# [Splunk Custom Visualization - Tree Layout Chart](https://confluence.airbus.corp/display/A2T58ANSR/Splunk+Custom+Visualization+-+Tree+Layout+Chart)

Custom Visualizations give new interactive ways to visualize data during search and investigation, and to communicate results in dashboards and reports. After installing this app you’ll find a Tree Layout as an additional item in the visualization picker in Search and Dashboard.

The tree layout produces a "node-link" diagram that lays out the connections among nodes in a way that displays the relationship of one node to another in a parent-child fashion.

1. Create Tree Layout query

To generate a Tree Layout, use a query that returns results in the correct data format.

1. Query syntax

To generate a Tree Layout visualization, use this query syntax.

... | table <parent>||<child-1>||<child-2> ....

1. Query components

A Tree Layout query includes the following components.

**Parent**

* **Required**
* Indicates parent or root of tree layout

**Child-1**

* **Required**
* Child would be node in the hierarchical structure.

Minimum one parent and one child should be there. remaining children are optional.

1. Search result data formatting

A tree layout query returns results formatted as a table.

Check the Statistics tab after running a query to make sure that the results table includes the required one column with data in below format.

**Results table with single column**

| **Column** |
| --- |
| Parent||child 1 |

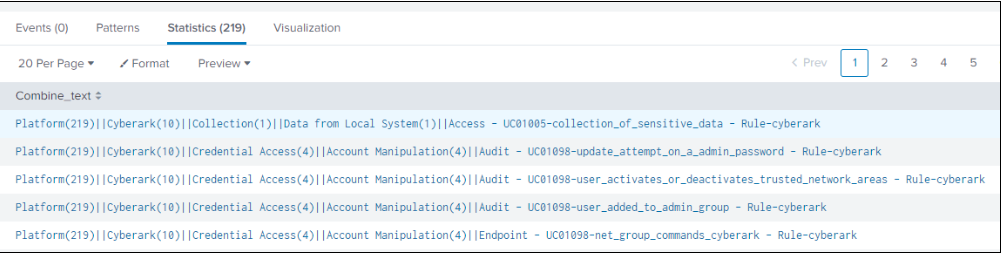
1. Example query 1

below is the query that shows existing use-cases that linked with MITRE ATT&CK Techniques and Tactics

...| eval combine\_text= Type\_n\_Count."||".Tactic\_n\_Count."||".Technique\_n\_Count."||".Use\_cases|table

combine\_text

The query generates the following results table.



1. Example query 2:

below query takes Airbus Roof "ACDC" as parent node its Platforms and domains as children

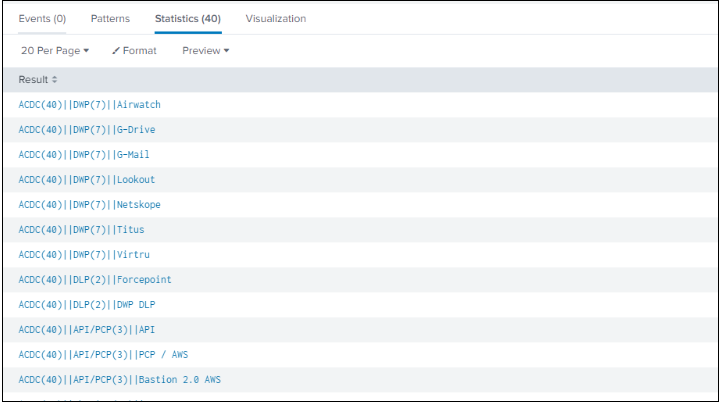
... eventstats dc(Domain) as root\_ct by Root

|eventstats dc(Domain) as platform\_ct by Platform

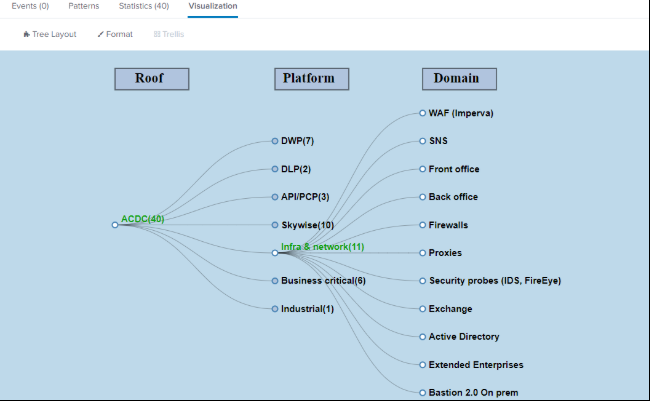
|eval platform = Platform."(".platform\_ct.")",root = Root."(".root\_ct.")", Result = root."||".platform."||".Domain

|table Result

The query generates the following results table.



Below is the visualization.



1. Customize  tree layout

Use the **Format** menu to customize a Tree Layout Chart. The menu has panels for the following settings.

1. General

The General panel includes height, width and caption option.

1. **Link Length**

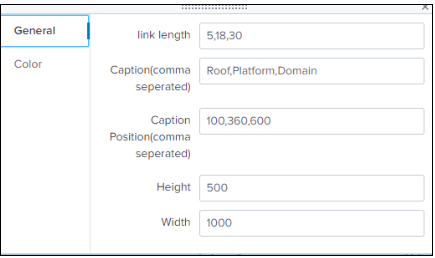
depends on parent and child node link length can be fixed.

1. **Caption/Caption position**

Label for each node section can be updated with its position.

1. **Height/Width**

height and width of tree layout can be fixed here.



1. Color Option
2. **Background**

We can go with background color or image. for image need to specify URL.

1. **Node Color**

Color of the Circle at end of each link.

1. **Text Color**

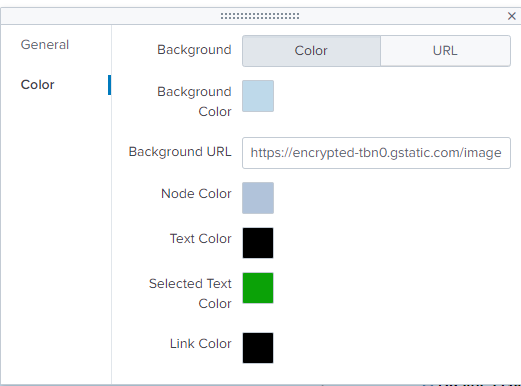
Color of Text and Caption text color too.

1. **Selected Color**

If a node is selected to view its children that color will be changed. This option decides color of selected node's text

1. **Link Color**

Color of link can be changed using this option.



1. Simple XML for Tree Layout

Custom visualizations use a name spaced syntax for Simple XML. Specific visualization properties are appended to the app and visualization context.

1. Add a Tree Layout to a dashboard panel

You can add a tree layout to a dashboard panel using Simple XML. Use this syntax to specify the app and visualization context.

<viz type="tree\_layout\_app.tree\_layout\_chart">

</viz>

1. Configure tree layout properties

To configure a tree layout property, add any of the following property names and a value to this syntax.

<option name=

"tree\_layout\_app.tree\_layout\_chart.[property\_name]">[property\_value]

</option>

| **Option name** | **Accepted values** | **Default** | **Example** |
| --- | --- | --- | --- |
| linklength | any number (depends on number of children) | 5,15,23,35,52 | <option name="tree\_layout\_app.tree\_layout\_chart.linklength">5,15,23,35,52</option> |
| label | category of each children section |  | <option name="tree\_layout\_app.tree\_layout\_chart.label"></option> |
| label\_yaxis | y axis distance of label depends on children's position |  | <option name="tree\_layout\_app.tree\_layout\_chart.label\_yaxis"></option> |
| height | height of chart in pixel(px) | 500 | <option name="tree\_layout\_app.tree\_layout\_chart.height">500</option> |
| width | width of chart in pixel(px) | 1000 | <option name="tree\_layout\_app.tree\_layout\_chart.width">1000</option> |
| background | [ color | url ] | color | <option name="tree\_layout\_app.tree\_layout\_chart.background">color</option> |
| background\_color | Any valid CSS color string | #bed9ea | <option name="tree\_layout\_app.tree\_layout\_chart.background\_color">#bed9ea</option> |
| background\_url | Any valid image url |  | <option name="tree\_layout\_app.tree\_layout\_chart.background\_url"></option> |
| node\_color | Any valid CSS color string | #b1c3da | <option name="tree\_layout\_app.tree\_layout\_chart.node\_color">#b1c3da</option> |
| text\_color | Any valid CSS color string | #000000 | <option name="tree\_layout\_app.tree\_layout\_chart.text\_color">#000000</option> |
| htext\_color | Any valid CSS color string | #0ba207 | <option name="tree\_layout\_app.tree\_layout\_chart.htext\_color">#4B0082</option> |
| link\_color | Any valid CSS color string | #000000 | <option name="tree\_layout\_app.tree\_layout\_chart.link\_color">#000000</option> |

1. Example Simple XML source

Here is an example dashboard panel.

<dashboard>  
<label>Tree Layout Chart</label>  
<row>  
<panel>  
<viz type="tree\_layout\_app.tree\_layout\_chart">  
<search>  
<query>| inputlookup acdc.csv| eventstats dc(Domain) as root\_ct by Root |eventstats dc(Domain) as platform\_ct by Platform|eval platform = Platform."(".platform\_ct.")",root = Root."(".root\_ct.")", Result = root."||".platform."||".Domain|table Result</query>  
<earliest>1585630800.000</earliest>  
<latest>1585720475.000</latest>  
<sampleRatio>1</sampleRatio>  
</search>  
<option name="drilldown">none</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.background">color</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.background\_color">#bed9ea</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.background\_url"></option>  
<option name="tree\_layout\_app.tree\_layout\_chart.height">500</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.htext\_color">#0ba207</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.label">Roof,Platform,Domain</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.label\_yaxis">100,360,600</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.link\_color">#000000</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.linklength">5,18,30</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.node\_color">#b1c3da</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.text\_color">#000000</option>  
<option name="tree\_layout\_app.tree\_layout\_chart.width">1000</option>  
<option name="trellis.enabled">0</option>  
<option name="trellis.scales.shared">1</option>  
<option name="trellis.size">medium</option>  
</viz>  
</panel>  
</row>  
</dashboard>