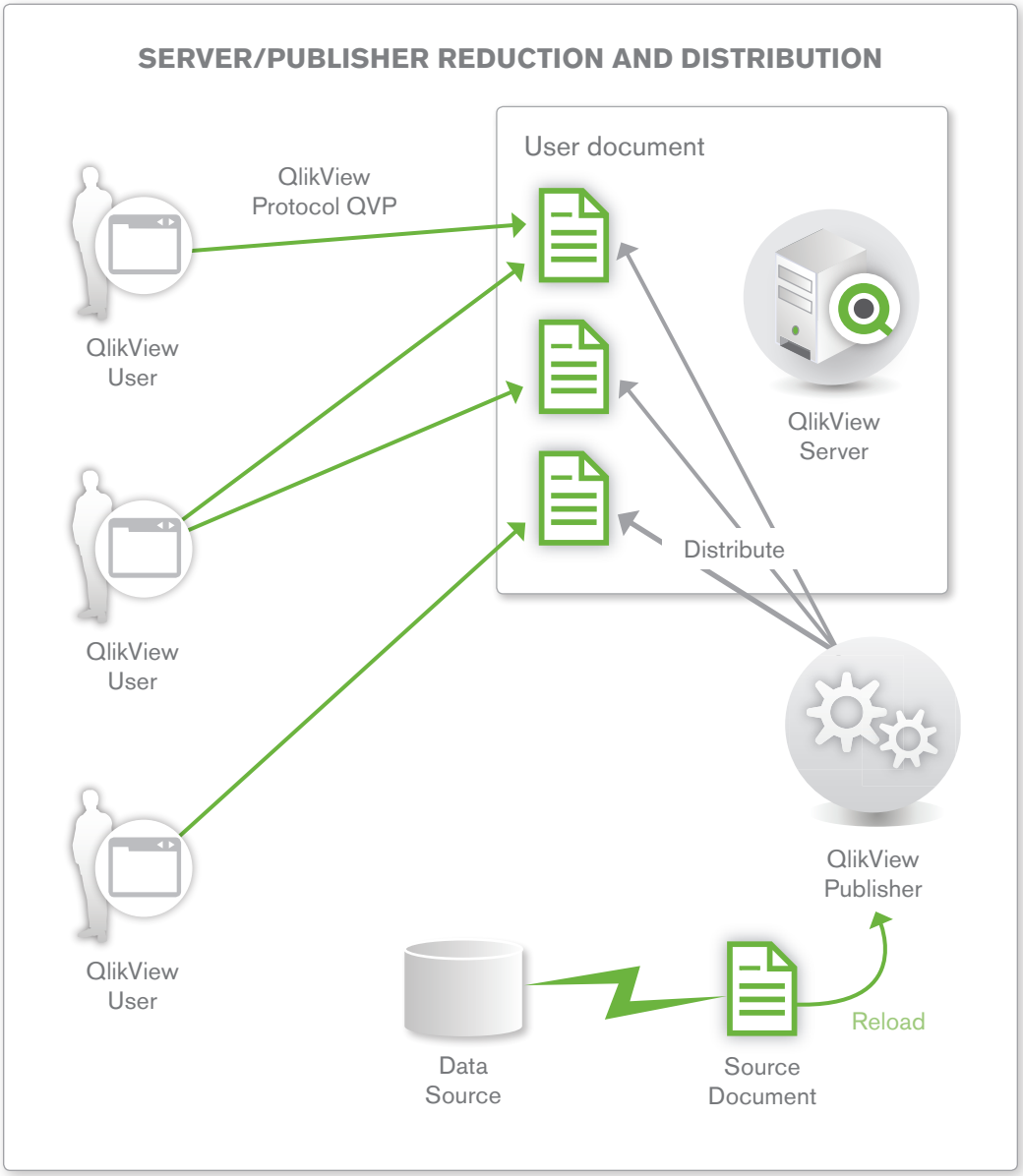
One of the benefits of reducing and distributing source files in this manner is that the documents that are created in this process contain no explicit reference to the source data (e.g. a database connection string) in their script environments. Therefore, if a user is interacting with the document via the desktop client, they will not be able to see the location of the source data. All of the data pertinent to their needs is contained in the document.

An Administrator can use the QlikView Enterprise Management Console to create tasks on source .qvw or .qvd files to accomplish this. At a basic level, the steps are as follows:

1. On the source document (either .qvw or .qvd), apply data reduction criteria (e.g. choose field name on which to reduce the data)
2. Apply distribution criteria to newly created (reduced) files.
    a. Assign authorization privileges using either DMS or NTFS ACL’s.
    b. Choose type of distribution (e.g. qvw files or pdf report)
    c. Chose location for newly created files
3. Apply notification criteria for completion of task (e.g. email notification)

The newly created files will *only* contain the data that the user or group is authorized to see because the data has been ‘reduced’ from the master document according to the reduction criteria set out. This is why the process is termed ‘Static Data Reduction’. Therefore, there is no chance of an unauthorized person viewing data because only authorized data exists in each file.

Figure 8: Server / Publisher Reduction and Distribution



COMBINING SECTION ACCESS AND SERVER / PUBLISHER REDUCTION AND DISTRIBUTION

It is often desirable to merge the capabilities of Section Access and the Reduction/ Distribution capabilities from the centrally managed server.

Reduction and Distribution tasks for static reduction combined with Section Access for dynamic data reduction will give you a flexible way of protecting the information. An example may be the distribution of Country or Division specific data to a collection of remote users (using static reduction), who can then see only their local data that they are individually permitted to see (dynamic reduction).

Using the two techniques, both row and field level security can be accomplished for the same master document. Row-level security can be accomplished using Reduction/ Distribution on the server side. For example, a source document might contain a field containing 3 unique geographical regions. This file can be reduced to produce 3 new files containing data pertaining only to each region respectively. Further field-level security can then be applied by invoking a Section Access script at the file level, restricting any given user (or group) to entire fields of data.

Conclusion: Security is an absolute requirement for enterprise deployments

This paper has outlined how security in QlikView is approached and executed, taking into account overall deployment architecture topics in addition to specific topics about user authentication and authorization and integration with existing security infrastructure.

For any enterprise software solution in production, security is an absolute requirement and IT professionals from a DBA all the way to the CIO are tasked with ensuring that

a) Unauthorized access to data never occurs

b) Data is made readily available to those who need it

Implementation of these twin goals can often be at odds with each other and it is important that software vendors and implementation partners provide clear and accurate information related to how their solutions adhere to any organization's security standards.

QlikView provides robust means of securing data utilizing industry standard technologies such as encryption, access control mechanisms, and authentication methods. It easily integrates with authentication methods via standard directory services, allows administrators to choose between the creation of user-defined ACL's or utilizing Window's ACL's for file authorization and has a range of capabilities for data-level security at both the file and the server level.