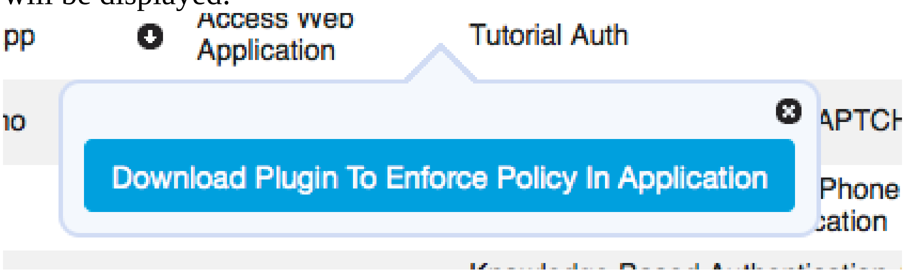
The Web Application variant of Trust Tag is well suited for access control of web applications. In this interactive guide we have used a combination of Database Policy Authority and Phone Authentication to demonstrate web application access control through Trust Tag. This interactive guide is available to users who have created an account on Resilient Access. The Database Policy Authority extracts the name and phone number entered during user registration and passes the name as an identity attribute to this Wordpress application and performs a Phone Authentication using the phone number.

Access Web Application Trust Tag consists of a few server side scripts that are available as a plugin for a web application. By adding a single line of code in the main template/layout file of the web application, access to the application controlled through the policy defined in Resilient Access. The steps for integration with your web application are as follows:

1. Create the policy you wish to enforce in Resilient Access, you must specify a valid hostname where the web application will be hosted, e.g. ***https://example.com***. Select the **Access Web Application** tab and then select the W**eb Application Technology** your web app is built using and the name of the web application (context path). Enter "/" if the web application will be installed in the root context.
2. In the policy list page click on the ⬇ icon in the **Policy Used For** section, the following popup will be displayed:



3. Click on the **Download Plugin to Enforce Policy in Application** button to download the plugin ZIP file.
4. Extract the plugin ZIP into the root directory of the web application
5. In the main template/layout file of the web application enter a single server side include statement at the top of the page to enforce the policy for the web application. e.g

```
For PHP: <?php include 'rns/enforcePolicy.php';?>

For JSP: <%@include file="rns/enforcePolicy.jsp"%>

For Python: from rns import enforcePolicy
```

Access Web Application Trust Tag uses a cookie to keep track of policy evaluation context and the parameters for the policy. This cookie is created in your applications domain and is named as follows: **RNS.<app name>.Policy**. The expiration of the cookie is set to the value specified for **Access expires in** field while creating policy. The policy will be enforced again when the expiration time is reached.

The plugin consists of the following four server scripts:

1. **enforcePolicy**: This is the entry point script that is referenced through the server side include. It checks if the Trust Tag cookie exists, if not, it will redirect to the login script to enforce the policy. If cookie exists, the application pages will be rendered. This script file also contains the javascript code that injects the widget consisting of the Logout button and policy parameters that you see in the top right corner of the application.
2. **login**: The only content in this file is the Trust Tag script tag embedded in <head> section. The policy evaluation UI is rendered in this page. If you wish to render your applications header/footer or other widgets during policy evaluation you may edit this file.
3. **postCredential**: This script will be invoked by Trust Tag when the policy evaluation is successful. This script received policy evaluation context information. The script creates the cookie and places the context information in it.
4. **logout**: Called when the Logout button is clicked. It removed the Trust Tag cookie and closes the Trust Network session and redirects to the home page of the application which will again redirect to the login page through the **enforcePolicy** script

Access Web Application Trust Tag uses Cross Origin Resource Sharing (CORS) to facilitate the secure execution of cross-domain Javascript code. This feature works in all modern browsers as shown in the Browser Support section of the Wikipedia link. In order to successfully execute Access Web Page Trust Tag, the web page should be served from the domain value specified in the **Application Hosting Domain** field including the HTTP scheme, e.g. ***https://www.example.com***