# Threat View – Splunk Enterprise and D3 Information:

By: Danae O’Connor

Table of Contents

[Threat View – Splunk Enterprise and D3 Information: 1](#_Toc164936645)

[Introduction and Purpose & Note from the Author: 1](#_Toc164936646)

[Quick Overview of Major Items of Importance: 2](#_Toc164936647)

[Custom Visualization Creation - Splunk: 3](#_Toc164936648)

[Custom Visualization Tutorial & OS Setup: 3](#_Toc164936649)

[Inputting Data into Splunk Enterprise: 5](#_Toc164936650)

[Basic Query & Visualization Navigation: 6](#_Toc164936651)

[D3 Functionalities and Interactions with Splunk: 7](#_Toc164936652)

[Threat View Specific Items: 7](#_Toc164936653)

[Debugging Tools for Splunk: 7](#_Toc164936654)

## Introduction and Purpose & Note from the Author:

This document is made as a jumping off point for the back-end development for this project’s Splunk Enterprise Visualization – Threat View and what could be helpful to you or your team for maintenance work, overhauling, or app functionality for Threat View.

In this document, we will be explaining what we’ve have learned about Splunk Enterprise, Splunk Visualizations, D3 interactions, and other items that have been found while making a custom Splunk Visualization. It is assumed that whoever is taking over this project/application has no previous experience with any of these topics. As such this will cover what Splunk Enterprise is and what it does in relation to Custom Visualizations such as Threat View, as well as the construction that went into is and solving some of the problems that our team ran into, and how items such as the D3 visualization tools that Threat-View runs off of work.

The hope of this document is that no-one will have the need for this in the first place, but if they should ever need this document, it will be here. This should hopefully save hours of digging through files, searching for hours on forums with no progress, struggling with random items, failing to do what you want, and helping to create a Custom Visualization as my team and I have done.

## Quick Overview of Major Items of Importance:

It is important to understand that SPLUNK and Splunk Enterprise are NOT the same thing. Splunk is often a nickname for Splunk Enterprise but SPLUNK itself is a service provider for items like Splunk Enterprise which are to help visualize data from various sources primarily associated with cyber-security. Splunk Enterprise is a tool that allows for the visualization of data and putting it into a format that can be understood – primarily through items called visualizations, which visually represent the data within the Splunk Enterprise interface in various formats. These visualizations can either be standard – meaning they come with Splunk Enterprise, like a bar-chart – or custom, a unique visualization that is developed not by Splunk Enterprise but can be added into a user’s Splunk Enterprise account/workspace.

Splunk Enterprise really does not care what data is put into it or even how it is inputted, but it does rely on making sure that its inputs have groupings that can be analyzed. These inputs are primarily tables and rows for those tables. However, with that said – visualizations care about what data is inputted otherwise visualizations do not work as intended if the data provided doesn’t match the visualization’s needed inputs. A bubble chart will not operate off the same data a bar chart does. The Threat View application has data requirements that will be discussed under the “Threat View Specific” section.

Splunk Enterprise has two primary ways of data ingestion – lookup tables and streamed data. Streamed data is the data input performed by the Splunk Enterprise system itself while a lookup table is a table that is inputted by the user and is controlled by the user and allows it to work within the streamed data framework – this is sometimes referred to as a ‘static table’ as the table is not modified unless directly told to do so. For our application and testing, we used the lookup table because it allows for items to be static but can still be accessed in the Splunk Enterprise interface for finding bugs and checking items. In addition, setting up streamed data for a product can be especially tricky if you don’t already have systems for that purpose already set-up beforehand. This document does not cover this setup of data as the data we needed to work with couldn’t be acquired by ourselves due to needing separate systems. In addition to this data set-up, it is important that you set-up permissions for yourself to see these tables otherwise either you, the application, or both will not be able to access the data, more on this topic will be discussed in the “Custom Visualization Creation - Splunk” section.

Now if you aren’t a developer for Splunk Enterprise – you will likely want to apply for a Splunk Developer’s license found at <https://dev.splunk.com/enterprise/dev_license/>, if you have not already done so. This license will allow you to upload the application to Splunkbase.

It is also critical to know what operating system you are developing on! Splunk has products that primarily work off Linux/Unix and thus design their systems to use this. Because of that, it means that if you are operating on a Windows or Apple/Mac systems you will have to do additional setups in order to properly develop your visualizations on your system. If you are an Apple/Mac user however, this document can’t give you any absolute answers on it but might help guide you. More shall be discussed in the “Custom Visualization Creation - Splunk” section of this document.

In addition, if you are on Windows there are almost no IDEs that can truly integrate with Splunk Enterprise due to the IDE’s needing node.js, several JS react items, and node-packet-managers (npms) and the GitHubs that provide the Splunk Enterprise IDE are Linux/Unix based. Because of the structures and needs of developing for Splunk Enterprise which uses a lot of JavaScript, you will likely need to either develop your code for it something like Visual Studio, Visual Studio Code, or even Notepad++. Anything that can handle JavaScript, JSON, HTML, and CSS will be helpful for creating custom visualizations as that is what Splunk Enterprise relies on for custom visualizations. Also, if you plan on using anything in relation to D3 – it is suggested that you practice developing the D3 code separately from the Splunk Enterprise Visualizations until you are familiar with D3 operations and interactions because of the length of time and steps needed to compile code for adjustments in Splunk Enterprise is lengthy.

**Note**: at the time of writing the D3 code that is used in Splunk Enterprise is that of D3 version 3.5 and most tools that help learn D3 are in that of D3 version 4 or higher. More on this will be in the D3 and Splunk Interactions section of this document.

## Custom Visualization Creation - Splunk:

### Custom Visualization Tutorial & OS Setup:

Splunk Enterprise has a lot of documentation items to help create custom visualizations but there are some things that are extremely unclear, have issues due to being out of date, are not updated, or are limited to certain versions.

To start off with you will likely want to start with Splunk’s Custom Visualization tutorial, which includes a test visualization to create. The link to the Custom Visualization tutorial is found here: <https://docs.splunk.com/Documentation/Splunk/9.1.1/AdvancedDev/CustomVizTutorial>

This visualization is the basis of the Threat View visualization application, and it is recommended that you attempt it yourself to familiarize yourself with both Splunk Enterprise and Custom Visualizations. In addition, attempting this small visualization on your own can allow you a better understanding of the file structure that the visualization lives in and how it interacts with the other parts of the application.

However, the following must be performed in order for it to function properly:

1. Setup the web.conf file.

The web.conf file must be set! This allows for the Splunk Enterprise application to actually contact the Splunk files on your system. Without this your code could compile and run, but it will not show up in the Splunk Enterprise web-window. To create the web.conf file, create a file in something like Notepad++ and label it as “web.conf”. The web.conf file should have the following in it:

[settings]

minify\_js = False

minify\_css = False

js\_no\_cache = True

cacheEntriesLimit = 0

cacheBytesLimit = 0

enableWebDebug = True

Save this file in the “local” folder reached by this path – assuming you have downloaded Splunk Enterprise in its default download location otherwise replace the <…> with the file path you selected for Splunk Enterprise: <C:\Program Files>\Splunk\etc\system\local\web.conf

1. Setup the SPLUNK\_HOME variable if you are not on a Linux/Unix system.

If you run Splunk on a system that is NOT Linux/Unix (i.e Windows or Mac), it is essential that you set up or modify the variable of SPLUNK\_HOME. This variable is a system variable that helps to run the Splunk applications by pointing to where Splunk has been installed since it has the potential to be different for every user. However, the tutorial and basic system of Splunk Enterprise is usually meant for Linux users, and thus will not work if you are on a non-Linux/Unix system.

For Windows’ systems the SPLUNK\_HOME variable is called by %SPLUNK\_HOME% and this variable needs to be assigned to the Splunk installation folder area. In order to accomplish this, you will need to do the following:

1. Open up a command prompt window.
2. Type the command “echo %SPLUNK\_HOME%” and run it. If “%SPLUNK\_HOME%” is the result you need to set the variable.
3. To set the variable type the command “setx SPLUNK\_HOME [file path to the Splunk folder]”. If the file path to the Splunk folder contains spaces, you will need to encapsulate those spaces by adding additional backslashes to activate a quote deactivate a quote in order to encapsulate the string properly. For example, by using the default file path for Splunk the setx command would be as follows: setx SPLUNK\_HOME “C:\\\”Program Files\”\\Splunk”
4. In order to check that the SPLUNK\_HOME variable was set properly, open up a new instance of command prompt (a new window) and repeat the “echo %SPLUNK\_HOME%” command. This should appear with a file path such as: “C:\”Program Files”\\Splunk”. This means that the spaces have been properly preserved and can now function in Splunk with a little bit more modification.

In addition to the SPLUNK\_HOME variable you also need to go into the actual files of your visualization and find the “package.json” file under the filepath: >Splunk > etc > apps > [appname] > appserver > static > visualizations > [visualization name], and change some of the scripts within it to allow your system to process the JSON objects.

For the case of Windows, the package.json scripts should look like the following:

“

"scripts": {

"build": "%SPLUNK\_HOME%/bin/splunk cmd node ./node\_modules/webpack/bin/webpack.js",

"devbuild": "%SPLUNK\_HOME%/bin/splunk cmd node ./node\_modules/webpack/bin/webpack.js --progress",

"watch": "%SPLUNK\_HOME%/bin/splunk cmd node ./node\_modules/webpack/bin/webpack.js -d --watch --progress"

}

“

This file runs the scripts to compile/construct the visualization and if it isn’t pointing in the correct file path with the SPLUNK\_HOME variable, you will receive an “npm run build” error, where the script will fail to find the Splunk\_Home file path and thus not find the visualizations. With the %SPLUNK\_HOME% variable set you need to change the package.json file to have the above code to properly point to where the visualization is otherwise you will not be able to compile the visualization. Make sure to properly escape spaces and potential slashes if needed.

**Note:** This has already been done in Threat View’s application and once the program is compiled and built – the operating system it is running on does not matter. It only matters if you are constructing and building the application.

**Note:** If you are on a Mac OS you will need to do these steps as well, but it is unknown to us how to do such as the team developed this on Windows OS systems.

### Inputting Data into Splunk Enterprise:

Following the Custom Visualization Tutorial, you will likely need or want to input data into Splunk Enterprise in order for you to test items or create custom visualizations. To upload static tables of data you will need to do the following:

Login to the Splunk Enterprise application.

Click on the settings in the upper right corner, from the main menu that appears when you login.

Click on the “lookups” option underneath the “Knowledge” section on the left-hand side of the menu.

Click on the “add new” from the “Lookup table files” section which is on the top of the list.

From there follow the prompts to input in your data sheet. Remember what you name the data as that is what will be called in a query to the data.

In order to make sure that your application and you can query the data you input, you will need to set permissions. The manual way to do this is the following:

Click on the settings in the upper right corner, from the main menu that appears when you login.

Click on the “lookups” option underneath the “Knowledge” section on the left-hand side of the menu.

Click on the “Lookup table files” item which is on the top of the list.

This brings you to your Lookup table directory.

Look for the file paths of the data you have inputted and want permissions to.

Click the “Permissions” option in the row of the data you want to modify the permissions for.

This will bring you to the permissions of that data file.

In the prompt:

* Click the radio button next to “All Apps (system)” which makes it visible everywhere on your Splunk Enterprise.
* Under the “Read” label select the check box for “Everyone” – gives all users the ability to read the data.
* Under the "Write” label select the check box for "Admin” – gives permission to modify data to the admin-level workers.

Click “save” to save the permissions of that data file.

### Basic Query & Visualization Navigation:

Since Splunk Enterprise is basically a database, users of Splunk Enterprise can query data and have that query populate visualizations. However, it requires you to run a given query. The easiest query to run is the “Input Lookup” query which can be done as follows:

From the main menu (which can be accessed from the “Splunk>enterprise” logo in the top-right corner, click on the “Search & Reporting” button on the left-hand side. Otherwise go to the “apps” section next to the “Splunk>enterprise” logo on the top-right side and click the “Search & Reporting” app from the drop-down menu.

This will bring you to the Search page.

Click on the search bar and type the following:

|input lookup <filename>.<extension\_type>

Then either hit enter on your keyboard or the magnifying glass at the end of the search bar to start the query.

When the query is completed, it will give a list of what was found in that query. You can then select the “visualization” button underneath the search bar to see the data in various visualizations – the default visualization is a bar-chart. Custom visualizations will appear at the bottom of the visualization menu.

If the visualization allows it the “Formatter” button next to the “visualization” button will allow you to change parts of the visualization such as colors or axis-flipping. This is visualization dependent.

### Splunk documentation for advanced features:

To get more specific information on Splunk Enterprise custom visualizations we recommend that you check out these sources to further your knowledge and see if their functionalities will help you.

## D3 Functionalities and Interactions with Splunk:

D3 is a visualization tool that aids in the creation of visuals primarily for HTML though some JavaScript code that can be accessed from an HTML document. However, in Splunk Enterprise, D3 can be accessed so long as you have the libraries available – usually by installing the D3 module which comes with all of the libraries. This enables a Splunk Enterprise visualization to access the tools associated with the D3 visuals, and it can be accessed by putting commands for the D3 as part of the required items in the visualization\_source.js as stated in the Custom Visualization Tutorial provided by Splunk.

However, it should be noted that at the time of writing (April 2024) the most common tutorials for D3 are in D3 version 4 or higher as D3 is currently at version 6. But Splunk only has the copyright – and thus the support – of D3 version 3.5. This means that functionalities that are applied in higher versions are not in custom visualizations for Splunk. Take this into consideration when learning D3 items. It is recommended that you go to this resource page in order to find more information on D3 version 3 items: <https://devdocs.io/d3~3/>

Most of D3 works off of placeholders known as divs and appending items to those divs, such as text, shape, color and other such items. Due to the variety of D3 items that are possible there will only be a brief overview of the specific ones used in Threat View’s application in the next section: “Threat View Specific Items”.

## Threat View Specific Items:

This is a short run down on how Threat View specifically works as it is not like any of the base visualizations that come with Splunk Enterprise’s Custom Tutorial and its construction is drastically different from the Custom Tutorial Visualization as well.

First off, the Threat View application is made out of a gigantic if-else statement that is reliant on some of the defaults established in the UpdateView() function that ties to how Splunk Enterprise can receive user input from formatters to influence the visualization. To quickly find your way around the massive visualization\_source.js file you can control+f (using the find feature) to find “--- XXXX ---” as that is a comment to denote where in the start areas of the two visualizations.

The default values above the visualization if-else block are the primary items used in both views which include: card preview heights & widths, hex-code colors for the tactics they are assigned to, the indexing of fields to prevent data errors, and other containing items. The colors for the tactics and the variable labeled “viewTime\_TF” are particularly important because they are actually values that can be changed by the user of Splunk Enterprise. This is accomplished by having the default values for the variables be both in the visualization\_source.js and the “savedsearches.conf” in the default section of the Threat\_Timeline file structure. These two also have to have a correlation with the “savedsearches.conf.spec” that contains the expected/needed data types in the READ\_ME section of Threat\_Timeline file structure. In addition, if you want to edit the formatter or have the user interact with these default variables, you will need to adjust the “formatter.html” of the Threat\_Timeline application – be warned that when editing the formatter a Full Restart (information in “Debugging Tools for Splunk” section) is required for every edit to be shown.

D3 is used heavily throughout the Threat View application and some explanation is needed. In both views the data is pre-processed for its given purpose (either sorting by time or sorting by tactic) before being put into the D3 processing. The first part of the D3 processing creates the container for the view, then the chart is appended to the container but is treated as its own object which allows it to then get color and margin definitions attached. These two create the background of the visualizations in both views. Following that is the creation of the x-axis and this is view dependent, but this creates the x-axis bar and how large the sections in the bar need to be or what is on the bar. Following the x-axis creation, it is then placed onto the background and then gets its labels affixed to it. Though the methods are different in each view due to constraints, the idea is the same – labeling the sections for the user. Following this in both views there is the attaching of the card previews though they are labeled as “bars”. After the bars are added to the views, their additions come in such as text, color, format, and other such items.

Now probably the most important is the data that you need to input in order for the application to function. The data must have the following fields with the proper labels, in the given constraints or the application will not function:

* Titles – has a column labeled as “title” which contain a custom string data label from the detection system and are labels for the cyber-threat that was detected.
* Tactic – has a column label of “tactic” which contain the tactic name associated with the cyber-threat that is in accordance with the MITRE ATT&CK database for cyber-threats. This must be in the form of lowercase-split-by-dashes, uppercase-split-by-spaces, or Uppercase-split-by-dashes in order to function.
* Technique – has a column label of “technique” and contains the technique name of the cyber-threat in accordance with MITRE ATT&CK.
* Technique ID – has a column label of “technique\_id” and contains the technique ID of the cyber-threat in accordance with MITRE ATT&CK.
* Timestamp – has a column label of “\_time” and contains the time the attack took place in the form of yyyy-mm-dd hh:mm:ss.sss+00:00
* Description – has a column label of “description” and contains a description of the attack in accordance with MITRE ATT&CK.

Example of one (1) data item:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **tactic** | **technique** | **technique\_id** | **\_time** | **description** | **title** |
| defense-evasion | Steganography | T1027.003 | 2024-01-30T08:22:07.322+00:00 | Adversaries may use steganography techniques in order to prevent the detection of hidden information. Steganographic techniques can be used to hide data in digital media such as images, audio tracks, video clips, or text files.  [Duqu](https://attack.mitre.org/software/S0038) was an early example of malware that used steganography. It encrypted the gathered information from a victim's system and hid it within an image before exfiltrating the image to a C2 server.(Citation: Wikipedia Duqu)   By the end of 2017, a threat group usedâ€¯<code>Invoke-PSImage</code>â€¯to hide [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands in an image file (.png) and execute the code on a victim's system. In this particular case the [PowerShell](https://attack.mitre.org/techniques/T1059/001) code downloaded another obfuscated script to gather intelligence from the victim's machine and communicate it back to the adversary.(Citation: McAfee Malicious Doc Targets Pyeongchang Olympics) | None |

## Debugging Tools for Splunk:

Splunk Enterprise is hard to debug due to needing to see it in the web application and it being disconnected from the source code for the application. In order for you to access the application’s source code Splunk Enterprise needs to have a built version of the application to access, and that is what runs on the webpage – which is why the “npm run build” command is so important. The “npm run build” command is what allows you to compile and build the new version of the application for Splunk Enterprise to access.

But because it runs on a webpage and doesn’t directly interact with your source code, it is highly suggested that you use the web browser’s “Inspect tool” while on the visualization in the Splunk Enterprise webpage in order to see if the proper items are showing up. In addition, the inspect tool has a console that you can see, and by applying “console.log(“<sentences>”, <variables>); ” in the source code can aid in debugging.

However, occasionally Splunk Enterprise will visibly stall or not apply changes of the source code to the webpage. There are several reasons for this, but it is mostly because the Splunk Enterprise application does not register that there is a change because it is referencing the information stored within your web browser’s cache, the cache that Splunk stores, or doesn’t see there is updated information.

To fix this issue, here are some tips and tools for debugging:

Make sure you have run the “npm run build” in the given structure of your application to make sure that the changes are applied and compiled. If this does not fix the item or you have already done this follow the next steps.

1. **Basic restarts/refresh**

For basic restarts/refresh – for when it isn’t applying the information of small changes:

* Select the “Splunk>Enterprise” menu button in the top left to bring you back to the main menu.
* Select your app or the “Search & Reporting” button to force the Splunk application to contact your application.
* Apply your search, select the visualization.
* See if edits or console.log() items are applied.

1. **Debug Refresh**

Debug/Refresh Restart is for when the basic restarts aren’t applying anything at all.

* On your browser make sure you are on the Splunk webpage.
* Select it and either:
  + After the “8000/” type: “debug/refresh”

Or

* + Delete and replace the entire line with: http://127.0.0.1:8000/debug/refresh
* That way the <http://127.0.0.1:8000/debug/refresh> becomes the new URL.
* Follow the URL. This should lead to a page with a button labeled refresh.
* Click the button and wait for it to change to a screen with multiple lines and says “DONE” at the end.
* Either close that window or go back up to the URL and make it so it is <http://127.0.0.1:8000> to return to the homepage of Splunk Enterprise.

1. **Full Restarts internally from Splunk Enterprise.**

Full Splunk restarts are primarily when basic restarts have no effect, or you are dealing with formatter variables for Splunk Enterprise in which case this restart is needed after every change.

* Select the “settings” menu on the upper bar.
* Navigate to “server controls” and click it.
* Press “Restart Splunk” button.
* This will take some time and require you to re-log into Splunk.
* Select your app or the “Search & Reporting” button to force the Splunk application to contact your application.
* Apply your search, select the visualization.
* See if your edits or console.log() items are applied.

1. **Absolute Restarts from system.**

Absolute Restarts are when absolutely nothing is working from the above list and requires you to turn it off from your “services” area of your device.

On Window’s you do the following.

* Go to your search bar for your computer system and type “Services”.
* Click on the gear icon labeled “Services”. This brings you to your services window.
* Scroll down till you find the service labeled “Splunkd Service”.
* Select “Splunkd Service” right click it and select the “Stop” option.
* Allow it to process the command which can take a minute.
* Once it is not running – select “Splunkd Service” right click it and select the “Start” option.
* Allow it to process and the Absolute restart is finished.

There are similar ways to do the above instructions on different OS’s but the exact methods are currently unknown.