Salesforce Policy Authority is only available within Salesforce. It executes a Salesforce Object Query Language (SOQL) Query and retrieve attributes from Salesforce objects. These attributes can be mapped to the authority parameters of the output policy configured. The SOQL query should only return one result row for the authority to correctly forward the evaluation to the output policy.

Configuring the Salesforce Policy Authority

To configure the authority, complete the following steps.

1. Sign into the Administrative Console.
2. Click **Create New Authority**.
3. Ensure that **Salesforce Policy Authority** is selected in the **Authority Type** list box, the following
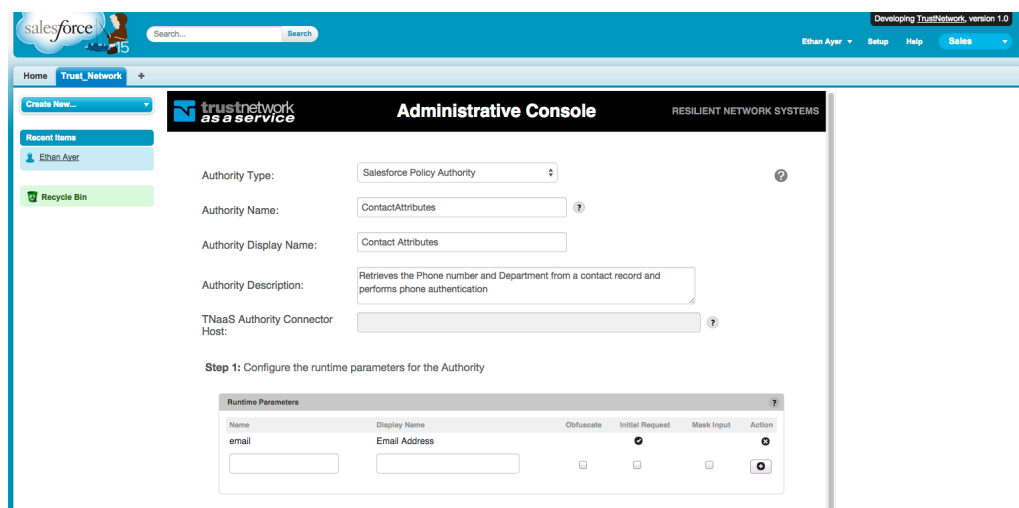
4. Type a name for the authority in the **Authority Name** box.
5. Type a name for the authority in the **Authority Display Name** box.
6. Type a description of the authority in the **Authority Description** box (optional).
7. Use the **Runtime Parameters** area to add and configure the parameters without literal values. The values of these parameters will be supplied by the end user at runtime. For each runtime parameter, specify the following:
   ○ Type the name of the parameter in the **Name** box. The parameter name gets paired with the value provided at runtime and sent to the authority.
   ○ Type the label of the box displayed to the end user in the **Display Name** box.
   ○ If the parameter will contain a sensitive value, such as personally identifiable information, select the **Obfuscate** check box. This instructs the Trust Network to substitute an opaque token for the value as it transits the network, ensuring that the value never passes through the central Policy Workflow Engine component and does not get stored in the Trust History.
   ○ If the user will provide the value in the initial request form, select the **Initial Request** check box. NOTE: Resilient recommends leaving the **Initial Request** check box blank if the value is sensitive or contains personally identifiable information.
   ○ Select the **Mask Input** check box to mask the values with bullet characters as the user types them in. This protects against shoulder surfing.

screen will appear.
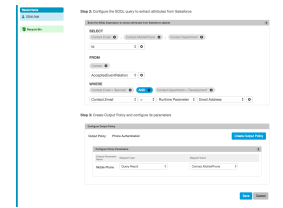
1. First select the Salesforce object to query from the list of Salesforce objects under the **From** drop down and click on the + button to add the object to the expression. Any number of Salesforce objects can be selected to build the SOQL query.
2. After adding a Salesforce object, the object field selector will appear under **SELECT**. Select the object field that you wish to add to the SOQL query and click the + button to add it to the query. The fields in the query should include the attributes you wish to extract and the fields that will be used in where clauses
3. After adding a field the controls for adding WHERE clauses will appear. The left-hand side of the WHERE clause expression will be populated with the fields from the SELECT section, the right-hand side can either be *Runtime Parameter* a *SOQL Field* or a *Literal*. The widget next to it will be appropriately populated to specify the right-hand side value. Click the + button to add the WHERE clause.
4. The **SELECT** field or the **FROM** object or the **WHERE** clause can be delete by clicking on the delete icon. The SOQL query will be updated as required to ensure it always remains a valid SOQL query.

9. Define the output policy and configure the policy parameters using the steps below
    1. Click the **Create Output Policy** button to define the output policy. The output policy is created in a popup window with a similar interface as the Create Policies page. Drag and drop authorities and define the output policy
    2. The **Configure Policy Parameters** table will list the parameters of the output policy. These can either be mapped to *Runtime Parameter* defined above or a *Literal* or a *Query Result.*
        - If **Mapped Type** is *Runtime Parameter* then **Mapped Value** will be populated with the runtime parameters defined. Select the one to use from the drop-down
        - If **Mapped Type** is *Literal* enter the value in the **Mapped Value** box
        - If **Mapped Type** is *Query Result* then **Mapped Value** will be populated with the fields from the SOQL query defined in the **Build the SOQL Expression to extract attributes from Salesforce object** section. Select the one to use from the drop-down.
10. Once you have finished configuring the authority, click **Create** or **Save**.

8. Build the SOQL Expression to extract attributes from Salesforce records.