

An introduction to QlikView Security using Web Tickets

<http://community.qlikview.com/docs/DOC-3614>

QlikView provides a range of features to allow users to login and view documents using Single Sign On with a range of different technologies. A high level document detailing all of the various security options in QlikView can be found [here](#) this article covers the “Web Ticketing” approach and provides an example to get started from.

This document is intended for QlikView professionals and requires some understanding of security technologies and a small amount of programming and web technologies.

So what is a Web Ticket and when would I use one?

At a high level a QlikView Ticket is a way to seamlessly transfer the identity of a user from one system into QlikView. This however can be useful approach in a number of scenarios so lets look at where you could use it.

Please note: Web Ticket was added to QlikView in version 11, an older version of a similar method was available prior to this but carried a number of limitations. From this point forward we will focus on QlikView 11 and Web Ticket.

Scenario 1 – im already logged into a portal or website and now want to use QlikView

Let's say you are logged into your company's in-house built sales portal, when you logged in you gave this portal your user ID and password and the portal checked these against a database to be validated. Once logged in it creates a session for you and lets you look at all the pages you are permitted to. This application now trusts who you are.

Now your company has added QlikView and they want users to be able to navigate between content on the portal server and content on the QlikView server but they want it to be seamless and you shouldn't need to log in again, only the applications know nothing about each other and QlikView doesn't know about the database with user IDs in.

QlikView needs a way to trust the users session from portal when being passed over. Using a the QlikView web ticket mechanism it is possible to set up a hand shake between the portal and QlikView so the portal can securely transfer a users session over. QlikView has established a trust between itself and the Portal application to be able to identify the user when they browse to QlikView content.

Scenario 2 – QlikView used in isolation, users stored in a database or LDAP

Lets assume we have list of user IDs and passwords held in a database or LDAP directory. QlikView doesn't provide a “custom” login page that can handle the authentication of these users.

In this scenario a login page can be build which handles the user authentication part and then uses the Web Ticket mechanism to pass that user identity into QlikView. In this case QlikView has still set up a Trust with an application, as in the above case, only here that application is just a single web page.

Other Scenarios

There are a range of other scenarios where this approach could be used such as those listed below. Of course there is more than one way of achieving security integration and using Web Ticket is just one method of achieving them:

- Login with Group information/SAML – Web Ticket allows you to not only send the username but the membership of groups too. This can make administration of what the user can subsequently see much easier
- Login from dozens of online systems – eg Login with your Facebook ID, Google account, Salesforce.com account. If you do not wish to hold any user credentials for QlikView you could rely on online sources for identity also

How does Web Ticket work

Below are the steps that typically occur when using ticketing, sometimes there can be more or less steps depending on the scenario.

1. A new login page is created and its first task is to obtain the Identify of the user
 - If the page is being added to a web Portal or application this can easily get the current user ID from their active session
 - If the page is standalone it might provide a form to enter a user id and password and validate them first (this is the example provided below)
 - In either case a range of technologies could be used to do this.
2. QlikView is set up so that it trusts this login page to pass over validated users (see below for options)
3. The login page now makes a request to QlikView which says “I have a user I Trust called FRED, can I have a ticket for him”

4. QlikView returns a ticket to the login page which takes the form of a long string and internally records the user that is represented by that ticket
5. The login page now builds a URL to the QlikView server address adding the ticket into the URL, the user is then redirected to that URL
6. When the user hits the QlikView server, it extracts the ticket checks its list of approved tickets and if its valid, the user is then successfully logged into QlikView as FRED

A couple of notes on web tickets

1. A ticket once created is valid for 2 minutes, if it is not used in that time is destroyed
2. Once a ticket is used to establish a session it cannot be used again. Within that session a user can open several documents
3. A web ticket can be used to open a single document or the QlikView Access Point
4. Web Tickets can be used with features such as Clustering and Document chaining
5. For information QlikView 10 and prior ticketing had the following limitations that are NOT present when using Web Ticket.
 1. Could only open one document, and could not use the Access Point
 2. Could not use clustering
 3. Could not support document chaining

How to implement Security using Web Tickets

To use web tickets there are two areas that need to be addressed: **The Code** in your login page and the **Server Configuration**.

The configuration of QlikView:

- First of all, while not mandatory it can be easier to use IIS for the QlikView Web Server when working with ticketing. It may be required to host your login page anyway.
- QlikView needs to be an Enterprise Edition Licence
- QlikView needs to be running in DMS mode for security (see manual)
- The QlikView web site in IIS needs to be set up to use Anonymous permissions – it will be expecting windows permissions by default – specifically it is the QVAJAXZFC directory that needs its permission changing.
- QlikView needs to trust the code asking for the ticket. There is a web page within the QlikView web server called **GetWebTicket.aspx** which handles requests for tickets, this will only return a ticket to a trusted user/process and this is identified using one of two options
 - Option 1 – use windows permissions
 - The code or process asking for a ticket needs to run as or provide a windows user identity. The user ID must be a member of the QlikView Administrators windows group on the QlikView server.
 - As the GetWebTicket page runs under the QVAJAXZFC directory on the web server and one of the above steps made the directory work with Anonymous users –**for this page only**– enable windows authentication in IIS
 - Now the Login page must authenticate itself when requesting a ticket as a windows user and that user must be a QlikView Administrator. In the attached example the login is hardcoded, however this could be configured in other ways
 - Option 2 – use an IP address white list
 - For some code technologies NTLM is not available or it may just not be appropriate to use it. For these scenarios a “white list” of approved IP addresses can be used rather than a named user. Using this approach the GetWebTicket page will only return a ticket to code running from a specific IP address
 - To configure this option:
 - Open the web server config file from C:\ProgramData\QlikTech\WebServer\config.xml
 - Locate the line `<GetWebTicket url="/QvAjaxZfc/GetWebTicket.aspx" />`
 - Replace it with the following specifying the IP address(s) of the web server(s) running the code


```
<GetWebTicket url="/QvAjaxZfc/GetWebTicket.aspx">
  <TrustedIP>192.168.0.1</TrustedIP>
</GetWebTicket>
```
 - Save and restart the web server

The Code & Example:

The function to request a ticket is a fairly simple HTTP post command which returns a ticket as a string, this method can be used by a wide range of technologies. The examples provided here use asp.net but it can easily be adapted for PHP, Java and others.

In its raw form the HTTP Post entry to the getwebticket.aspx page on the QlikView server looks like this:

```
<Global method="GetWebTicket">
```

```
<UserId>FRED</UserId>
<GroupList><string></string></GroupList>
<GroupsIsNames>true</GroupsIsNames>
</Global>
```

The response containing the ticket should look like this:

```
<Global><_retval_>ABC123XYZ567FGH456</_retval_></Global>
```

The above raw commands need to be embedded into the code for the login form, the attached example is intended to be a basic demonstration of how to use these commands using asp.net. The example provides no actual logic to authenticate the user against anything but provides the framework to add it in. The example takes the form of a simple login page asking for a user ID, password and optional groups, for cases where you already have the identity of the user then the UI from the example could be removed to give a seamless login from one system to another.

The first part of the example is the login form, this is just a simple form with three boxes.

QlikView WebTicket Example

Username:

Password:

Groups:

Once the login button is clicked there is a Login function that is called. This function in the example does nothing and always says the user is valid: here you would need to provide code to suit your environment, the example just trusts the entry automatically.

```
loginOK = ValidateUser(username, password);
```

Following user validation the next function makes the HTTP post to request a web ticket. The code requires the input of the URL of the QlikView server and in this case embeds a userid and password which has permissions to ask for a ticket (see notes on setting up QV trusts).

```
ticket = getTicket(username, groups, ticketinguser, ticketingpassword);
```

Once the ticket is returned a URL is built and the user is redirected to either the access point or to a single document.

```
Response.Redirect(RedirectLink);
```

You should now be redirected to the QlikView AccessPoint and be logged in at the user specified in the form and see the documents that user is permitted to see.

Welcome MyTestUser | Favorites & Profile


Last updated Tue Nov 06 2012 14:59:36 GMT+0000 (GMT Standard Time)

QlikView

AccessPoint

Showing 1-1 of 1 1 12 items per page

Category: Attribute: View as:



★ Sales Compass.qvw
 Last Update: 2011-09-29 16:58
[view details](#)

This is a basic example of how to using the web ticket function. I have written comments into the code example so that you may be able to follow through what is happening. I have also left the example as a single aspx file so you can edit it and use it with Notepad rather than visual studio.

I hope it is helpful in building you own login pages to QlikView using Web Ticket!!