PYRAMID
ANALYTICS

BI Office

Web Authentication Model Guide

Version 6.5

Contents

1. Web Authentication Model Overview

- Basic Authentication
- Windows Authentication
- Forms Authentication (which comes in 2 variations: “Direct” and “Federated”)

A. Basic Authentication Models

The user is prompted to enter credentials when they browse to the BI Office URL address. The credential prompt is supplied by Windows IIS and is authenticated against the local OS security or the Active Directory (see figure 1 below). The resulting security token, returned to IIS, is then passed on from the web application to BI office which in turn uses is directly against cube data sources without any further translation. The user name and password are passed from the client web browser to the server in clear text so Basic Authentication models are usually deployed with SSL certificates to encrypt the data packets across the network.

Basic Authentication works through firewalls (ports 80 or 443) and universally works on all browsers on both PC’s and MAC’s. It’s a mature, efficient and incredibly fast authentication method and is highly recommended for extranet deployments.

In the figure 1 example below, the users could be authenticated using an “EXTERNAL” Active Directory sitting in a DMZ. Often, this is convenient when trying to keep external end users separate from internal users.

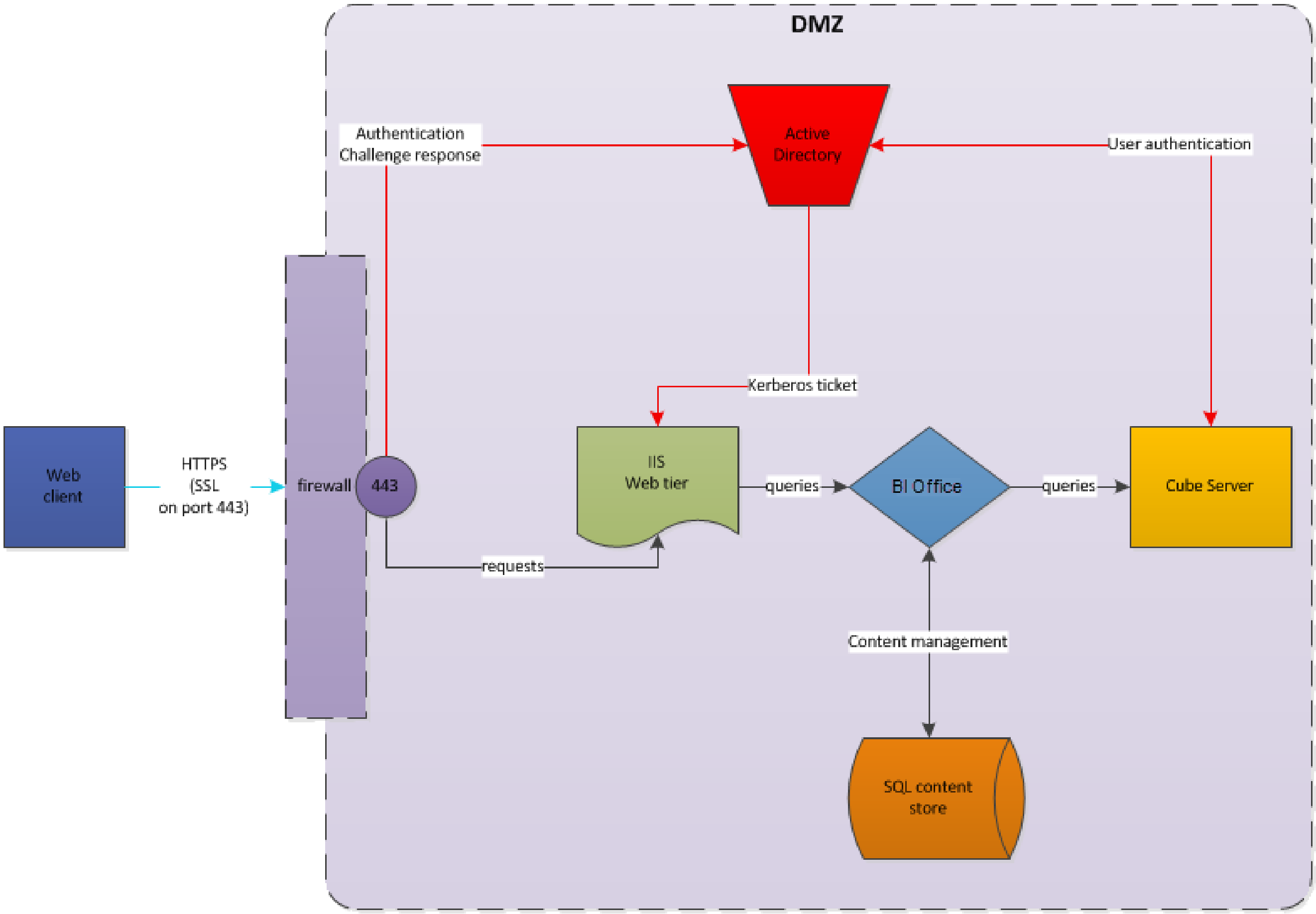


Figure 1 Basic and Windows Authentication

B. Windows Authentication Models

Windows Authentication provides a single sign on model for users of PC's connecting to the BI Office application. The user is NOT prompted when they browse to the BI Office URL address; instead their workstation credentials are used to authenticate against the website automatically. Like Basic, the authentication is handled by Windows IIS and is credentialed against the local OS security or the Active Directory (as per figure 1 above). The resulting security token can be used directly against cube data sources without any further translation.

Windows Authentication generally does NOT work through firewalls and only works on Internet Explorer, Firefox and Chrome browsers on PC's only. Because of these limitations, it is used in limited circumstances. It's a mature, efficient and incredibly fast authentication method and is only recommended for intranet deployments.

In the figure 1 example above, the users could be authenticated using an "INTERNAL" Active Directory sitting in a DMZ – or any other network segment. Apart from the need to manually login, Windows Authentication acts very similarly to Basic Authentication.

C. Forms Authentication Models

Unlike Windows and Basic Authentication, Forms authentication provides a customized entry point for users to websites and web applications that are not inherently managed by IIS. Typically, the user is forwarded to a login page where they are prompted to enter credentials. The credential prompt is supplied by the application itself and is authenticated inside client defined code. The authentication can be against any type of credentialing engine selected by the developers - including against an Active Directory or SQL Server data store. The resulting forms security token is used by the custom application but is CANNOT be used directly against cube data sources and therefore usually requires some type of translation. Further, the user name and password are passed from the client web browser to the server in clear text so Forms Authentication models are usually deployed with SSL certificates to encrypt the data packets across the network.

Forms Authentication works through firewalls and universally works on all browsers on both PC's and MAC's. Because it provides for customized authentication frameworks, it is often used when an Active Directory cannot be used directly (or at all).

Since forms authentication, is by definition, a customized application, there are numerous techniques and technologies that can be demonstrated. To make sure BI Office facilitates as many of these variations, the application provides 2 methods for Forms Authentication. A simple, in-built forms authentication model called "Direct" Forms and more complex model for interfacing with existing forms solutions called "Federated" Forms.

- **Direct Forms** – if deployed, users are redirected to an in-built BI Office login page where users can enter their details. The authentication is applied directly against the Active Directory itself. To consume direct forms, users are simply redirected to the "login.aspx" page in the BI Office Application, where they must login before proceeding.
- **Federated Forms** – is an automated mechanism for clients to redirect users from an alternative login framework to the BI Office Suite. In doing so, clients provide the impersonated Windows account that will be used for the given user. BI Office in turn provides a framework for the end user to auto-login into its application, delivering *a virtual single-sign-on* facility.

Forms Authentication can be deployed by selecting "Forms Authentication" during the installation of the BI Office application. Use of federated forms, however, also requires clients to add new code to their custom forms login process. The code provides a conduit for BI Office to issue a browser based cookie with encrypted tokens that will allow the user's browser session to use the application without further prompt. Figure 2 below illustrates an example of federated forms authentication.

2. How Federated Forms Work

Use the BI Office federated forms authentication solution when you want to enable users from outside of a domain to use BI Office web application without creating a unique Active Directory user for each user; or when you have an external users' database that you want to authenticate against instead of using the Active Directory authentication.

The solution requires very minimal development to integrate in to your own authentication process (see next section).

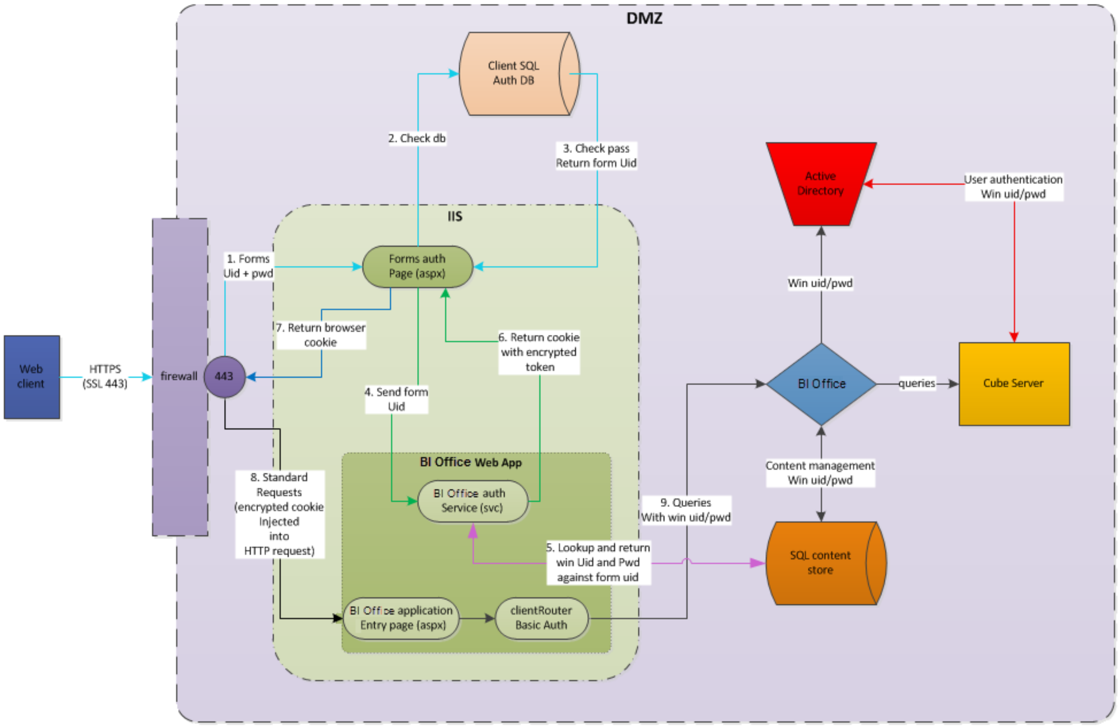

DMZ
Client SQL Auth DB
2. Check db
3. Check pass Return form Uid
IIS
Active Directory
User authentication Win uid/pwd
1. Forms Uid + pwd
Forms auth Page (aspx)
Win uid/pwd
7. Return browser cookie
6. Return cookie with encrypted token
Web client
HTTPS (SSL 443)
firewall
443
4. Send form Uid
BI Office
queries
Cube Server
BI Office Web App
BI Office auth Service (svc)
Content management Win uid/pwd
8. Standard Requests (encrypted cookie Injected into HTTP request)
9. Queries With win uid/pwd
5. Lookup and return win Uid and Pwd against form uid
SQL content store
BI Office application Entry page (aspx)
clientRouter Basic Auth

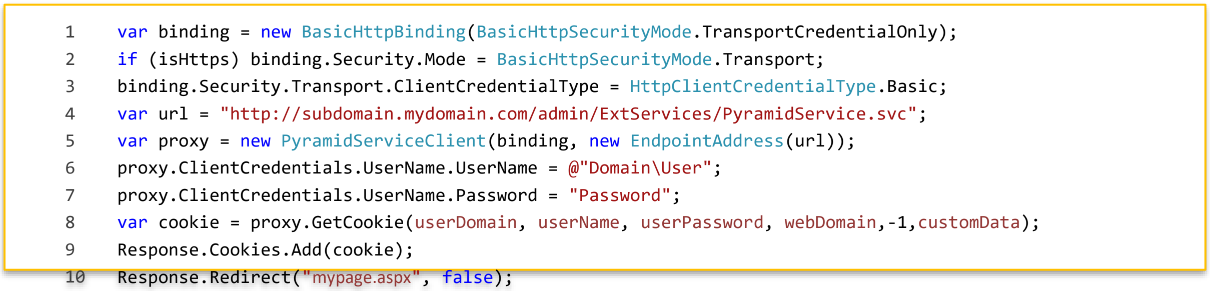
Figure 2 Federated Forms Flow

D. Enabling Federated Forms Authentication

1) The user logins into your application with your custom credentials.
2) Your forms will check the user against your user store (in this case a SQL database).
3) If the credentials pass, the system returns a "mapped' Active Directory user account proxy that will be impersonated for the original user account.
    a) You will need to map each external user account to an internal AD user account proxy. The proxy is used to authenticate against SSAS for querying, metadata and accessing BI Office content.
    b) In its simplest form, all users can be mapped using a single account – effectively exposing cubes without security. At the other extreme, each external account can be mapped one-to-one with a proxy account to deploy fully secured cubes. And anything in between.
4) Your forms code then connects to the BI Office Service to generate the security token and cookie. (See next section for details on this coding exercise). The cookie generation requires user proxy details:
    a) Username
    b) Password
    c) Domain
    d) Cookie domain
5) The BI Office Service returns the encrypted cookie
6) The cookie is added to the end user's browser session cookie collection
7) When the user browses to the BI Office entry page, the cookie is retrieved and checked for a security token. If the token is missing, they are redirected to the base login page provided during product installation.
8) BI Office will now use the credentials in the cookie to query the system.

E. Implementing Federated Forms Security

The installation of the BI Office federated forms solution involves using the BI Office Web Services. The Web Services are installed with the BI Office. Use the following code snippet in conjunction with the explanation below:



```
 1   var binding = new BasicHttpBinding(BasicHttpSecurityMode.TransportCredentialOnly);
 2   if (isHttps) binding.Security.Mode = BasicHttpSecurityMode.Transport;
 3   binding.Security.Transport.ClientCredentialType = HttpClientCredentialType.Basic;
 4   var url = "http://subdomain.mydomain.com/admin/ExtServices/PyramidService.svc";
 5   var proxy = new PyramidServiceClient(binding, new EndpointAddress(url));
 6   proxy.ClientCredentials.UserName.UserName = @"Domain\User";
 7   proxy.ClientCredentials.UserName.Password = "Password";
 8   var cookie = proxy.GetCookie(userDomain, userName, userPassword, webDomain,-1,customData);
 9   Response.Cookies.Add(cookie);
10   Response.Redirect("mypage.aspx", false);
```

1) Adding the External web services service reference to your "Visual Studio" forms project
   a) The external services are under the “[Pyramid Website]/admin/ExtServices/PyramidService.svc”
   b) In your code, add the following lines of code, once the user has been authenticated. You may also need to scope the external service reference in your code page (with “using”).
2) Set up service ‘BasicHttpBinding’ (*lines 1 to 3*)
   a) For HTTPS use : “BasicHttpSecurityMode.Transport”
   b) For HTTP use: “BasicHttpSecurityMode.TransportCredentialOnly”
3) Connect to BI Office services using the service URL “…../admin/ExtServices/PyramidService.svc” (*lines 4 and 5*)
4) Provide credentials for an Active Directory account to connect to the service. The account must be a user that can login into the system. It is DOES NOT have to be the account that will be used for querying the cubes. (*lines 6 and 7*)
5) Next we generate the security token for a given user. The response is a cookie which must be added to the user’s cookie collection. (*line 8*)
   a) To avoid cross-domain issues, the cookie must be added to the same web domain that houses both the main client application and the BI Office application.
   b) The function call requires the following details of the user that will be impersonated: domain, username and password. The prevailing web domain is also required. Currently, the expiration value is ignored, and should be set to “-1”.
   c) If the application is licensed for “anonymous viewers” the optional “customData” field can be set to provide access to custom data switches in the data model queries and security settings.
6) Now add the cookie to the user's session and then redirect the forms as normal to the appropriate landing page. If using “redirect”, use the “false” flag to ensure the cookie is written to the user’s browser. (*lines 9 and 10*)

3. Considerations for Load Balancers

Load Balancers introduce more complexity to the web authentication setup. Below are some implementation guidelines

There are few headaches for Basic and Windows Authentication - since most systems are able to bounce authentication tokens between web servers.

Forms authentication relies on *encrypted* cookies – which pose a challenge in certain load balanced web farms.

- If using either forms model (direct or federated), developers will typically need to persist cookies on the client, to ensure the cookie is able to move from one web server node to the next as the load balancers spread the web requests across multiple servers. This problem becomes even more acute if administrators have chosen to offload SSL processing from the web servers to the load balancing framework. This step can often be avoided if administrators have an alternative strategy for persistence implemented on the load balancing framework. (Techniques for persisting cookies is presented below).
- If cookies are persisted, developers should provide a mechanism for users to “logout” of the application. In doing so, they can then manually delete all persisted cookies – maintaining authentication integrity. This is not an issue if cookies are NOT persisted (the default setting in the BI Office framework).

F. Persisting Cookies

- **Direct Forms** – In the BI Office web.config file, set the value property for “FormsPersistence” to a value above “0”. This is the amount of time to persist the cookie, in seconds.
- **Federated Forms** – Add the following line of code after line 8 in the code snippet above (persisting the cookie by “x” minutes)

```
cookie.Expires = DateTime.Now.AddMinutes(x);
```

G. Processing SSL on Load Balancers with Federated Forms

This model is somewhat complex, as the external facing websites are processed via SSL / HTTPS, while the internal web framework continues to operate under HTTP. Since federated forms require a connection to BI Office’s external services, there are several considerations to be made here:

- If the external forms application is OUTSIDE of the BI Office load balanced framework, the web service address supplied in line 4 above can, in most situations, continue to be addressed to an HTTPS address.
- If the external forms application is INSIDE the same framework as the BI Office application, we recommend setting up an internal URL address for the service that uses HTTP. This will prevent the call from leaving the framework, and re-entering via the HTTPS URL on the load balancers. In this scenario, the BI Office external services framework is addressed by the forms application without going through the load balancers.

4. Installing Multiple Web Sites

The new BI Office has the ability to use multiple websites, each implementing its own authentication method.

For example you can use one internal site with windows authentication and one external with basic or federated forms authentication.

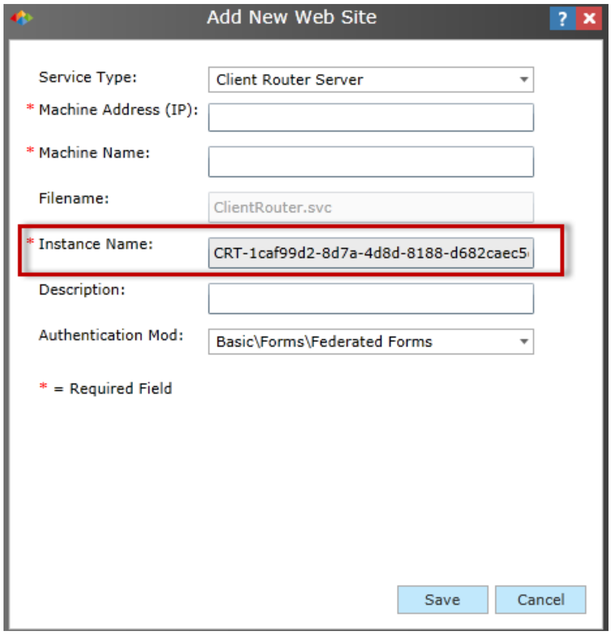
The following explains how to install multiple websites:

- Install BI Office Version 5, single or multi, for help with the installation you can go to “Install guide”, it has a detailed explanation for all the possible ways to install BI Office.
- When running a multi installation, make sure to install “Web Site” is on two different servers.
- Connect both sites to the same database.

Following these steps will allow you to have multiple websites.

5. Manual Steps for Adding a Secondary Web Site

1) Create a copy of the website source folder (C:\Program Files\Pyramid Analytics\BI Office 5\websites\paBio and paBioAdmin).
2) Manually create another web site in IIS referencing it to the copied folders C:\Program Files\Pyramid Analytics\BI Office 5\websites\pabio\ > for Client, and C:\Program Files\Pyramid Analytics\BI Office 5\websites\pabioAdmin\ > for Admin.
3) Set the appropriate authentication method for the new website in IIS (For an explanation on how to configure for each authentication type, go to “Installation Guide” > “Authentication Issues”.
4) Add DNS reference to the web site, or add it to the host file.
5) Go to BI Office Admin console > Click on “Servers” tab and then “Add New Web Site”.
   a) Select “Service Type” > Client or Admin. Fill the Required fields
   b) You will need to repeat the process twice, once for Client and once for Admin.
   c) Copy the “Instance Name” for the next step (See image below).
6) Then go to “C:\Program Files\Pyramid Analytics\BI Office 5\websites\”. Paste the copied “Instance Name” to the “ServiceName” value inside “Web.config”.



Service Type: Client Router Server
* Machine Address (IP):
* Machine Name:
Filename: ClientRouter.svc
* Instance Name: CRT-1caf99d2-8d7a-4d8d-8188-d682caec5
Description:
Authentication Mod: Basic\Forms\Federated Forms

* = Required Field