Network Security Visualization Using Virtual Reality

This work offers a method of IT security visualization presented through the use of a virtual reality program created using the Unity engine. This program has been developed for displaying network security big data in virtual reality using the HTC Vive. Using motion controllers for interaction a user is able to interact with this data in a virtual environment. This program is intended for use as an exploratory data analysis tool for network security specialists for both low and high level investigations. It can be used as a high level investigative tool by examining overall trends in the data to then find matches to specific results. It can also be used at a low level by being able to examine each and every object to see the details in the data.

There are seven major steps for the processing of data within the program as can be seen in Figure 1. The program was designed to be as flexible as possible and to achieve this the system can work with many types of IT security data including packet captures (pcaps), snort logs or any other logging such as firewalls, routers, and any other IDS systems. This data can be fed into the program as long as it is structured in a CSV file as seen in step 1. After the file has been placed in the proper directory within the Unity project the application can start. This begins the second step which uses a C# script to read the data into the program. Every record in the file is represented as an object and is stored within an array that contains each individual object. Each object contains all of the values for that record. Step 3 begins when the Unity engine reads the array and represents each object within as a physical block in the virtual world. Step 4 is when the user begins interacting with the program. Here the user selects the parameters they want for the query using the user interface described in the previous section. The dropdown box options can be auto generated if the records were labeled or can be entered manually. Step 5 is another C# script that takes all of the search data the user entered and uses it to search the array for all matches. Each object in the array is checked against every active search entry. Those objects that match every entry are added to another array that contains all the matching results. Step 6 is reads the array of matching objects and changes the colour of each of them to the colour specified by the user. At this point the user can make another search beginning the process again at step 4. This sequence can be repeated an unlimited number of times as indicated with the green arrows. Step 7 is the final step in which the user may export selected records to be used by other programs. The data can be sent out the same way it was entered as a CSV file. This feature allows the program to be used in conjunction with other tools.

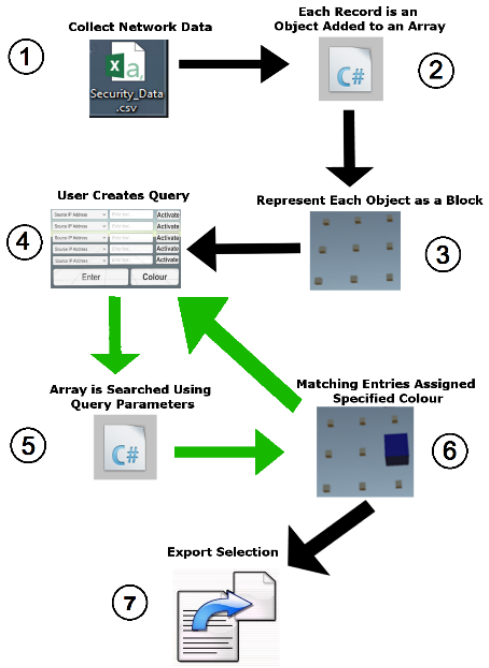

Figure 1: Data Flow

An example use case is now provided. Blocks surround the user that can contain any type of security data in this example each object represents a network flow. A user inputs their search parameters into the program using motion controllers that interact with a menu system. The resulting colours from the previous searches still remain and each search is

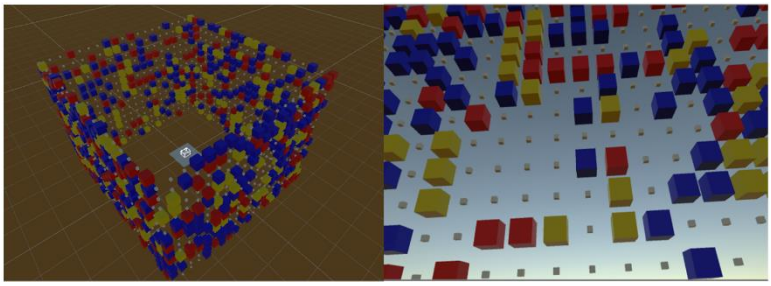

Figure 2: Layered Search

done on top of the previous one to create layers. Searches can be represented in stages to gradually narrow in on the most interesting traffic. The user can start searching for very broad characteristics of the traffic and then start specifying in greater detail the data they want to examine. This can be done starting with a simple query and then adding to it and running the modified search, this time using a different colour for the result. Each time the user makes a change to the search it will be layered on top of the previous one. By adding extra parameters to the search only some blocks of the previous search will match again. This means that a subset of the previously coloured blocks will change to the colour chosen for the new search. In this way the past search results are shown alongside the current results showing a progression of the search from most general to the most specific. This can be seen in Figure 2 with the left side showing a third person perspective and the right side showing a first person perspective positioned in the center of the arrangement.

Once the analyst has completed their queries they can remove all other entries they are not interested in. For example they can choose to keep any subset of the data they want such as the first query (blue, red and yellow), the second query (red and yellow) or just the most specific results (yellow). Being able to see all the versions of the query they can get a better sense of which data they want to keep. This gives more options for deciding which direction to take the data exploration. If the remaining dataset is still too large they can narrow the results further or if the most recent query removed too much data they can roll back and try another search.

Now that the analyst has identified the target sessions they can more closely examine the individual sessions by selecting each individual block and reading the additional details. The use of two controllers means that two entries can be directly compared at once. Black text means this data was part of a previous search, green indicates matching values and red shows values that do not match. This can be seen in Figure 3. A non VR program would involve scrolling, panning and clicking on each individual element one at a time, repeatedly clicking back and forth between two points they want to compare. Two controllers at once provides a quick and easy way to spot both similarities and differences within the set. Even when not comparing values, having two points of interaction can allow the analyst to search through the data with improved speed.

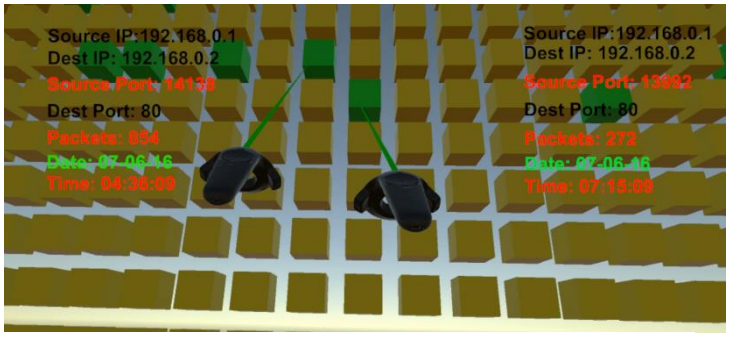

Figure 3: Comparisons

This progressing layering of queries acts as a visualized search history. The analyst can see a history of their searches as they get closer to the specific entries they are looking for. By being able to see the changes as the query evolved they can get a better sense of how their investigation proceeded. Visualizing the narrowing of the results can help analysts reflect on their approach which may help get faster results in a future investigation. This visualized search history can aid in the progression of threat escalation. The layered colours act as a mechanism for recording the stages of the investigation. The particular patterns that are of interest are seen through the changing proportion of the colours. Each stage of the investigation is tracked so when an analyst is working on a hypothesis the progression of changing colours aids to visualize the transitioning between the various stages. Another benefit of this approach is that the user is able to see both information that matches and does not match their query. Also provided are all the blocks of the base layer to compare against. With spreadsheet or command line analysis, once you filter out data it is no longer visible with the rest of the data. There is not enough room to display the old data alongside the new results. To see this data a user would have to revert back to before the search. With this technique all of the data is visible while working with the pieces you are interested in. If it is decided that you would no longer want the old data and want to load in more this can easily be done. This is all possible due to the increased workspace available in VR.