Working with secure ArcGIS services

ArcGIS Server Web services may be secured to permit only authorized users. The administrator of the server should provide information necessary for you to connect to the service. The information below will assist you once you have the connection and login information.

If you are the administrator of an ArcGIS Server system and wish to restrict access to your ArcGIS Web services, information is available in the "Securing Internet connections and Web applications" chapter in the ArcGIS Server Help. The Help is available on your server or online at ESRI for both .NET and Java .

The way to work with a secured service depends on how the service authenticates users. An ArcGIS Server instance can use one of two authentication methods: token-based authentication or HTTP (including Windows) authentication . Contact your server administrator if you are uncertain about the approach used for the service in your application.

Services secured with token-based authentication

Services secured with token-based authentication require that a token be included in each request for a map, query, and so on. A token is an encrypted string that is derived from information about the authorized user, the date and time, and information about the client making the request. The token helps ensure that only authorized users have access to the service.

Note that if you are using an ArcGIS Online premium service, the token is referenced as a "Development API Key".

To use a service that requires tokens, you must first obtain a token, and then embed the token in your application. The following instructions cover these steps.

Obtaining a token

To obtain a token, use a browser to visit the token service associated with the ArcGIS service. The administrator of the server may provide the URL of the token service. If not, visit the Services Directory for the server that hosts the service. You can enter the URL of the service in a browser's address bar to bring up the Services Directory (it will probably display a login screen). In the Services Directory, click the "Get Token" link in the upper right. If no Get Token link appears, the service either is not secured, or it uses HTTP/Windows authentication . Contact the ArcGIS Server system administrator if you have trouble locating the token service page.

If you are using an ArcGIS Online premium service, the URL to obtain a Development API Key is http://premium.arcgisonline.com/server/tokens/agol_gettoken.aspx

When you visit the token service Web page, note the use of https in the URL. The token service normally uses **HTTPS** to ensure that transmission of user data is encrypted.

In the token service Web page, enter the following information:

- The **user name** and **password** provided to you by the ArcGIS Server system administrator. For ArcGIS Online premium services, use your ESRI Global Account.
- A **client ID** . If you will embed the token in the web page, use the WebAppURL option, and enter the URL of the Web application that the browser will use, for example, http://www.example.com/MyArcGISVEWebApp. If you will be using the proxy page (see below), then use the IP option, and enter the IP address of the server that will send requests to the ArcGIS Server computer.
- **Expiration** time. The token will be valid for the time period chosen here. Shorter expiration periods are safer in the event that the token is intercepted by unauthorized users. But you must obtain a new token and replace the value in the page before the old one expires. Expired tokens will cause the ArcGIS service to refuse requests. When obtaining a Development API Key for ArcGIS Online premium services, you will not have an option to include an expiration date. This date is determined by when you placed your order.

Once you fill out the token form, click Generate. A token should appear below the "Token:" heading. Copy this value and use it in your application as directed below. If no token appears, or an error message occurs, check that the values you entered are correct. Also note that no token will be generated if the token service requires the use of https and your page URL uses http.

If your application is server-based (ASP.NET, JSP, PHP, etc.), then another option is to generate tokens dynamically at runtime. You may use code to send the token request, use the token, detect timeout of the token, and renew the token upon timeout. Since the programming approach is similar to when using SOAP-based services, see the Developer Help topic "Working with the SOAP API" (Overview), available online at the ESRI Developer Network or, if

you have an installation of ArcGIS Server, in your local Developer Help. You will also need to route requests through your server, using an approach similar to that discussed in Using the proxy page .

Using the token in your application

Once you have a valid token, add it to your application. You can include the token in your application in one of two ways: include the token in the HTML page, by setting the Token property of the service; or, use a proxy page, and include the token in the configuration for the proxy page. The proxy page option offers a higher level of protection for the token, if implemented correctly. End users do not have access to the token with the proxy page option. Since the token is not sent to the browser, it also prevents the possibility of interception of the token during transmission of the application page (though the token is still sent from your server to the ArcGIS Server service). See Using the proxy page for information on that approach.

To include the token in the HTML page, set the Token property of the task or service. For example, if you are adding an ArcGIS Server map service to the ArcGIS JavaScript API map, you will use the ArcGISTiledMapServiceLayer class to create the layer. The ArcGISTiledMapServiceLayer class takes a url in its constructor. You can append the token to the url in the query parameter by setting the token property. For example:

```
var token = "5fFo4%2fI4Tv8IGSqSYbpUNORRD%2fYxXMSPopt9CMknpXIjEVqYGm3uuQnU";
var mapServiceURL = "http://premium.arcgisonline.com/ArcGIS/rest/services/ESRI_StreetMap_Wor
var tiledMapServiceLayer = new esri.layers.ArcGISTiledMapServiceLayer(mapServiceURL
+ "?token=" + token));
```

Note the Token in the example above may wrap to multiple lines, but is a single string. The token may also have one or two periods at the end. These periods are part of the token.

Alternatively, you can include the token directly in the URL:

```
var tiledMapServiceLayer = new esri.layers.ArcGISTiledMapServiceLayer

("http://premium.arcgisonline.com/Server/rest/services/ESRI_StreetMap_World_2D/MapServer?

token=5fFo4%2fI4Tv8IGSqSYbpUNORRD%2fYxXMSPopt9CMknpXIjEVqYGm3uuQnU");
```

Each class that communicates with an ArcGIS Server system has a Token property. If you use a combination of multiple tasks or layers, you will need to set the Token property for each. If multiple servers are used, each will require a separate token, even if the user name and password is the same. Services on the same computer can use the same token, unless the services require a different user name and password.

If you are concerned about possible interception of the token during requests from browsers, you may wish to require the use of HTTPS for your application, and to require users to log in to your application. Your token will contain the referrer URL of your application, so normally users could not use the token in a Web application hosted at a different server. However, it is possible to spoof the referrer. Therefore you should assume that if someone intercepts your token, they would be able to use it to gain access to the services used in your application. If you require users to log in and to employ HTTPS for your application, then it is much less likely that the token may be intercepted. Using the proxy page is also a way to prevent transmission of the token to and from the browser.

Services secured with HTTP / Windows authentication

When a request is made to a service secured with HTTP authentication (including Windows authentication using IIS), the server issues an authentication challenge. The application or user must respond with appropriate user credentials in order to continue. You do not supply credentials in the form of a token, but instead must use standard HTTP authentication methods to supply user credentials.

If the service you wish to use in your application to use is secured with HTTP authentication, you may use one of two approaches for your application:

- Do not supply any credentials within your application. Instead, let server challenge the browser user. The user will see a login dialog pop up in the browser. The user must respond with a valid username and password for the ArcGIS Server system that issued the challenge.
  -- OR --
- Use server-side code (ASP.NET, JSP, PHP, etc.) to set an identity for the request. The server sends the request with the identity, and the end user never sees a login dialog. This approach requires that the service request be

sent through your server. This is similar to the option discussed for tokens in Using the proxy page , but you will need to create your own code in the proxy page to set the identity.

For the first option above, where the browser user logs in via a pop-up dialog, you must furnish the end user with a username and password that is recognized on the ArcGIS Server system and that is permitted for the service. You must work with the administrator of the ArcGIS Server system to ensure that end users have login information. If you are the administrator of the ArcGIS Server system, consult the Help, under the topic on securing services, for information on creating and managing user accounts.

If the authentication method used is HTTP Basic, then you should require that users of your application employ HTTPS when accessing it. HTTP Basic does not encrypt data transmitted to the server, and therefore user passwords can be intercepted during transmission. Other authentication methods, such as Digest or Integrated Windows Authentication, may protect user logins, but for maximum security, HTTPS is recommended when users are logging in.

Supplying end users with username and password is not appropriate when services from more than one ArcGIS Server system are used in an application. If more than one server is used in an application, the end user would see multiple login dialogs, and would likely not know which server is issuing the challenge. The multiple logins would be required even if the same username and password is set for both servers. This limitation does not apply when using multiple services within the same ArcGIS Server system, since the challenge is issued for the entire server.

Another limitation with the end-user login approach is that the login dialog is presented only when the initial request to the secure service occurs. For example, say your application performs a query to a secure ArcGIS Server at some point, but initially displays a map from a non-secure map service. The user fills out a form and clicks a button to perform the query. Only when the user clicks to run the query will the login dialog pop up. A workaround would be to send a request in the background to the ArcGIS Server system when the application starts, such as a simple REST request for service information. The user would be prompted to login on startup rather than when using the application.

The second option above, where server-side code sets the identity, requires that your application be written using a server-side API, such as ASP.NET, Java/JSP, or PHP. The process of setting the identity in your code is similar to when using SOAP-based services. See the Developer Help topic "Working with the SOAP API" (Overview), available online at the ESRI Developer Network or, if you have an installation of ArcGIS Server, in your local Developer Help. As a starting point for routing requests through your server, you may be able to modify the proxy page available at Using the proxy page for use with this option.

Services secured with HTTP / Windows authentication

When a request is made to a service secured with HTTP authentication (including Windows authentication using IIS), the server issues an authentication challenge. The application or user must respond with appropriate user credentials in order to continue. You do not supply credentials in the form of a token, but instead must use standard HTTP authentication methods to supply user credentials.

If the service you wish to use in your application to use is secured with HTTP authentication, you may use one of two approaches for your application:

- Do not supply any credentials within your application. Instead, let server challenge the browser user. The user will see a login dialog pop up in the browser. The user must respond with a valid username and password for the ArcGIS Server system that issued the challenge.
  -- OR --
- Use server-side code (ASP.NET, JSP, PHP, etc.) to set an identity for the request. The server sends the request with the identity, and the end user never sees a login dialog. This approach requires that the service request be sent through your server. This is similar to the option discussed for tokens in Using the proxy page , but you will need to create your own code in the proxy page to set the identity.

For the first option above, where the browser user logs in via a pop-up dialog, you must furnish the end user with a username and password that is recognized on the ArcGIS Server system and that is permitted for the service. You must work with the administrator of the ArcGIS Server system to ensure that end users have login information. If you are the administrator of the ArcGIS Server system, consult the Help, under the topic on securing services, for information on creating and managing user accounts.

If the authentication method used is HTTP Basic, then you should require that users of your application employ HTTPS when accessing it. HTTP Basic does not encrypt data transmitted to the server, and therefore user passwords can be intercepted during transmission. Other authentication methods, such as Digest or Integrated Windows Authentication, may protect user logins, but for maximum security, HTTPS is recommended when users are logging in.

Supplying end users with username and password is not appropriate when services from more than one ArcGIS Server system are used in an application. If more than one server is used in an application, the end user would see multiple login dialogs, and would likely not know which server is issuing the challenge. The multiple logins would be required even if the same username and password is set for both servers. This limitation does not apply when using multiple services within the same ArcGIS Server system, since the challenge is issued for the entire server.

Another limitation with the end-user login approach is that the login dialog is presented only when the initial request to the secure service occurs. For example, say your application performs a query to a secure ArcGIS Server at some point, but initially displays a map from a non-secure map service. The user fills out a form and clicks a button to perform the query. Only when the user clicks to run the query will the login dialog pop up. A workaround would be to send a request in the background to the ArcGIS Server system when the application starts, such as a simple REST request for service information. The user would be prompted to login on startup rather than when using the application.

The second option above, where server-side code sets the identity, requires that your application be written using a server-side API, such as ASP.NET, Java/JSP, or PHP. The process of setting the identity in your code is similar to when using SOAP-based services. See the Developer Help topic "Working with the SOAP API" (Overview), available online at the ESRI Developer Network or, if you have an installation of ArcGIS Server, in your local Developer Help. As a starting point for routing requests through your server, you may be able to modify the proxy page available at Using the proxy page for use with this option.