IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 17, 2020, accepted January 29, 2020, date of publication February 11, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973230

Owleyes: A Visual Analytics System for Functions and Connection Patterns of IPv4 Addresses in Networks

YAN YAN[1,2], LINGJUN HE[3], LI LIU[1,2], TAO YANG[1,2], WENHUA HOU[1], HONG XIANG[1,2], XIAOFENG XIA[1,2], AND HAIBO HU[1,2]

[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 400044, China
[2]School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China
[3]Department of Visualization Platform, Beijing Qianxin Technology Company, Ltd., Beijing 100068, China

Corresponding authors: Xiaofeng Xia (xiaxiaofeng@cqu.edu.cn) and Haibo Hu (haibo.hu@cqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1836114, and in part by the Chongqing Research Program of Basic Science and Frontier Technology under Grant cstc2017jcyjB0305.

ABSTRACT Netflow log files commonly contain massive transfer records in tiny time interval, making analytical works complex and burdensome. By combining human cognition abilities with computerized techniques, visual analytics systems have become efficient tools for showing network states and locating abnormal behaviors. However, traditional visual analytics systems tend to be designed for solving certain problems and unable to synthesize various types of data sources. Despite recent advances in network security visualization, academia still starves for a proper solution to visualize IPv4 address behavior modes and IPv4 connection patterns within limited drawing space. Thus, we propose a visual analytics system called 'Owleyes' which reprocesses Netflow log data with simple statistical operations in basic dimensions and fulfills the aforementioned requirements with proper novel graphs such as 'sunburst-hive-plot graph' (SHG) and link-wheel graph (LW). The SHG provides a stable and comparable means of visualizing connection patterns efficiently in a limited drawing space. The LW represents the hourly connection counts of main ports in a specific IPv4 connection during one day. With the use case dealing with the ChinaVis 2016 Challenge I data, the efficiency and practicability of Owleyes are demonstrated.

INDEX TERMS Visual analytics, network security, sunburst-hiveplot graph, link wheel graph, user-centric interaction.

I. INTRODUCTION

In recent decades, network technologies have been applied in many fields because of the development of data science and the increasing areas of its usage. Meanwhile, large amounts of malicious attacks appeared in network applications and place considerable demands on network security analytics systems [1], [2]. Netflow log data, such as logs from Intrusion Detection System (IDS) [3] recording time-dependent transmission details, has become an increasingly important data source for network security supervision and protection.

Extracting basic dimensions to get integrated results is an ideal way of reprocessing original Netflow log data for further analytical works. In most cases, the original Netflow log data contain problems. Firstly, to avoid the disclosure of sensitive information, some kinds of Netflow log data may be desensitized before being put into use for security analysis, which means certain information about file content will be erased. Secondly, Netflow log data from different sources may have plenty of dimensions and are in different formats, leaving the integration operation necessary. Thirdly, Netflow log data from specific sources such as firewall logs or other network security software logs may be difficult to integrate for their overcomplicated format or oversized volume. Finally, data redundancy and storage wasting may also occur. Extracting basic dimensions such as IPv4 address and port numbers from original Netflow log data can solve these problems by integrating network related information from multiple sources in uniformed format and low data complexity, making further analytical works easier.

The associate editor coordinating the review of this manuscript and approving it for publication was Walter Didimo.

The basic dimensions of NetFlow log data are determined by the aspects of one file transmission behavior, including the frame involved in the transmission process and the content information. The basic framework of a transmission process is contained in the network layer (OSI model), where the IPv4 address, port number, and transmission time describe the main connection process. In the application layer (OSI model), the content information of the file itself may include file size, file format, header file, basic content, etc. These contents have different focuses, but only the file size can cover the content, form, and characteristics of a specific file, which is concise enough and is also included in most Netflow log data content. Therefore, the transmission time, source IPv4 address, destination IPv4 address, source port number, and destination port number recorded in the network layer, and the transmission file size recorded in the application layer can be combined as the basic dimensions for integrating multi-source data.

However, even with the basic dimensions extracted, the increasing volume of data has placed a massive cognitive burden on analysts [4].

Visual analytics systems combine data analytical works with interactive visualization approaches, allowing analysts to comprehend network situations hidden in a large volume of data, which guarantees the ability to analyze the processed data with basic dimensions efficiently.

IPv4 nodes are the basic constituent elements of the network layer which can be regarded as the skeleton of the network. The distribution characteristics of IPv4 addresses can be visualized clearly by representing the hierarchical structure of their segments. Among the many visualization schemes, the sunburst graph can represent the hierarchical relationships and statistical values of the elements in Netflow log data within a limited drawing space. Moreover, compared to the basic tree graph and bubble tree graph, the sunburst graph is more flexible and can be combined with other graphs easily for its radial layout.

It is equally important to describe the characteristics of IPv4 connections [5], which can be abstracted as a subset of the entire network and involve the field of network visualization. In the academic of recent decades, connections among nodes in a network were tended to be visualized via force-directed graph [6] or parallel coordinate graph [7], which either have complex structure, poor stability, or limited visualization abilities and may cause visual deviations or require additional expertise and experience to understand the internal information. By contrast, the hive plot graph has the advantages of using simple drawing algorithms, high space utilization, intuitive, easy to understand, flexible, and easy to be combined with other radial graphs. Meanwhile, the hive plot graph builds a stable and comparable coordinate system that can help users visually compare different connection patterns and distinguish different IPv4 connection modes clearly and intuitively. Hence, we decide to use the hive plot graph as the main method of visualizing IPv4 connections.

TABLE 1. Services of specific port numbers.

Ports	service	Ports	Service
445	CIFS	443	HTTPS
389	LDAP	88	Kerberos
135	RPC	20	FTP
21	FTP Transfer	25	SMTP
139	CIFS	1025	Remote access
80	HTTP	53	DNS

By combining the hive plot graph and sunburst graph with appropriate interactions, we created a new novel radial graph [3] named Sunburst-Hiveplot Graph (SHG) to achieve a macro representation of the hierarchical relationships of IPv4 segments and the structure of IPv4 connections at the same time.

Next, the detail information of certain IPv4 addresses and IPv4 connections need to be visualized. First of all, the information of a certain IPv4 address includes data statistics results such as the count of connections, port usage details, and file transfer records in a specific period. The functions of the IPv4 address can be summarized from different port usages (see Table.1). Thus we use stream graph, boxplot graph, and scatter graph to visualize the detail statistical values of connection counts and port usages one IPv4 addresses related to. Then, a force-directed graph and a novel graph called Link Wheel graph (LW) are used to visualize the information of a certain IPv4 connection which include connection counts and port usage to help users analyze connection patterns in the network.

At last, we combine SHG with other graphs using proper interactions in user-centric design principles [8]. By integrating IPv4 address behavior modes and IPv4 connection patterns, users can determine the functions of IPv4 addresses in the network, summarize their overall running modes, and locate exceptional behavior through comparisons.

In summary, the main contributions of this paper include:

- We propose the SHG graph. As a combination of sunburst graph and hiveplot graph, SHG is capable of representing the IPv4 address behavior modes and IPv4 connection patterns over different periods and different filtering conditions.
- We propose the LW graph, which is a new designed graph representing the hourly connection counts of main ports in a specific IPv4 connection during one day.
- The interaction design process is under User-centric design principle. Reasonable interactions are added to the graphs and the entire Owleyes system to generate complete analytical processes.
- The system supports efficient anomaly detections. The system can accurately locate anomalies in the network by enabling users summarizing behavior patterns of entities and support intuitive comparison among visualized results.

The remainder of this paper is organized as follows. Section 2 contains an introduction to the design process and the visual analytic system. Section 3 list the related works. In Section 4 we describe the details of the design principles of graphs and interactions in the whole system. In Section 5, we verify the availability of Owleyes by performing use case studies with data presented in Chinavis'2016 challenge I. Section 6 contains a summary of our work. Section 7 highlights also some areas of deficiency and offering suggestions for further development.

II. RELATED WORKS

Numerous researches were referring to network security visual analytics in the literature, which introduced plenty of visual analytics systems to help analysts to understand network structure, detect exceptional behavior, and summarize typical patterns of cyberattacks. In general, visualization of the IPv4 address behavior modes and IPv4 connection patterns is the starting point of all analytic works for network security, which can help users identify possible exceptional behaviors by summarizing specific behavior patterns of entities in the network.

A. ANALYSIS OF NETWORK LOG FILES

Netflow log data record the file transfer process and communication history among entities in the network, which generate a time-oriented data format and is appropriate for time-series visualization methods. Wu *et al.* [9] satisfied this requirement by developing the Pianola system to visualize multivariate time-series security event data with low cognitive overhead. However, because of the tiny time interval when recording log files, the original Netflow log data are always cumbersome and redundant. Cappers et al. proposed an alert-oriented method known as Contextual analyzed network traffic alerts to help users analyze network traffic [10]. Yoo [11] proposed LongLine to enable visual analytics of large-scale audit logs.

B. IP HIERARCHICAL STRUCTURE VISUALIZATION

Visualization of the hierarchical structure of the IP addresses segments is important for visual analytics of the network. In academia, a hierarchical structure was generally represented by a basic tree graph, a bubble tree graph, or a sunburst graph. To visualize the hierarchical structure, Lin *et al.* [12] developed a discovery and visualization system for time series patterns called VizTree based on augmenting suffix trees, which can be used to transform the time series into a symbolic representation. Wang *et al.* [13] proposed a new type of tree structure called a diamond tree, which begins with a simple spatial layout using geometric theory, and is continually improved until a design that makes the best use of the screen estate is achieved. Mansmann *et al.* [14] superimposed a hierarchy on the IP address space, and considered the suitability of the treemap variants for each level of the hierarchy.

Although graphs based on the basic tree graphs were clear and intuitive, as a node-link diagram, they occupied plenty of display space and lead to poor visualization efficiency. Due to the need for clearly representing the connections between parent and child nodes, the number of leaf node elements could not generally exceed one-tenth of the pixel counts of the screen width.

The circle packing graph with nested circles was also an ideal method of representing hierarchical structure data [15]. Jochen et al encoded hierarchically structured data along with their uncertainties in a combined diagram called Bubble Treemaps [16].

Compared with the basic tree graph, the circle packing graph could be used to comprise more node elements in a limited drawing space, but there is still a limitation on the number of layers to be visualized; four layers are taken to be the limit regarding effects on user perception. Moreover, the circle packing graph has limited scalability and cannot be combined with other types of graphs harmoniously.

With the Self-Adapting Sunburst Algorithm (SASA) [17], the sunburst graph dynamically determined the width of every sector elements by their attribute value and had a compact radial format that uses a space-filling presentation method to show explicitly the proportion of the total value represented by each node [18]. Liu and Wang [19] adopted the Necklace Sunburst algorithm to optimize the overall arrangement via dynamic expansion and contraction of related nodes. In comparison with other hierarchical data visualization methods, the sunburst had high utilization of space and good scalability, was easy to add interaction or to be combined with other graphics, and was suitable for visualizing the hierarchical relationship of IP addresses.

C. NETWORK CONNECTION VISUALIZATION

The force-directed model was widely used in massive visualization projects related to social networks, biological networks, 3D modeling, citation networks for the simplicity of its algorithm, and its high stability and versatility. With the increase in the amount of researches on large-scale complex networks, the use of force-directed models to visualize large-scale node graphs have become a key point in most relevant fields [20].

In the visualization result of [21], several obvious central nodes divide the hosts in the network into different areas, in which is more likely to form centers due to unusual network scanning behaviors. Dzwinel *et al.* [22] proposed a new and fast graph-drawing method called ivga to support visual analysis for complex networks consisting of |V| 106+ vertices. Böhringer and Paulisch [23] added user constraints when generating the final layout, which can generate a more aesthetic rendering result and increase the usefulness of the current layout. Wang *et al.* [24] presented an improved stress majorization method that incorporates various constraints, including directional constraints, without the necessity of solving an optimization problem. For network simplification, Guozheng [25] proposed an attribute-based edge bundling algorithm that displays similar edges in nearby locations, achieving better clustering effects by grouping similar edges together.

However, the rendering result heavily depends on the selection of algorithms. The different algorithms would lead to different layouts, even under the same situation. Moreover, these algorithms tend to be sensitive to small changes in network structure, which could lead to large differences in the final rendering result. This incompetence in generating a robust visual layout meant that a force-directed graph is not ideal for comparing different network connections patterns.

Relatively, parallel coordinates could visualize multidimensional data efficiently. Yao et al. presented a new method to add a time dimension, which can extend parallel coordinates into 3D space [26]. In [27], Hanssan introduced a study evaluating the usability of 2D and 3D parallel coordinates for pattern identification in temporal multivariate data. However, limitations on drawing could have large impacts on the number of dimensions visualized in parallel coordinates, and large areas of overlap among line elements might occur when too much data needs to be visualized. Meanwhile, the intersections of oblique lines meant that visual errors can be introduced in parallel coordinate plots. In general, the disadvantages included rendering results with unfixed structure, too much occlusion between elements and unsuitable for limited drawing space, all of which could be conquered by introducing a visually coordinate system.

A stable coordinate system was necessary to help arrange the syntagmatic relationships among the graphical elements in the rendered layout, which could be satisfied perfectly in a hive plot, as introduced in 2011 by Krzywinski et al. for network structure visualization [28]. The use of an appropriate coordinate system could keep the layouts of network elements consistent if they had similar structural profiles and can be tuned to reveal specific patterns. This consistency made the hive plot layout appropriate for comparing network differences, monitoring their evolution, and analyzing changes in the connectivity of individuals in a network over time. In [29], Engle et al. applied hive plots to message-passing communication networks. The efficient drawing space utilization meant a hive plot can generate a comparable layout in a limited drawing space, and it was possible to combine it with other types of graphs thanks to its circular format.

However, in a real network, IPv4 addresses were divided into different groups with different functions. Directly using the original hive plot graph to represent complex connections was inadvisable because of its inability to represent different connections between and within different IPv4 groups, all of which needed further research.

D. VISUALIZATION ON BEHAVIOR PATTERNS

A security attack contained a sequence of related events, which could be a pre-planned set of simple incidents. According to the respective characteristics and presentation capabilities of different graphs, Yelizarov and Gamayunov [30] combined histograms, glyphs, scatter plots, color maps and parallel coordinate axes into a single visualization system to demonstrate the multifaceted nature of one cyber-attack. Different attacks [31] may show different patterns in visual systems due to their different characteristics. Chen *et al.* [32] proposed a connection streamgraph that contains six axes representing different attributes in the packet data, and Lee *et al.* [33] designed a visual signature view to show the attack pattern. For a firewall log, Ghoniem *et al.* [34] proposed a visualization system called VAFLE to help analysts to interpret firewall log events, which allows multiple coordinated interactive visualizations. Chen *et al.* [32] proposed the OCEANS system of visual analytics to provide close cooperation among analysts and greater situational awareness, and Simsek [35] proposed a lattice-based visualization method to capture the correlation between malicious hosts within the Intranet of an enterprise.

In summary, whether implemented by visually or other analytical methods, proper definition and comparison in behavior patterns are of paramount importance [36].

III. VISUAL DESIGN

In this section, we discuss the design principles of the Owleyes system including the visual graphs and interactions. The visual methods comprise SHG, LW, a force-directed graph, a stream graph, a boxplot graph and two scatter graphs. The interactions are under the user-centric principle.

A. SUNBURST HIVEPLOT GRAPH

The SHG model consists of a sunburst graph, a modified hive plot graph, and proper interactions. The sunburst part represents the hierarchical relationships and link count size comparisons among IPv4 addresses, while the hive plot part represents the IPv4 connections in the network. Each pair of arc elements in the sunburst part corresponds to a curve element in the hive plot part.

1) THE SUNBURST PART

IPv4 addresses are defined as 32-bit numbers and compose the basic structure of the entire network by identifying uniquely a host interface on the Internet with four levels of segments, these segments represents different functions. The distribution of IPv4 addresses can be represented clearly by visualizing the hierarchical structure of their segments. Different IPv4 addresses containing the same segment part may have related functions. As shown in Fig. 1, in an IPv4 address, segments in different parts work differently in the network. And they together form a hierarchical structure with four levels, where each level from top to bottom corresponds to each byte of the IP address from left to right.

As explained above, we chose the sunburst graph to visualize the hierarchical relationships as well as comparisons of connection counts among IPv4 addresses. As Fig. 1 shows, the sunburst model is composed of arc elements corresponding to specific IPv4 segments with three attributes: hierarchical level, inner angle and filling color. The hierarchical level of one arc element is determined by the level of its corresponding IPv4 segment. The internal angle size of the arc element is determined by the connection counts of the IPv4 segment. Larger connection count corresponds to the

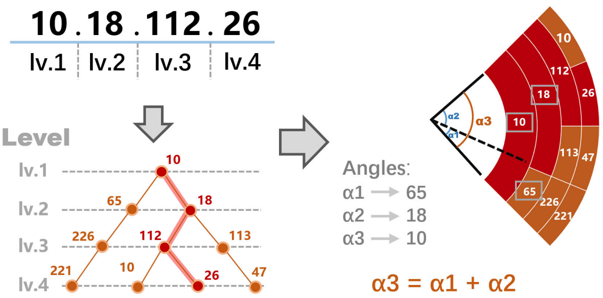


FIGURE 1. The four-layer structure of IPv4 address is constructed to generate the sunburst, where an IPv4 address is represented by a path from the root node to a leaf node. Each arc element in the sunburst contains three properties, namely its angle, its filling color, and its radial position. The angle corresponds to the numerical value of the IPv4 segment count. The filling color corresponds to the value range where the IPv4 segment connection count was in. The radial position corresponds to the level of the IPv4 segment. The red part of the figure corresponds to the highlighted part of the sunburst graph in the system after clicking the corresponding arc, which indicates the currently selected IPv4 address segment.

greater internal angle of the IPv4 segment. In the hierarchical structure, the connection count value of the parent IPv4 segment is the sum of its child IPv4 segments. Meanwhile, in order to represent the distributions of IPv4 segments more intuitively, their connection counts are divided into six range areas corresponding to different filling colors.

In addition, we added appropriate interactions to the sunburst part. Firstly, When the mouse moves over any arc element, a pop-up box would display its IPv4 address and connection count. Secondly, Users can click any arc element in the sunburst, then the auxiliary part on the top left will update the corresponding IP address segment and all its parent segments. At the same time, all related arcs will also be highlighted in the sunburst graph. As Fig. 1 shows, the red part of the figure represents the highlighted part of the sunburst graph after clicking the corresponding arc, which indicated the currently selected IPv4 address segment. On this basis, the sunburst part could be used as a basic interaction to help users choose specific IP addresses of interest for further analysis.

2) THE HIVEPLOT PART

According to the theory of hive plots explained in [28], the model contains three groups of components including axes, nodes, and edges. The nodes work as the skeleton of the model and their allocations are determined by certain rules. In detail, the attributes of axes such as position, scale, and orientation confirm a stable coordinate system. Then each node is placed onto a settled location of one certain axis element, which means one node only corresponds to one position in the final graph. For the edges representing connections between entities in the network, they can be simply visualized by curves connecting those nodes.

As shown in Fig. 2, the original hive plot model contains three radial axes, with different classes of nodes distributed on each. Therefore, it is necessary to determine the classification scheme of node elements and map them to a fixed axis. Different data sources and requirements have different classification schemes, but the uniqueness of node classification shall be ensured, that is, a node can only belong to one group. After determining the rules mapping each node to a certain axis, we need to determine their positions on the axes. We set connection count as sort rule, which means nodes with larger connection count will be placed on the outside of the axis. Because the value differences among node connection count are usually exponentially. By default, we take the pairwise value of connection count to determine the coordinates of nodes on the axis. Then we connect these nodes with Bezier curves to represent connections among entities these nodes corresponding to. To help the user build an intuitive understanding of the link count of each curve element, the period of the link count is divided into six intervals, corresponding to six different filling colors respectively.

Although the current design represents the curves between nodes on different axes well, the curves between nodes on the same axis may cross cover each other, resulting in cluttered views. To solve this problem, as shown in Fig. 3, each axis

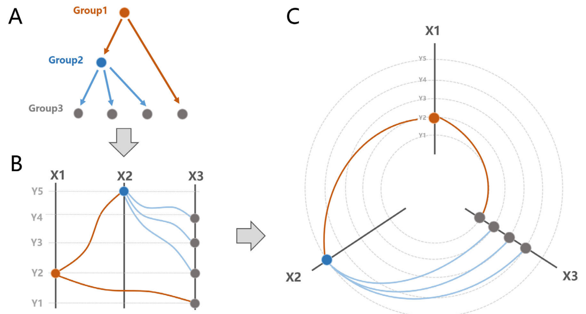


FIGURE 2. In the evolution process of hiveplot model, firstly, different groups of nodes in a network structure are divided into separated axes, as shown in part A and part B. Then the distribution of these axes is changed into radial, as shown in part C. The certain attributes of axes (position, scale, and orientation) confirm a stable coordinate system. Each node is positioned onto a settled location of one certain axis in one-to-one correspondence. The stable coordinate system can efficiently and comparatively show the connection relationships between nodes in a limited circular drawing area.

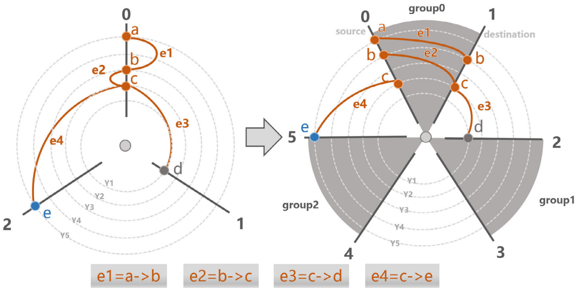
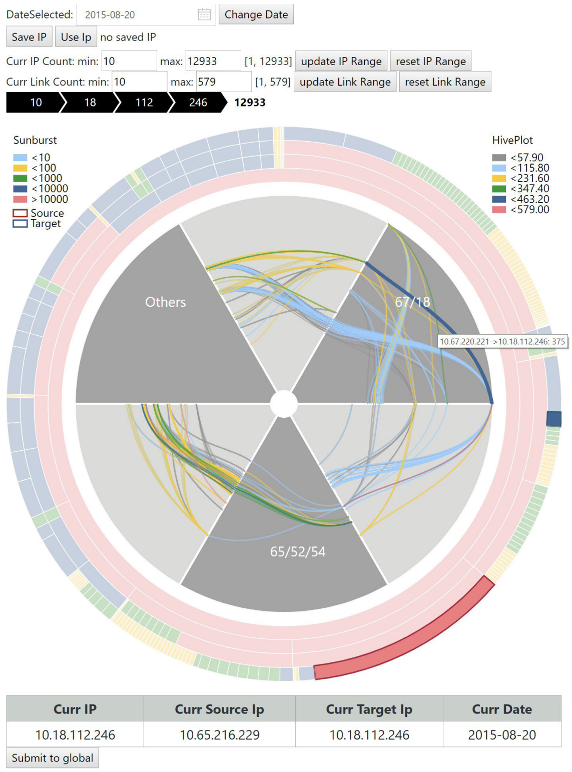


FIGURE 3. The curves between nodes on the same axis may cover over each other, so we split each axis into two axes in clockwise direction, where the first axis contains source IPv4 nodes and the other one contains target IPv4 nodes. The curves between the nodes on different axes is unchanged. For example, in connection e1, the source IPv4 address is node a and the destination IPv4 address is node b, which are all on axis 0. In the newly generated graph, the source IPv4 address node of e1 is on axis 0 and the destination IPv4 address of e1 is on the axis 1.



DateSelected: 2015-08-20

no saved IP

Curr IP Count: min: 10 max: 12933 [1, 12933]

Curr Link Count: min: 10 max: 579 [1, 579]

Curr IP	Curr Source Ip	Curr Target Ip	Curr Date
10.18.112.246	10.65.216.229	10.18.112.246	2015-08-20

FIGURE 4. We assemble the SHG by combining the sunburst graph and the hive plot graph together with appropriate interactions. The SHG can handle analytical works all alone for its complete interaction loop, which includes directions from sunburst part to hive plot part and the reverse. Besides, the SHG contains two filter interactions, the first is in the sunburst part for IPv4 address count and the second is in the hive plot part for IPv4 connection count. These filter functions make SHG suitable for various scales of Netflow data.

is copied into two in clockwise direction, where the fore axis holds the source IPv4 address nodes and the back axis holds the destination IPv4 address nodes. The newly generated fan area between the two axes is used to place the curve elements. Until now, the connection between all node elements in the hive graph can be clearly and intuitively observed.

Then Appropriate interactions are added into hive plot. When users moving the mouse above a curve, the width of the curve will be thicker. Meanwhile, a pop-up box will appear near the mouse to display the specific information of the connection this curve element corresponding to, including the source IPv4 address, destination IPv4 address, and the connection count. The hive plot part not only represents the connections between IPv4 addresses but also serves as an interactive tool to help users select specific connections between two IPv4 addresses.

3) COMBINATION AND INTERACTION

Next, we combine the hive plot graph and the sunburst graph with appropriate interactions to generate a new graph named SHG, as shown in Fig. 4. The sunburst graph is used as a tool representing the basic situation of each IPv4 address. After users click the arc element in the sunburst graph, the corresponding IPv4 segment is selected. Meanwhile, in the hive plot graph, the curve unrelated to this IPv4 segment will be hidden, and the filling color of the remaining curves will be reassigned. Correspondingly, when the user clicks one curve element in the hive plot graph, the source IPv4 address and destination IPv4 address of this connection will be selected, and the corresponding arc elements in the sunburst graph will be surrounded by red boxes and blue boxes respectively. So far, a complete interactive loop has been constructed between the sunburst graph and the hive plot graph.

Then we add date selecting interaction to SHG. In the beginning, we set the time interval of selection precisely to ensure flexibilities, where the time unit includes specific dates, hours, and minutes. While the too flexible interaction can not bring the desired efficiency, it does bring too much burden to the users because of the overmuch choices. At the same time, as a result of the superabundant time intervals, there is no constraint between the rendered hive plot graphs or sunburst graphs, which makes it difficult to compare each pair of them. Therefore, in the end, we decide to reduce the flexibility of time selection to date selection, so that users can only choose to visualize the data of a certain date.

However, the overlarge volume of the source data still causes plenty of overlaps in the hive plot regardless of the time interval size. Hence, filtering interactions for the IPv4 address counts or for the IPv4 link counts are necessary for users to get their desired rendering results. We determine the priority between the two filtering operations by their sequence, which means that the latter operation has a higher priority than the former. For example, suppose the first filtering rule is that the IPv4 address count shall be within [0, 1000], the second filtering rule is that the IPv4 link count value shall be within [0, 100]. For example, we assume there exits one IPv4 connection that had link count value 80 and a source IPv4 address with count value 2000. Although the source IPv4 address failed to meet the condition of being within [1, 1000], according to our priority strategy, it was still rendered on the sunburst graph.

B. LINK GRAPHS

So far, the SHG has macroscopically represented the information and connections of IPv4 addresses. The next step is to visualize the behavior details of each specific port during IPv4 connections. A novel graph named Link Wheel is designed to show the statistics of the ports with prominent behaviors and the differences between these port usages in different periods. Besides, we choose force-directed graph to indicate the connection relationships between these ports.

1) LINK WHEEL GRAPH

The Link Wheel (LW) graph records the hourly connection counts of main ports in a specific IPv4 connection during one day. As shown in Fig. 5, in LW, time flows in clockwise direction, as the arrows in the outmost circle show. The current month and date are placed in the graphic center, while the two IPv4 addresses corresponding to the connection are placed in

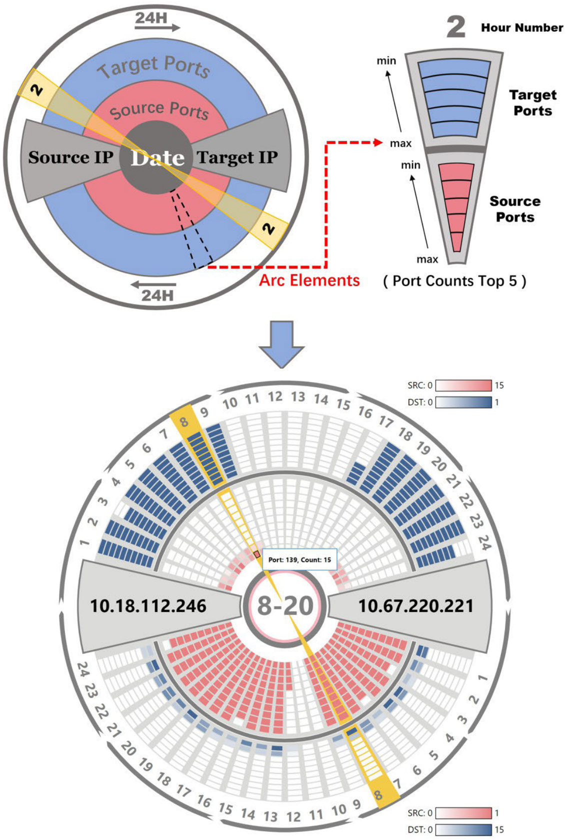


FIGURE 5. The Link Wheel (LW) graph records the hourly port connection counts of a specific IPv4 connection in inverted order, which corresponds to direction inside to outside in the final graph. The passage of time is clockwise. The top part and the bottom part represented different connecting directions between the two IPv4 addresses. The red part represents the source ports, the blue part represents the target ports. The yellow arcs represent the interactions used to compare statistics in both directions within the same hour.

the horizontal middle. The other graphic elements are divided into two groups, indicating different connection directions. Each part contains one arc inside in red and one arc outside in blue. The arcs filled up in red show the statistical results of the source ports, while the blue ones show the statistical results of the destination ports. Each arc is divided into 24 sub-arcs, corresponding to 24 hours in one day, as the numbers outside the arc represent. Each sub-arc contains 5 in-arc elements, indicating the top five ports with the largest count value, which are sorted ascending in radial. The color transparency of an in-arc element was proportional to the count value of its corresponding port. The default maximum value in the legend was 1000, which will dynamically change according to the maximum number of all port count values in the data.

To display port usage information more clearly and efficiently, appropriate interactions are added to LW. Firstly, the inputs of the LW include the date, the source IPv4 address, and the destination IPv4 address, all of which are from the SHG submission results (details about the submission interaction are in the global interactions part). Second, to help users compare the differences between the two directions in the same hour, we add a comparison interaction: when the user moves the mouse to an in-arc element, the background color of the entire hour arc in both directions would be filled in yellow, as shown in the figure. Finally, we add a pop-up box to display the relevant information including the port number and the count of the port corresponding to each in-arc element.

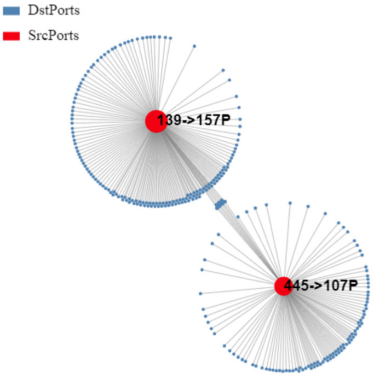


FIGURE 6. In force-directed graph, all the nodes are divided into two groups based on their related IPv4 addresses. The ports in the source group are filled in red, while the ones in the target group are filled in blue. The radius size of a port node corresponds to its connection count value. Meanwhile, the port number and the counts of ports connected to this port is represented beside the circle.

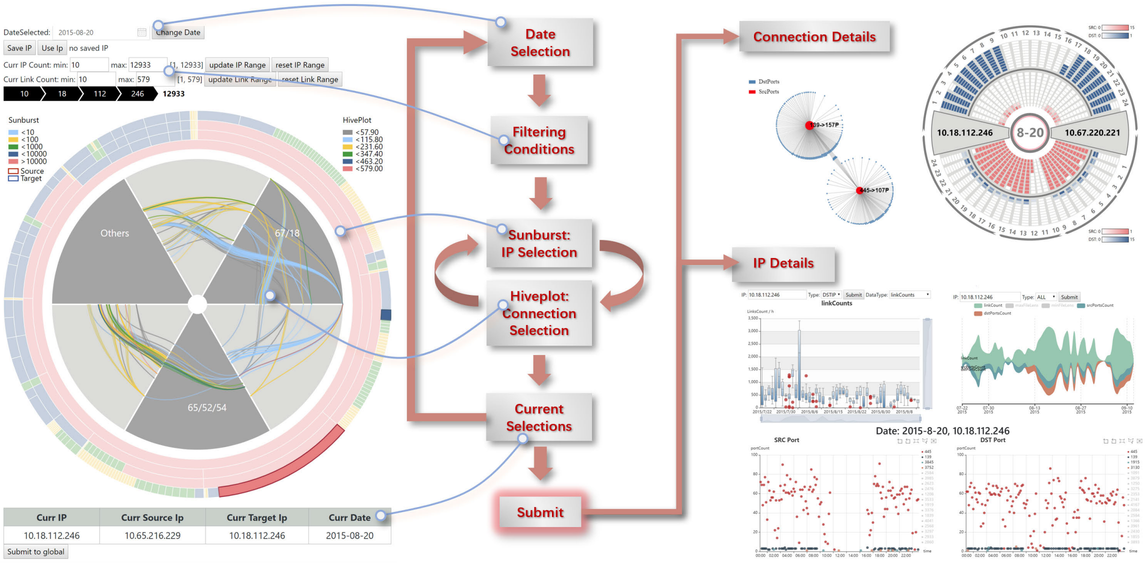
Based on the characteristics of LW, the IPv4 connections with different functions will generate different rendering results, while connections with the same function would lead to similar rendering shapes. Users can summarize specific behavior pattern characteristics from multiple layouts to analyze various modes of IPv4 connections and find anomalies.

2) FORCE-DIRECTED GRAPH

Because LW only focuses on visualizing the usage details of each port in the IPv4 connections, as a supplement, a force-directed graph is added to show the connection relationships among the related port. In the graph, nodes represent ports and edges represent connections, as shown in Fig. 6. All the port nodes are divided into two groups based on their related IPv4 addresses. The source port nodes are filled in red, while the target ones are filled in blue. The radius size of a port node corresponds to its connection count value. The port number and port counts connected to this port are located beside the circle.

New interactions are added to the force-directed layout, which allows users to reverse the source and target roles of the IP addresses in order to compare the connection relationships between them. Users can also hide or show the port numbers around the nodes in the graph to analyze port usage more accurately.

As for limitations on graphic rendering of the force-directed graph, the number of nodes can not exceed 4000, which is enough for our system. Our force-directed graph uses d3.js built-in algorithm, whose performance is limited by the front-end browser calculating performance and canvas



Curr IP	Curr Source Ip	Curr Target Ip	Curr Date
10.18.112.246	10.65.216.229	10.18.112.246	2015-08-20

FIGURE 7. The SHG forms a complete interactive loop, which means users can use SHG to obtain basic analytical results independently. A commit operation exists between SHG and other graphs. After the user determines current selections in SHG, including the date, the current IPv4 address, and the current IPv4 connection, and submit them to global, the other graphs will update accordingly.

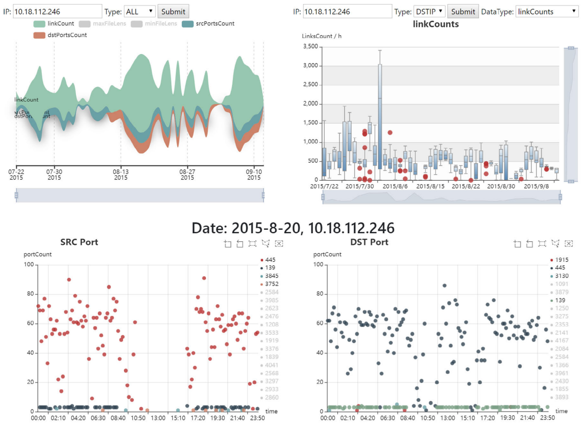


FIGURE 8. Certain IPv4 address graphs.

rendering performance. During our tests, the force-directed graph refreshes normally when the number of nodes is less than 1000. Although different degrees of stutter may occur when more than 1000 nodes are rendered on the drawing space, the final rendering result will not be effected much.

C. IP GRAPHS

As shown in Fig. 8, we divide the IPv4 address information graphs into two groups: global group (GP) and specific date group (SP).

The GP contains a stream graph and boxplot graph, representing the overall statistical results of an IPv4 address in the whole time period of the data. The stream graph represents basic statistics of this IPv4 address, including the sum of daily connections, maximum transfer file size, minimum transfer file size, the sum of source ports counts, and the sum of destination ports counts. The user can choose different statistical results of this IPv4 address based on its roles in connections. The roles include the source IPv4 address, the destination IPv4 address, or both. The boxplot graph uses hourly data statistics as the basic element to show the distribution of the quantities in the daily statistical results. Because boxplots can only display individual dimensions of the statistical result, we add interactions to help users choose which dimension the boxplots should display.

The SP contained two scatter graphs with the same structure, which are used to visualize the usage of the main source ports and destination ports of this IPv4 address during the selected date.

D. GLOBAL INTERACTIONS

As the first step for users to grasp the overall situation of the internal network, SHG helps users determine their interested IPv4 address and IPv4 connection, which would work as inputs to the other graphs. To the beginning, we simply place interactions of the other graphs after SHG. During our test processes, we find that the interaction frequency on SHG is much higher than the other graphs, which lead to the frequently refreshing of all graphs in the system if simply combining SHG with other graphs directly, resulting in reducing of the operating efficiency and the slowing down of the user analysis speed.

In consideration of SHG itself has formed a closed interaction loop, which means users can use SHG for analysis without other graphs, we isolate SHG from other graphs by adding a new submit process in the middle. As shown in Fig. 7, we add a table into SHG to represent currently selected date,

TABLE 2. The original data examples.

STARTTIME	SRCIP	DSTIP	SPORT	DPORT
2015/7/22 0:00:01	10.52.128.2	10.118 .163.108	1433	3799
2015/7/22 0:00:01	10.67.220.27	10.145 .216.221	2445	135
2015/7/22 0:00:08	10.52.140.75	10.118 .163.105	445	1534
...	...	...	...	...
2015/9/8 9:59:56	10.67.220.38	10.145 .216.221	2117	445
2015/9/8 9:59:56	10.145.216.221	10.67 .220.38	445	2117
2015/9/8 9:59:56	10.18.112.246	10.65 .216.101	445	1941
...	...	...	...	...

IPv4 address, and IPv4 connection. This process lets users select the IPv4 address or connection of interest in SHG first, which will dynamically update a table below the SHG. Then after determining their choice, users can submit these selections to global to update the other graphs.

IV. CASE STUDY

A. DATA DESCRIPTION

The data is from Chinavis 2016 Challenge I [37], containing 2 million records involving more than 20 thousand IPv4 addresses and 60 thousand ports. The recording starts from July 1st, 2015, and covers time periods of 2 months, where the minimum time interval is second. We consider only six dimensions of attributes in the data including the start time point, the source IPv4 address, the destination IPv4 address, the source port, the destination port, and the transferred file length to build the Owleyes system, as shown in Table. 2.

As mentioned above, the node elements in the hive plot graph are supposed to be divided into three groups, where grouping rule is determined by specific data condition. However, the data of Chinavis 2016 challenge I does not involve grouping rules and will not have great impact on the next analysis process. Hence, we decide to divide the IPv4 segment into different groups by the second segment of the IPv4 address. In detail, in consideration of the connection count distribution, we decide to group the IPv4 addresses with 67/18 as second segment value into the first group, the IPv4 addresses with 65/53/54 as second segment value into the second group, and the rest of addresses are divided into the third group.

B. CASE STUDY 1: ROLES AND FUNCTIONS OF IPV4 ADDRESSEE

We take the IPv4 address 10.18.112.246 as an example to introduce the complete visual analytical process. Because from the sunburst part of SHG, it is obvious that 10.18.112.246 has the largest connection count on most of the dates compared to all the other IPv4 addresses. We presume that 10.18.112.246 is representative and it may function as the IPv4 address of a major server in the network.

Firstly, we observe the overall performance of IPv4 address 10.18.112.246 from SHG, stream graph, and boxplot graph.

As shown in Fig. 9, the SHG shows that before August 10th, 10.18.112.246 only works as a destination IPv4 address. During these days, there are several fixed IPv4 addresses connected to it with large connection counts, including10.67.220.221, 10.67.220.229, 10.65.216.229, 10.65.216.146, and 10.65.216.221 (We preliminarily conclude that these IPv4 addresses also work for main servers in the network, which can be used as the main object of the next analysis). Correspondingly, the stream graph also shows that before August 10th, 10.18.112.246 only works as the destination IPv4. During this period, The number of active ports is less than 20, which was small compared to normal situations. From the corresponding scatter graph we can see that although there are few active destination ports, the connection counts of these ports are large. Therefore, we conclude that at this time, 10.18.112.246 is mainly working as IPv4 address of a server in the network.

After August 10th (on August 14, for example), as shown in the figure, the SHG structure of 10.18.112.246 changes significantly, where new group of curves appear and intersect the original curves, representing that the IPv4 address starts to work as source IPv4 address, which also has fixed IPv4 addresses including 10.67.220.221 and 10.67. 220.229. These IPv4 addresses also maintain a large number of connections. It is observed from the stream graph that ports working as source ports start to appear from this date, which consistent with the results in the SHG, indicating that the function of this IP address changes significantly after this date.

Further observation finds that in the stream graph, on August 11th, 12th, 13th, 26th, and September 1st, the number of destination ports sharply decreases, and there is no crossover of connection sets in the hive plot graph, too (for lack of space, in Fig. 9, we only show situations on September 1st). During these days, the functions related to the destination ports are closed. The specific reason is analyzed by the scatter graph later. When 10.18.112.246 works as destination IPv4 address, the hive plot graph rendering result is divided into two regions with the date August 3rd as the division point, indicating that its function may have changed at this time point.

Secondly, with the scatter graphs and LW graph, we summarize specific functions of 10.18.112.246 by analyzing the usages of special ports on 10.18.112.246.

As shown in the scatter graph, before August 3rd, the active destination ports of 10.18.112.246 include 445, 139, 1025, 389, 135, 53, 88, etc. According to Table.1, these ports cover protocols with specific functions such as CIFS, LDAP, RPC, DNS, Kerberos. We believe that during these dates, the server of 10.18.112.246 is responsible for many different types of functions. However, the opening of port 1025 leaves the server a safety hazard, which can be one of the reasons for the reallocation of the functions.

In addition, on August 3rd, the usage of each port increases sharply, especially port 445, which is an order of magnitude more than before. Then from August 4th to August 6th, all the ports are used sporadically. From August 7th to August 10th, only 445 and 139 remain working as the destination ports that are heavily used (for lack of space, in Fig. 9, we only show situations on August 9th). We conclude that 10.18.112.246 will

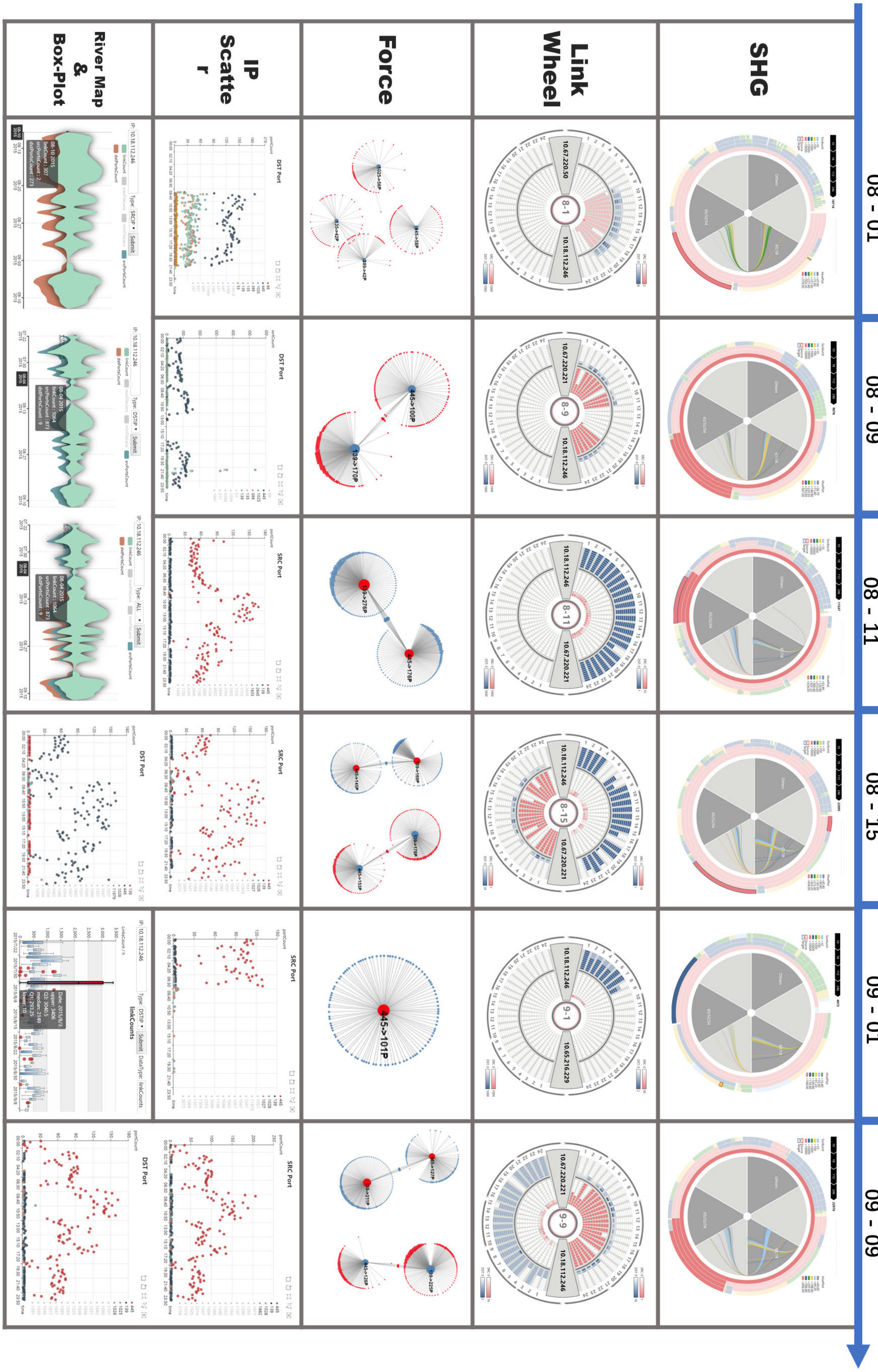


FIGURE 9. Case study 1: IP roles and functions.

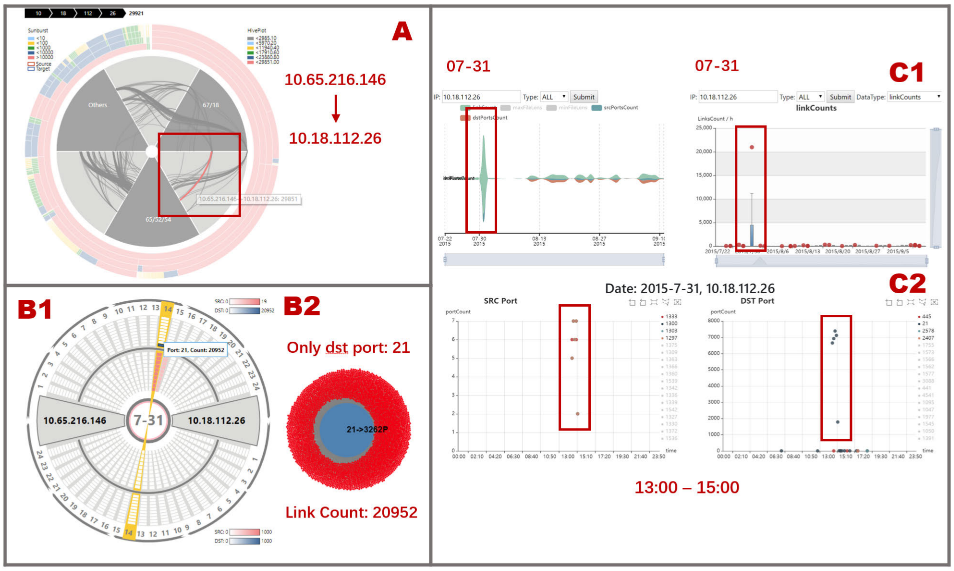


FIGURE 10. Case study 2: Abnormal events and behaviors.

no longer bear the functions related to other ports except 445 and 139. However, from August 11th to August 13th, all destination ports on 10.18.112.246 are closed. At the same time, 445 and 139 begin to work as source ports, which is in line with the conclusion on the hive plot graph mentioned above. Immediately from August 14th, these two ports begin to work both as source and destination ports, so it is inferred that from 11th to 13th, preparations are made for a new function of this IPv4 address, which is officially opened from August 14th. As shown in Fig. 9, the conclusions are also proved in the LW, where different situations lead to different rendering shapes.

Similarly, from September 1st to September 3rd, ports of 10.18.112.246 also only work as destination ports. But unlike the previous, no data is recorded in September 2nd. The amount of data on September 3rd is very small compared to the other dates, which is a incremental process. Hence, we conclude that the corresponding server of 10.18.112.246 is shut down for maintenance and backed to work on September 4th.

C. CASE STUDY 2: ABNORMAL EVENT (DDOS ATTACK)

As shown in Fig. 10, in SHG, we select the current date to July 31th and find that almost all the curves in the hive plot graph are in gray, which is abnormal from the colorful rendering results under normal conditions. A closer inspection reveals that only one curve was in red, indicating that the count of this connection is far larger than all the others. The filtering interactions showed that the maximum IPv4 connection count value was 29,851 that day, which was more than 20 times greater than normal cases. The connection details show that the connection is from 10.65.216.146 to 10.18.112.26. After ensuring the specific date and connection are selected, we submit them to the other graphs.

As Fig. 10.B2 shows, the force-directed graph indicates that 3262 ports of the source IPv4 address 10.65.216.146 are all connected to port 21 of the destination IPv4 address 10.18.112.26. In Fig. 10.B1 and Fig. 10.C2, the LW and scatter plots represent that almost all 10.65.216.146 ports were connected 18 times, which are happened from 1 p.m. to 3 p.m. In Fig. 10.C1, the stream graph, and boxplot graph show that this large count and concentrated connection only appears once in the entire time interval of the data set, which is obviously abnormal. Port 21 generally works as the entry point of a file server. Therefore, we determine that the IPv4 address 10.18.112.26 is the corresponding IPv4 address of the file server in the network. Above all, the connection indicates that the server of 10.65 .216.146 does a DDoS attack to the server of 10.18.112.26 on July 31, 2015.

V. CONCLUSION

In this work, we propose a visual analytics system named Owleyes and illustrate its usefulness through case studies using Netflow log data provided by ChinaVis 2016 Challenge I. In Owleyes, the SHG combines sunburst and hive plot graph to provide stable and comparable visual analytical methods for global analysis of IPv4 behavior modes and IPv4 connection patterns, which help users determine the entities they should focus on in the next step. For the details of IPv4 address behaviors, the stream graph, the boxplot graph, and the scatter graphs represent details of port usage, connection count and file size. For IPv4 connections, the Link Wheel (LW) graph recorded the hourly connection counts of main ports in a specific IPv4 connection during one day. And the force-directed graph represented connection relationships among different ports of two certain IPv4 addresses. By combining those graphs with interactions in the user-centric principle, Owleyes can help analysts investigate network events and find out abnormal events effectively.

VI. FUTURE WORK

In Owleyes, the detection of abnormal events mainly depends on the experience of users. The ability to help users analyze more types of abnormal events in complex networks needs to be improved. In future work, we plan to add more advanced visualization methods to ultimately help users obtain more insights and implicit knowledge from Netflow log data. Meanwhile, although aming at working on analysis tasks with multi source netflow log data, Owleyes is only suitable for data sources which have the basic six dimensions. For the data sources mismatch the condition, our system can not support a efficient analysis process, which is also need to be improved in our future work.

REFERENCES

[1] X. Li, Q. Wang, L. Yang, and X. Luo, "Network security situation awareness method based on visualization," in *Proc. 3rd Int. Conf. Multimedia Inf. Netw. Secur.*, Nov. 2011, pp. 411–415.

[2] S. Yoo, H. R. Ryu, H. Yeon, T. Kwon, and Y. Jang, "Visual analytics and visualization for android security risk," *J. Comput. Lang.*, vol. 53, pp. 9–21, Aug. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1045926X18300107

[3] Y. Shi, Y. Zhao, F. Zhou, R. Shi, Y. Zhang, and G. Wang, "A novel radial visualization of intrusion detection alerts," *IEEE Comput. Graph. Appl.*, vol. 38, no. 6, pp. 83–95, Nov. 2018.

[4] F. Zhou, W. Huang, Y. Zhao, Y. Shi, X. Liang, and X. Fan, "ENTVis: A visual analytic tool for entropy-based network traffic anomaly detection," *IEEE Comput. Grap. Appl.*, vol. 35, no. 6, pp. 42–50, Nov. 2015.

[5] Y. Yang, Q. Yao, and H. Qu, "VISTopic: A visual analytics system for making sense of large document collections using hierarchical topic modeling," *Vis. Inform.*, vol. 1, no. 1, pp. 40–47, Mar. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2468502X17300074

[6] T. Horeman, M. D. Blikkendaal, D. Feng, A. Van Dijke, F. Jansen, J. Dankelman, and J. J. Van Den Dobbelsteen, "Visual force feedback improves knot-tying security," *J. Surgical Edu.*, vol. 71, no. 1, pp. 133–141, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1931720413001852

[7] Y. Okada, "Network data visualization using parallel coordinates version of time-tunnel with 2Dto2D visualization for intrusion detection," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2013, pp. 1088–1093.

[8] Z. Zhong, Y. Zhao, R. Shi, Y. Sheng, J. Liu, H. Meng, and D. Lin, "A user-centered multi-space collaborative visual analysis for cyber security," *Chin. J. Electron.*, vol. 27, no. 5, pp. 910–919, Sep. 2018. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/cje.2017.09%.021

[9] C. Wu, S. Sheng, and X. Dong, "Research on visualization systems for DDoS attack detection," in *Proc. IEEE Int. Conf. Systems, Man, (SMC)*, Oct. 2018, pp. 2986–2991.

[10] B. C. M. Cappers and J. J. Van Wijk, "Understanding the context of network traffic alerts," in *Proc. IEEE Symp. Vis. Cyber Secur. (VizSec)*, Oct. 2016, pp. 1–8.

[11] S. Yoo, J. Jo, B. Kim, and J. Seo, "LongLine: Visual analytics system for large-scale audit logs," *Vis. Inform.*, vol. 2, no. 1, pp. 82–97, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2468502X18300159

[12] J. Lin, E. Keogh, and S. Lonardi, "Visualizing and discovering non-trivial patterns in large time series databases," *Inf. Vis.*, vol. 4, no. 2, pp. 61–82, Jun. 2005.

[13] P. Wang, K. Zhang, X. Song, T. Zhou, and D. Gong, "Geometry based visualization of hierarchical structures," in *Proc. IEEE/ACIS 13th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2014, pp. 465–470.

[14] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda, "Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1105–1112, Nov. 2007.

[15] W. Wang, H. Wang, G. Dai, and H. Wang, "Visualization of large hierarchical data by circle packing," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI)*. New York, NY, USA: ACM, 2006, pp. 517–520, doi: 10.1145/1124772.1124851.

[16] J. Görtler, C. Schulz, D. Weiskopf, and O. Deussen, "Bubble treemaps for uncertainty visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 719–728, Jan. 2018.

[17] G. Li-Wei, C. Yi, Z. Xin-Yue, and S. Yue-Hong, "A hierarchical data visualization algorithm: Self-adapting sunburst algorithm," in *Proc. Int. Conf. Virtual Reality Vis.*, Sep. 2013, pp. 185–190.

[18] K. Rodden, "Applying a sunburst visualization to summarize user navigation sequences," *IEEE Comput. Graph. Appl.*, vol. 34, no. 5, pp. 36–40, Sep. 2014.

[19] C. Liu and P. Wang, "A sunburst-based hierarchical information visualization method and its application in public opinion analysis," in *Proc. 8th Int. Conf. Biomed. Eng. Informat. (BMEI)*, Oct. 2015, pp. 832–836.

[20] Y. Zhao, F. Luo, M. Chen, Y. Wang, J. Xia, F. Zhou, Y. Wang, Y. Chen, and W. Chen, "Evaluating multi-dimensional visualizations for understanding fuzzy clusters," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 12–21, Jan. 2019.

[21] Q. Liao, L. Shi, and C. Wang, "Visual analysis of large-scale network anomalies," *IBM J. Res. Develop.*, vol. 57, no. 3/4, pp. 13:1–13:12, May/Jul. 2013.

[22] W. Dzwinel, R. Wciso, and W. Czech, "ivga: A fast force-directed method for interactive visualization of complex networks," *J. Comput. Sci.*, vol. 21, pp. 448–459, Jul. 2017. [Online]. Available: http://www:sciencedirect:com/science/article/pii/S1877750316301430

[23] K.-F. Böhringer and F. N. Paulisch, "Using constraints to achieve stability in automatic graph layout algorithms," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. Empowering People (CHI)*, 1990, pp. 43–51.

[24] Y. Wang, Y. Wang, Y. Sun, L. Zhu, K. Lu, C. W. Fu, M. Sedlmair, O. Deussen, and B. Chen, "Revisiting stress majorization as a unified framework for interactive constrained graph visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 489–499, Jan. 2018.

[25] Y. Guozheng, L. Yuliang, and C. Huixian, "A new network topology visualization algorithm," in *Proc. 1st Int. Conf. Instrum., Meas., Comput., Commun. Control*, Oct. 2011, pp. 369–372.

[26] Y. Zhonghua and W. Lingda, "3D-parallel coordinates: Visualization for time varying multidimensional data," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Aug. 2016, pp. 655–658.

[27] K. A. Hassan, N. Rönnberg, C. Forsell, M. Cooper, and J. Johansson, "A study on 2D and 3D parallel coordinates for pattern identification in temporal multivariate data," in *Proc. 23rd Int. Conf. Inf. Visualisation (IV)*, Jul. 2019, pp. 145–150.

[28] M. Krzywinski, I. Birol, S. J. Jones, and M. A. Marra, "Hive plots-rational approach to visualizing networks," *Briefings Bioinformatics*, vol. 13, no. 5, pp. 627–644, Sep. 2012, doi: 10.1093/bib/bbr069.

[29] S. Engle and S. Whalen, "Visualizing distributed memory computations with hive plots," in *Proc. 9th Int. Symp. Vis. Cyber Secur. (VizSec)*. New York, NY, USA: ACM, 2012, pp. 56–63, doi: 10:1145/2379690:2379698.

[30] A. Yelizarov and D. Gamayunov, "Visualization of complex attacks and state of attacked network," in *Proc. 6th Int. Workshop Vis. Cyber Secur.*, Oct. 2009, pp. 1–9.

[31] V. Bartos, M. Zadnik, S. M. Habib, and E. Vasilomanolakis, "Network entity characterization and attack prediction," *Future Gener. Comput. Syst.*, vol. 97, pp. 674–686, Aug. 2019. [Online]. Available: http://www:sciencedirect:com/science/article/pii/S0167739X18307799

[32] S. Chen, C. Guo, X. Yuan, F. Merkle, H. Schaefer, and T. Ertl, "Oceans: Online collaborative explorative analysis on network security," in *Proc. 11th Workshop Vis. Cyber Secur. (VizSec)*. New York, NY, USA: ACM, 2014, pp. 1–8, doi: 10.1145/2671491.2671493.

[33] C. P. Lee, J. Trost, N. Gibbs, R. Beyah, and J. A. Copeland, "Visual firewall: Real-time network security monitor," in *Proc. IEEE Workshop Visualizat. Comput. Secur. (VizSEC)*, Nov. 2005, pp. 129–136.

[34] M. Ghoniem, G. Shurkhovetskyy, A. Bahey, and B. Otjacques, "VAFLE: Visual analytics of firewall log events," in *Proc. Vis. Data Anal.*, Dec. 2013, Art. no. 901704, doi: 10.1117/12.2037790.

[35] S. Simsek, "Work in progress–tracking correlated attacks in enterprise intranets through lattices," in *Proc. Securecomm Workshops*, Aug. 2006, pp. 1–2.

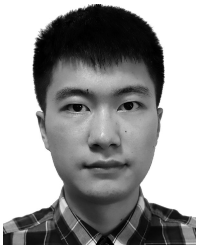
[36] Y. Zhao, X. Luo, X. Lin, H. Wang, X. Kui, F. Zhou, J. Wang, Y. Chen, and W. Chen, "Visual analytics for electromagnetic situation awareness in radio monitoring and management," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 590–600, Jan. 2020.

[37] Y. Zhao, Z. Zhang, and X. Yuan, "Chinavis data challenge from 2015 to 2017," *Chin. J. Netword Inf. Secur.*, vol. 4, no. 2, p. 55, 2018. [Online]. Available: http://www.infocomm-journal.com/cjnis/EN/abstract/article_166995.shtml

YAN YAN received the bachelor's degree in software engineering from Chongqing University, China, in 2016. He is currently pursuing the master's degree with the School of Big Data and Software Engineering, Chongqing University. His current research interests include software engineering, cyber security visualization, and visual analytics.

LINGJUN HE received the master's degree in software engineering from Chongqing University, China, in 2018. She is currently a Software Engineer with Beijing Qianxin Technology Company, Ltd., China. Her current research interests include data analysis and visualization.

LI LIU received the bachelor's degree in energy and power engineering from the North China University of Water Resources and Hydropower, China, in 2015. He is currently pursuing the master's degree with the School of Big Data and Software Engineering, Chongqing University. His research interests include information visualization, visual analysis, and network security.



TAO YANG received the B.Sc. degree in software engineering from Chongqing University, China, in 2019. He is currently pursuing the master's degree with the School of Big Data and Software Engineering, Chongqing University, where he is associated with data analysis and visualization. His current research interests include spatial-temporal big data visualization and view optimizing.

WENHUA HOU received the master's and Ph.D. degrees in management engineering from Chongqing University, China, in 2001 and 2012, respectively. She is currently a Research Associate with the Key Laboratory for Dependable Service Computing in Cyber Physical Society of MOE, China. Her current research interests include big data analytics and multiobjective optimization.

HONG XIANG received the B.Sc. degree in mathematics from Sichuan Normal University, China, in 1984, the M.Sc. degree in mathematics and computer science from Lakehead University, Canada, in 1992, and the Ph.D. degree in mathematics from the University of Alberta, Canada, in 1998. He is currently a Professor with the School of Big Data and Software Engineering, Chongqing University. His research interests include machine learning, information security, and cryptography.

XIAOFENG XIA received the master's degree from Chongqing University, China. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University. He is also a Research Associate with the Key Laboratory for Dependable Service Computing in Cyber Physical Society of MoE, China. His current research interests include cloud security, industrial control security, and cryptographic application.

HAIBO HU received the master's degree in software engineering and the Ph.D. degree in computer science from Chongqing University, China, in 2004 and 2012, respectively. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University. He is also a Research Associate with the Key Laboratory for Dependable Service Computing in Cyber Physical Society of the Ministry of Education, China. His current research interests include software engineering, pattern recognition, semantic web, and visual analytics.