



ARL-CR-0845 • MAR 2020



Vids Cyber Defense Visualization Project

prepared by Joshua Edwards and Gregory Shearer

ICF

9300 Lee Highway, Fairfax, VA

under contract W911QX-18-D-0002

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Vids Cyber Defense Visualization Project

prepared by Joshua Edwards and Gregory Shearer

ICF

9300 Lee Highway, Fairfax, VA

under contract W911QX-18-D-0002

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) March 2020		2. REPORT TYPE Contractor Report		3. DATES COVERED (From - To) 01 August 2016–01 January 2020	
4. TITLE AND SUBTITLE Vids Cyber Defense Visualization Project				5a. CONTRACT NUMBER W911QX-18-D-0002	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Joshua Edwards and Gregory Shearer				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) ICF 9300 Lee Highway, Fairfax, VA 22031				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) CCDC Army Research Laboratory ATTN: FCDD-RLC-ND 2800 Powder Mill Road Adelphi, MD 20783-1138				10. SPONSOR/MONITOR'S ACRONYM(S) CCDC ARL	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) ARL-CR-0845	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Cybersecurity poses a challenging terrain to visualize for the situational awareness of cyber defenders. Therefore, the Vids project (2016–2020) aimed to provide a 3-D platform for cybersecurity data visualization development. This report includes the background of the project, project status, project description, images of visuals produced by the project, and lessons learned from user and developer feedback during the project.					
15. SUBJECT TERMS cybersecurity, visualization, 3-D, virtual reality, augmented reality					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 30	19a. NAME OF RESPONSIBLE PERSON Gregory Shearer
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (301) 394-1578

Contents

List of Figures	v
List of Tables	v
Acknowledgments	vi
1. Introduction	1
2. Project Status	3
2.1 Project Schedule	3
2.2 Current Status	3
3. Related Work	4
3.1 Conceptual	4
3.2 Use Case Projection	4
3.3 Platform	5
3.4 Network Security Visualization and 3D Perception	6
4. Tool Description	6
4.1 Data Input	6
4.2 Data Views	8
4.3 Nodes	12
4.4 Edges	13
4.5 Interaction	13
5. User Feedback	14
6. Lessons Learned	15
7. Unexplored Areas	16
8. Conclusion	18

9. References	19
List of Symbols, Abbreviations, and Acronyms	21
Distribution List	22

List of Figures

Fig. 1	Appearance of the original VIDS software.....	2
Fig. 2	Example breakdown of the CSV data parsing methodology	7
Fig. 3	Upper left: force-directed layout of network flow data with selected links glowing. Upper right: force-directed and cubic layout of network flow data. Lower left: two force-directed network flow data graphs, showing highlighting on common attributes between graphs. Lower right: force-directed network flow data, circular graph network flow data, and 2-D data chart.	10
Fig. 4	Upper left: force-directed network flow graph with selected flow edges. Upper right: circular graph with selected edges. Lower left: two types of x-y charts; on left, x axis represents time, y axis represents number of bytes observed per datum, and the z (depth) axis represents multiple datum points occurring at the same time and byte count; on right, x axis represents the number of recurrences of a key within the data (for example, the number of times two IPs communicated with each other over a period of time), with the y axis representing a frequency tally of how many data points fall into each count range. Lower right: circular graph showing network flows.....	11
Fig. 5	Upper left: on left, data are grouped by country code on x axis, with a frequency tally for each data point falling into each category, creating a frequency histogram where each point represents a unique and interactive data point; on right, data are grouped by country code on a pie chart. Upper right: hierarchical data are organized into a multitiered layout. Lower left: higher-dimensionality server log data encoded and dimensionally reduced using t-distributed stochastic neighbor embedding into a 3-D embedding to highlight clusters of similar data (color coding). Lower right: same as lower left, rotated to a new perspective.	12

List of Tables

Table 1	Project schedule	3
---------	------------------------	---

Acknowledgments

The authors would like to graciously acknowledge the generous contributions of the following people to the project: Lee Trossbach, for outstanding project leadership and vision, Curtis Arnold for project support and oversight, and Kaur Kullman for excellent idea generation and feedback over the course of the project, as well as highly appreciated technical collaboration.

1. Introduction

Visualization is a key element of battlefield planning and awareness for the warfighter. Traditional visualization uses maps, organizational charts, and analytics to plan, rehearse, and execute operations as part of the intelligence preparation of the battlefield performed before, during, and after military operations. In cyberspace, the terrain of the battlefield is constantly changing. In place of traditional terrain such as rivers and mountains, networks and associations are key features. Given the different terrain, new visualizations and visualization methods are needed to facilitate cybersecurity situational understanding.

Visualization for cybersecurity awareness can be used by cybersecurity analysts and decision makers to assess trends and patterns in large volumes of network traffic information, potentially faster than any other form of information media, such as textual lists or spreadsheets. Network traffic and organization visualization thus can be a key tool to building an ability to understand a rapidly evolving and complex network environment. In the near future, emerging technologies including virtual reality (VR), augmented reality (AR), and mixed reality (MR) will likely become more widespread, and these technologies may allow for a fundamental shift in the way data are currently visualized. To explore this new visualization environment, new tools are needed to translate network data into 3-D visualizations from 2-D ones, enable interactivity with these visualizations, and integrate into an analyst's workflow.

This report describes the Vids project created by the US Army Combat Capabilities Development Command (CCDC) Army Research Laboratory (ARL) in conjunction with the CCDC Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance, and Reconnaissance (C5ISR) Center. This provides an interactive 3-D environment for visualizing network intrusion detection system data for cybersecurity analysts, building off prior publication (Zage and Zage 2010) and patent (Trossbach and Robinson 2015) work. Vids builds upon the legacy of an earlier visualization project called VIDS, a joint project between ARL and Ball State University. Originally named the Visual Intrusion Detection System (VIDS), the project focused solely on visualizing Intrusion Detection System (IDS) alerts in 3-D space. The IDS alerts were placed in a structured 3-D scatterplot for cybersecurity analysts to quickly view alerts, categorize alerts based on location in the graph, and find correlations between alerts (Fig. 1).

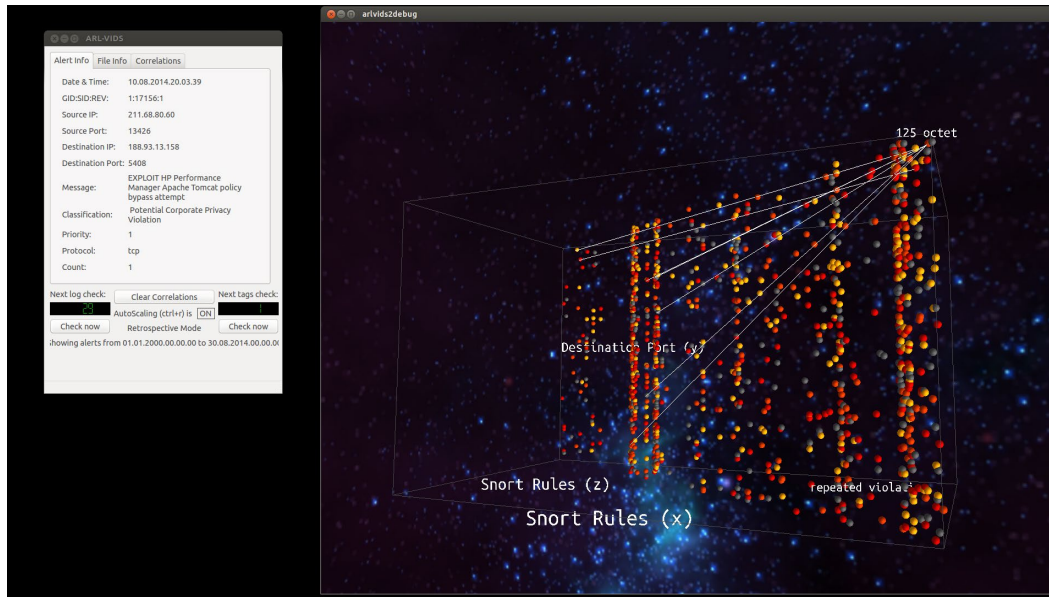


Fig. 1 Appearance of the original VIDS software

To render its 3-D visualizations, the original VIDS software used the Ogre 3-D engine for its visualizations and Qt for the user interface. This provided general cross-platform support and the necessities for 3-D graphics rendering. The project experimented with different rendering methods, examined performance limits of visualization, and investigated user preferences for elements such as backgrounds and illumination. The joint project ran for approximately 4 years before concluding in 2014. The final product contained all of the requested features but was hampered by stability issues, an overly rigid structure, and most of the original collaborative development team either graduating from the university or otherwise moving on to new projects. With most of the original VIDS team leaving, the decision was made to create an updated version of the project, developed internally at ARL and utilizing the lessons learned from VIDS. The new project name, Vids, no longer an acronym, expands beyond solely visualizing IDS alerts and gives cybersecurity analysts more freedom in deciding what data they will bring into the 3-D space for analysis and correlations.

VIDS, the predecessor of Vids, was presented to several cybersecurity analysts at ARL, and the feedback was largely positive. Analysts appreciated the scatterplot layout and the ease of understanding it provided. However, development of the previous iteration had reached its technical and organizational limit due to constraints of the original project. Restarting and incorporating technical advancements with Vids allowed the ideas of its predecessor to mature and expand to fill additional gaps. Concurrently, a CCDC ARL project, and later, C5ISR project named Virtual Reality Data Analysis Environment (VRDAE) pioneered using VR

and the Unity game engine to enable multiperson analysis and collaboration in a virtual environment. Vids functioned as a parallel project to VRDAE, exploring specific problems within the 3-D visualization research space. The intent was not to redevelop or duplicate work done on VRDAE; rather, Vids primarily aimed to create a non-VR-focused platform (i.e., designed for interaction with a mouse and keyboard and designed to be viewed on a traditional computer monitor or flat screen) for operationally useful data reading, extraction, and rendering in a 3-D environment. Vids is intended to be integrated into VRDAE at a future date (to be determined) whereby Vids operates both on non-VR-focused platforms and functions within the VRDAE as a supported 3-D analytical tool.

2. Project Status

2.1 Project Schedule

The original project schedule is outlined in Table 1.

Table 1 Project schedule

Date	Event
August 2016	Use case validation with analysts
June 2017	Concept review and progress demo
April 2018	First tech report: “Vids: Version 2.0 Alpha Visualization Engine”
May 2018	Overview and demo to ARL Technical Advisory Board (TAB) during poster session
June 2018	Hands-on demo and feedback session with CSSP Net Defense and USCYBERCOM reservist from 780th Military Intelligence Brigade
August 2018	Development milestone: “Alpha” version complete, “Beta” development started
May 2019	Demo presentation at Cybersecurity Services Provider Working Group (CSSPWG)
September 2019	Briefing at 15th Virtual Worlds Forum via Defense Intelligence Agency
September 2019	Briefing and reference in paper at NATO Modelling and Simulation Center of Excellence (M&S COE) Computer Aided Analysis, Exercise, Experimentation (CA2X2) Forum
October 2019	Briefing and demo for Federal Reserve Bank (FRB) National Incident Response Team (NIRT)

2.2 Current Status

As of May 2019, the CCDC ARL Vids development team transitioned active Vids development to the CCDC C5ISR VRDAE development team for further development, user feedback sessions, and possible integration within the VRDAE project. As of October 2019, the CCDC ARL Vids development team continues to assist development in a support role.

3. Related Work

3.1 Conceptual

Conceptually, the Vids project falls within a field known as immersive analytics. Marriott et al. (2018) define immersive analytics as “the use of engaging, embodied analysis tools to support data understanding and decision making”. They state that it is the goal of such analytics to remove barriers among people, their data, and the tools they use for analysis. Although Vids itself, as implemented at present, is intended to be a 2-D representation of a 3-D space (displayed on a 2-D monitor), (e.g., a desktop workstation) the purpose of the project as a whole was to provide a platform for further research and development on 3-D visualization for network analysis that would inform the construction of a fully immersive analysis environment. Therefore, in the long term, integration of Vids with a VR or AR environment was planned, and in the near term, Vids provided a platform for testing visualization concepts and gathering feedback from prospective users. Several works were influential in defining a long-term vision for Vids as described previously. Alberts et al. (2015) support the underlying need for visualization research to support network defense research. Payer and Trossbach’s (2015) envisioning of immersive analytics development to 2035 helped inspire some of the functionality goals for Vids features, such as the ability to select and interact with edges in a graph like ropes or strings. In a parallel and contributory capacity, work by Kullman et al. (2018, 2019a) was informative as to analyst cognitive considerations as well as virtual reality and mixed reality interaction (Kullman et al. 2019b). Another report by Shearer and Edwards (2018) on Vids was created at the conclusion of the alpha development phase and can be used as a point of comparison for this report.

3.2 Use Case Projection

The Vids project team envision the following use cases for cybersecurity analysts using the software. Some of these use cases were pioneered by the original VIDS project and remain relevant for the new Vids design. Such use cases are intended to elaborate usage considerations beyond the potential general benefits of 3-D visualization on large data sets discussed in the Conceptual section (3.1).

- Providing a visual aid structure for “known knowns” (i.e., providing a structure for shared knowledge). Such shared knowledge may include consistently monitored infrastructure, workflow models, or types of activity. This type of common visual model, once established, could then be leveraged to explain events or incidents based on a known visual

reference model. The reference model may be useful for training new analysts using a visual shorthand. Additionally, a shared visual reference model may allow for faster communication about a complex situation.

- Providing assistance for hunting “known unknowns” or “unknown unknowns” (i.e., allowing for greater correlative support across many dimensions). This type of in-depth support may help analysts to identify events by enabling a visual pattern searching methodology with cybersecurity-related data. Additionally, such multidimensional analysis may allow for data correlations that would normally be tedious to perform.
- Making complex analysis more accessible to less experienced analysts, particularly those who desire a way to visualize cybersecurity information. The investigation style of experienced cyber analysts is typically nonvisual, frequently involving extensive use of scripting, pattern matching, and command line tools. The learning curve for new analysts can therefore be significant, particularly for those perhaps more accustomed to polished and user-friendly interfaces. Therefore, visual analysis may function as a way of easing the learning curve, allowing for complex events to be explained in a more easily digested form. Indeed, such visualization may also be of benefit in explaining cyber events to non-analysts.
- Providing a flexible visualization option for multiple data types with adaptability to novel data types for both analysts, researchers, and developers can share a tool for visualization use and development. Such a tool may provide improved convenience and improved feedback as a collaboration of multiple involved parties at different levels of a research, developer, or user chain.

3.3 Platform

The CCDC ARL Vids team used the Unity Real-Time Development Platform for application development.* Unity is a widely used platform for game development, animation, and many other visualization applications (Peckman 2019). Unity is a cross-platform engine, with more than 20 supported platforms as of the Unity 5 major version release in 2015 (Robertson 2015). Additionally, Unity supports AR and VR. As described in the Introduction, the VRDAE project also used Unity for development. Other studies in 3-D perception used Unity as a research platform (Sicat et al. 2018). The choice of Unity as a development platform was therefore a logical choice based on suitability as well as prior use and availability.

* <https://unity.com/>

3.4 Network Security Visualization and 3D Perception

With the Vids project, the CCDC ARL Vids team aimed to address open questions in the logical layout of network features into a topology suitable and amenable to interactive 3-D visualization for operational intrusion detection monitoring and analysis. The question of which layouts are best suited to network security needs is an open one; at least two studies (Shiravi et al. 2011; Ferebee and Dasgupta 2008) suggest a variety of different layouts with no clearly superior layout for specific tasks such as network situational awareness. Other studies have addressed some of these questions in a general manner, including the measurement of user performance in 2-D versus 3-D perception tasks (Filho et al. 2018), and some studies examine different ways of rendering a plot and their effect on perception (Sarıkaya and Gleicher 2018). Vids is capable of organizing data into any available layout while allowing users to seamlessly transition between layout states. This feature was considered by the CCDC ARL Vids team to be key to help users remain cognitively immersed in the data analysis task while transitioning between data layouts, and as a key research feature to enable comparative research on algorithms and perception paradigms (i.e., 2-D vs. 3-D) in a common visualization environment.

4. Tool Description

At the time of transition, the development level for the Vids application was “early beta”, meaning the core features were in place with some further refinement required based on user feedback. The description of the Vids features and design to follow is representative of the state of Vids upon transition to CCDC C5ISR.

4.1 Data Input

The alpha version of Vids currently reads data only from comma-separated value (CSV) files. In the future, a JavaScript Object Notation (JSON) data-reading module will read data from JSON files in a similar way. A CSV file may originate from recorded data and be updated by local processes or the output of a database operation. To create a graph in Vids, the user specifies the location of a CSV file and a data configuration file. Vids then reads the CSV data and processes the raw data into nodes and edges according to the rules defined in the configuration file (Fig. 2).

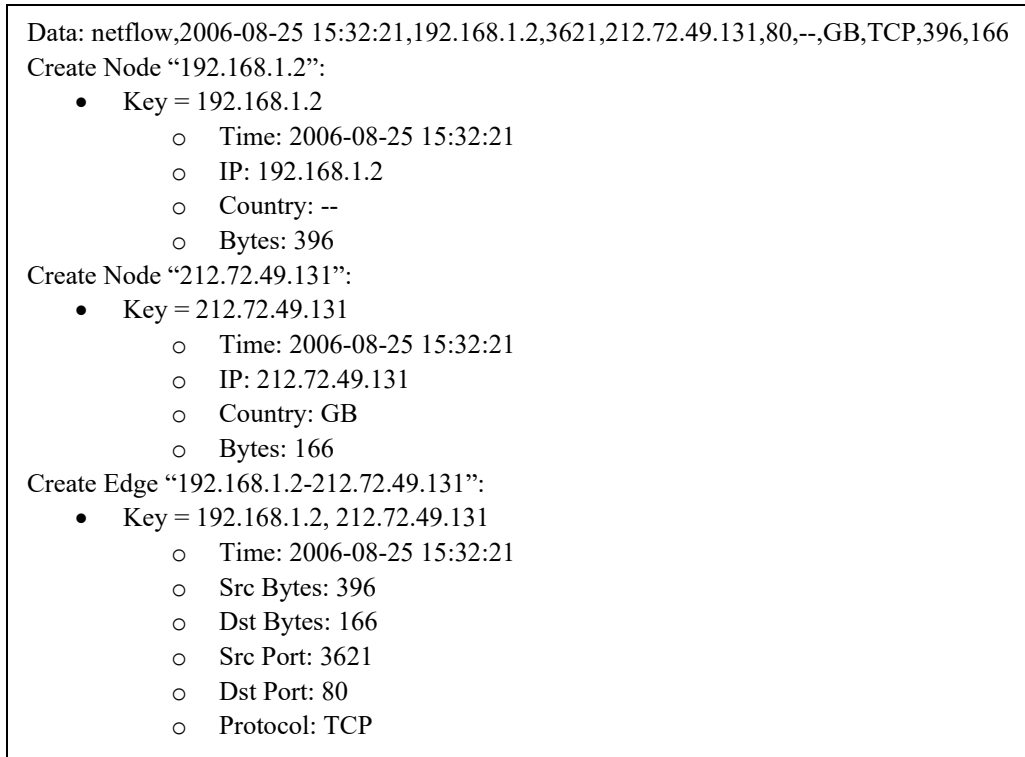


Fig. 2 Example breakdown of the CSV data parsing methodology

In essence, the data for both nodes and edges are handled in the same way in terms of configuration. The data configuration defines a unique key field or set of key fields for each type of node and edge to create. In addition, each node or edge type defines a set of attribute fields, specified by CSV column index(es), to collect each time the unique key appears in the CSV file. These attribute fields may include string, numeric, IP address, and/or date/time data.

The data processing process does not necessarily create a new node for every line of input. A single line of data, in this case a netflow record produced using the tool DShell,* can be used to create several node and edge objects. One or more key fields are defined by the configuration file, and these fields specify how node and edge data should be grouped. For example, if multiple flows are observed that involve the IP address 192.168.1.2, the time and number of bytes will be aggregated and recorded in a common data object rather than a unique object per appearance. This strategy has the effect of reducing the typical number of nodes that must be rendered and allows for some natural aggregations, such as quickly summing the total number of bytes from or to a given IP.

* An extensible network forensic analysis framework. <https://github.com/USArmyResearchLab/Dshell>

Data processing configuration also specifies the styles to be applied to the data. For example, textures can be assigned to nodes based on IP address, colors can be assigned to nodes and edges either randomly or according to a mapping, and the size of nodes and edges can be assigned according to linear or logarithmic scales.

4.2 Data Views

Vids alpha provides a variety of data views—currently eight different major types of data views—some with additional subtypes. Vids presents these data views to the user as a set of selectable layouts dictating how data are arranged within the virtual 3-D environment, and each data view has parameters that can be adjusted by the user. Such parameters include algorithmic details, such as the desired radius of a randomly arranged sphere layout or the repulsion versus attraction coefficient of a force-directed graph layout, and feature selection details, such as which data features should be plotted on the x-, y-, and z-axes or which features should be used to form groups of nodes.

The following graph layout options represent the current set:

- Spherical and cubic volumes in which nodes are placed either in regular fashion in equally spaced intervals distributed within the volume, or distributed on the outer surface of the volume (Fig. 3, upper right, right graph).
- Nodes ordered along xy-axes based on user-selectable attributes with regular node spacing and node stacking in the z-axis. Node position on the x-axis is determined by the magnitude of the node's x-axis mapped attribute relative to the magnitude of the attribute in other nodes within the graph. Node y-axis position is determined similarly using the y-axis mapped attribute, if one exists. Nodes with the same x-axis and y-axis position can either overlap or be stacked (placed at regular intervals) in the z-axis. The spacing between nodes is controlled by the available space in the graph or by a specified minimum value (Fig. 4, lower left, both graphs).
- Nodes placed in a scatterplot along x-, y-, and z-axes based on user-selectable attributes. Nodes are placed in position according to the magnitude of their axis-mapped attributes and are placed according to the selected scaling method for each axis, either linear scaled or logarithmically scaled. Node stacking, in which nodes with overlapping positions can be placed at regular intervals in another axis, is possible in the x- or z-dimension, depending on what other axes are already mapped to attributes.

If all x-, y-, and z-axes are in use for mapping, stacking will not occur and nodes will overlap by default.

- Force-directed (Hu 2006) algorithm plotting. This algorithm is principally used for node-edge networks in which no preexisting structure is present. In this algorithm, nodes are represented as repulsors while edges are represented as attractors (springs) connecting nodes, and the system is allowed to settle to an equilibrium over a series of iterations modeling the spring-repulsor interaction of the nodes and edges (Fig. 3, all images; Fig. 4, upper left).
- Group plotting of nodes based on user-selectable attribute. Nodes are collected into groups based upon the value of a specified attribute. These groups are then displayed in a number of different layout formulations:
 - Histogram chart, placing nodes along the xy-axis of the graph, where the x-axis position is an individual group and the y-axis indicates the overall number of nodes belonging to each group (Fig. 5, upper left, left graph).
 - Pie chart, acting similarly to the above histogram chart in the xy-axis of the graph, but grouping the nodes into slices of a pie chart (Fig. 5, upper left, right graph).
 - Spatial grouping of nodes in xyz-space within the graph. Nodes within a group are placed as tight clusters at different locations in 3-D space. This method is sometimes called “galaxy” visualization.
- Geographical globe plotting of locations based on longitude and latitude coordinates. Nodes are assigned a location based on their indicated physical location on Earth. A globe map of Earth is used as a prop for this layout. Edges curve outward from the surface to connect nodes.
- Circular plots by degree of connectedness. Nodes are plotted as a ring with edges plotted in the interior of the ring. The nodes are ordered by the number of edges that connect to or from them (Fig. 3, lower right, middle graph; Fig. 4, right images).
- Hierarchy plot based on direction of edge connections between nodes, similar to a cone tree plot. Source nodes—nodes that only have edges directed outward—are placed at the top of the plot in a large radius circle, and subsequent child nodes connected to these nodes are placed on smaller radius circles under their parent node (Fig. 5, upper right).

- 2- or 3-D Cartesian plane plotting for rendering reduced dimensionality data or data in which location embedding is meaningful. This methodology was used to support research on server log learning using dimensionality reduction and cluster recognition (Fig. 5, lower images). The task is somewhat similar to the one described in (Filho et al. 2018).

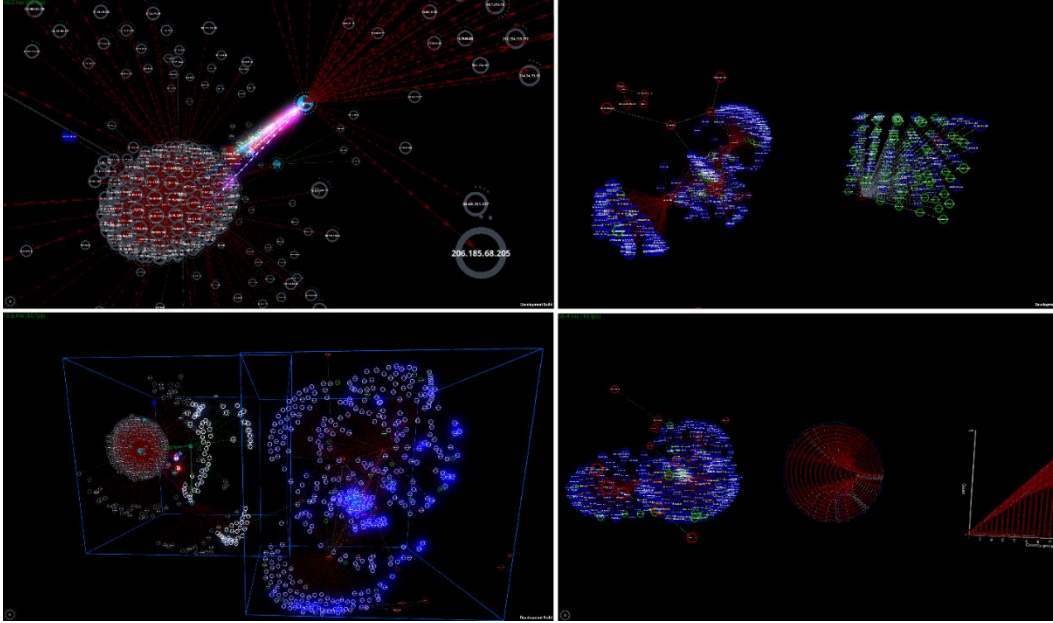


Fig. 3 Upper left: force-directed layout of network flow data with selected links glowing. Upper right: force-directed and cubic layout of network flow data. Lower left: two force-directed network flow data graphs, showing highlighting on common attributes between graphs. Lower right: force-directed network flow data, circular graph network flow data, and 2-D data chart.

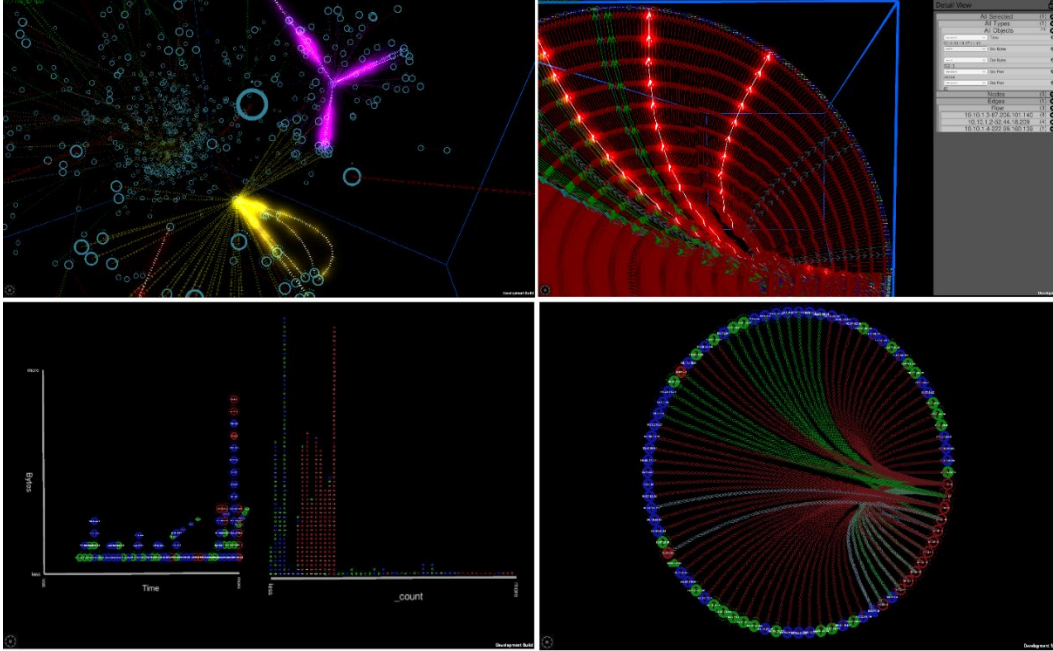


Fig. 4 Upper left: force-directed network flow graph with selected flow edges. Upper right: circular graph with selected edges. Lower left: two types of x-y charts; on left, x axis represents time, y axis represents number of bytes observed per datum, and the z (depth) axis represents multiple datum points occurring at the same time and byte count; on right, x axis represents the number of recurrences of a key within the data (for example, the number of times two IPs communicated with each other over a period of time), with the y axis representing a frequency tally of how many data points fall into each count range. Lower right: circular graph showing network flows.

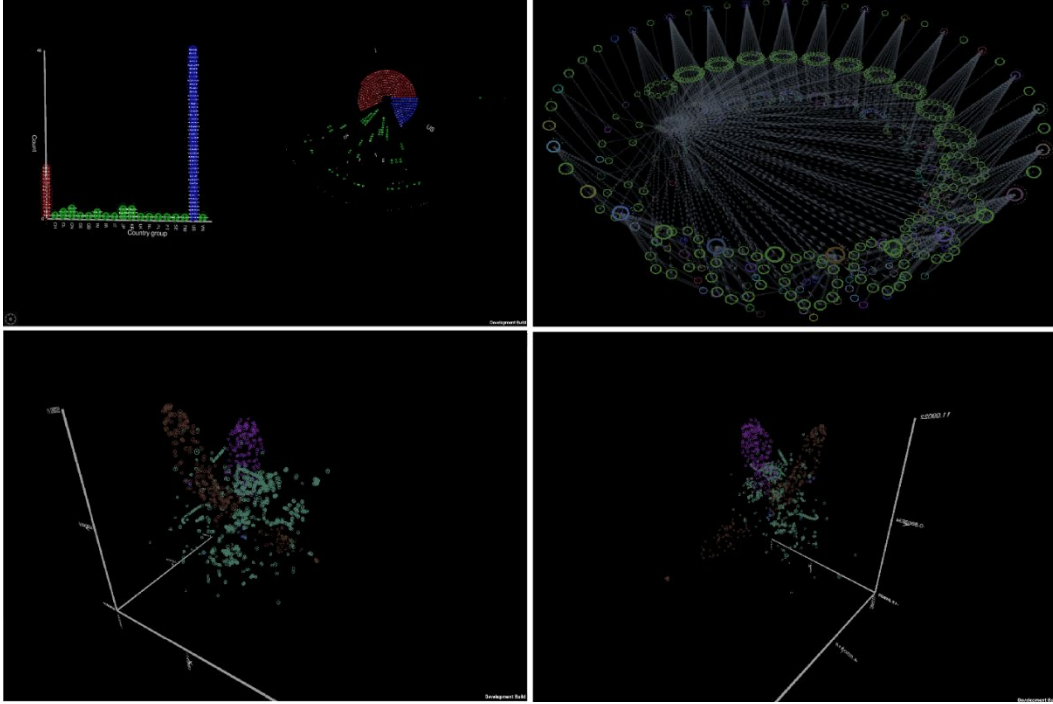


Fig. 5 Upper left: on left, data are grouped by country code on x axis, with a frequency tally for each data point falling into each category, creating a frequency histogram where each point represents a unique and interactive data point; on right, data are grouped by country code on a pie chart. Upper right: hierarchical data are organized into a multitiered layout. Lower left: higher-dimensionality server log data encoded and dimensionally reduced using t-distributed stochastic neighbor embedding into a 3-D embedding to highlight clusters of similar data (color coding). Lower right: same as lower left, rotated to a new perspective.

4.3 Nodes

Nodes in Vids represent a basic entity that can exist in a graph independent of other objects. In essence, nodes in Vids function as they do in any standard node-edge graph. The users can interact with the nodes via selection, detail examination, and movement functions. The rendering style of the individual nodes is configurable in eight different ways via style parameters. The style parameters may be configured by the user or the developer of the visualization to reflect the data the node holds. For example, node colors can be assigned to represent different groups of data, and the node texture may be an icon for the type of node represented. These style parameters include the following:

- Shape of node (i.e., a 3-D model used to represent the node)
- Scaled size of node
- Texture mapped to node shape
- Color (albedo) applied over node texture

- Material (including glow effects) with which the texture is rendered
- Node label text
- Rotation rate of node
- Node color pulse rate

Nodes also have a built-in counter feature to provide a visual indication of the number of times a data key appears in the data. In addition to the shape, texture, and color used to represent the node, Vids includes an additional ring of up to 24 dots around the node object, based on a count of how often the entity represented by the node appears in the data. This feature allows all nodes to be consistently sized, which reduces incidence of node overlap or other accidental occlusion of nodes within a visualization.

4.4 Edges

Edges in Vids relate a node to another node. Edges function in a similar manner in Vids as they do in a standard node-edge graph. They link nodes according to specific start and end key parameters. In addition, edges can hold informational attributes unique to each edge. Users interact with Edges in a similar way to nodes; they can be selected, examined, and moved. Edges also have style parameters similar to nodes. The exact set of parameters is subject to change during development, but active and planned features currently include the following:

- Edge thickness
- Edge start and end color, displayed as a gradient if they differ
- Edge texture
- Edge material
- Edge direction indicator density
- Edge movement simulation

4.5 Interaction

A critical component of Vids is the ability for users to interact with the displayed data and the environment in which the data are displayed. Interaction allows the user to more closely examine the data, either in aggregate or in detail. The viewpoint can be rotated to view a graph at new angles, possibly revealing data that would have been obscured in a 2-D representation. Interaction also allows users to dig into data via a contextual window panel to obtain additional information not

usually available in a graph visualization, including aggregated data, and sums, averages, and extremes of the data.

In Vids, every node or edge belongs to a graph. These graphs are selectable. Any click on a node or edge object belonging to the graph will select the graph, and a second click on the node or edge object selects the node or edge object. More than one object can be selected at a time by drawing a box around the objects using left click, or by using key-based multiselection. When one or more objects are selected, a display window can be opened that includes both individual information for each node and aggregated information from all selected objects. The aggregated information is collected from each node and edge's attributes data.

Selected objects can be manually moved to different locations. Users can click and drag nodes freely through space, allowing them to organize data as needed. Edge movement is handled by moving the control point for the quadratic Bézier curve traced by the displayed edge. Clicking and dragging a graph can move it, allowing the creation of custom dashboards for multiple data sources. In addition to basic selection features, some additional features exist for selected objects.

Vids alpha includes a generic and always accessible *Main Menu* for creating graphs, listing and selecting graphs, saving and loading dashboards (collections of graphs), adjusting display settings, displaying help information, and quitting the application.

When one or more graphs are selected, the menu expands to include a *Graph Menu*, containing three major graph-specific actions, setting the layout of the selected graph, setting the styles (color and texture mapping, sizing, line directionality, etc.) used on nodes and edges within the selected graph, and setting the filter (time interval, value range, value list) used to screen data in the graph. Additionally, when the *Graph Menu* is open and at least one graph is selected, a timeline bar is created to help the user control the time window for the currently selected set of graphs.

When one or more nodes or edges are selected, the menu expands again to include a *Selection Menu*, containing options to tag and correlate objects. The tag option effectively allows users to add a custom comment to the object label and color code the label as desired. The correlate option is a form of search function that performs a search and selects other nodes and edges that share specific attribute values.

5. User Feedback

Three hands-on sessions were held on 21 June 2019, 45 min per session, with analysts and team leads from the ARL (now C5ISR) cybersecurity service provider team. The intent of the exercise was to validate the project, ensure the prospective users agreed with the intent of the project, and solicit feedback on aspects of the

software such as feature utility, requested features, bugs, and issues. The state of development at the time of the feedback session was mid-alpha, meaning partial development of the major features. Therefore, the feedback session reviewed an in-progress version of Vids. The feedback mostly identified (no particular order) bugs, recommendations/requests, and commentary.

Several themes were identified from the feedback:

- Customizability is highly important in terms of the user interface and user interaction with data—about 45% of the recommendations/requests feedback we received was related to customization options (adjustable settings for fonts, movement sensitivity, user profiles)
- Data interaction desired (ability to dig deeper into data from visual) or ability to tie into another program for deeper interaction
- Detailed feedback on preferred navigation, interaction, and visualization mechanisms
- Concurrence that the provided project overview at the time of the feedback session matched the hands-on experience (i.e., Vids was in alignment with goals and expectations)

6. Lessons Learned

The Vids developers collected observations on lessons learned in the course of Vids development in addition to areas that could be revised in a future version of Vids. These range from general observations to specific design choices. The following lessons learned were adapted from developer comments:

- The Unity Real-Time Development Platform was a good choice as a platform for 3-D visualization of cyber data. Reasons cited include excellent documentation, a large user community, and good cross-platform compatibility.
- Organizing the code base was problematic. Because the Vids project represented something of an exploratory design effort, the original design was not as thorough as is ideal for a large software development project. A code-refactoring pass would be desirable to streamline object roles and interactions. Ideally, in a new project, object roles and responsibilities would be defined in the project's design phase and code would be organized accordingly. As an example, it should be decided in advance what object can control edge animation.

- Different data reading and processing methodologies involve making trade-offs. The original Vids concept envisioned a highly flexible architecture with multiple heterogeneous data sources being read simultaneously, with the data merged into a common format inside the Vids application, and displayed homogenously. While this goal was met, development experience proved this approach to have significant downsides compared with an application tailored to a specific and well-characterized singular data source. Allowing for multiple data input sources necessitated the construction and seamless integration of multiple data source interaction tool chains. Multiple tool chains meant that data structures, the user-customizable mapping of data fields to objects, and the user interface design to support each tool chain were more complex compared to a single tool chain. The developers suggest that the degree of desired data input flexibility should be carefully considered before developing a visualization tool. If only a single, well-characterized data source is expected, application complexity can be significantly reduced with a corresponding savings in resources.
- While the concept of data merging in the Vids application worked, it limited the amount of data that could be realistically processed in the application before performance degradation became noticeable. For larger quantities of data, the developers believe a better solution would be the use of a database separated from the Vids application.
- User interface design was more challenging and time consuming than originally anticipated. During development, the basic Vids functionalities were built before the user interface was designed. This sometimes resulted in unusual interaction patterns and a lack of overall user interface design coherence. The developers believe the Vids user interface would benefit from a user-oriented redesign and recommend that future visualization projects should carefully consider user interface design at an early stage and should follow a unified design pattern that fits with user's expectations of how they expect user interfaces to function.

7. Unexplored Areas

There were feature areas that remained unexplored at the end of the ARL Network Security Branch development period. These areas can be broken down into two groups: 1) incremental feature improvements and 2) extended vision objectives. The incremental feature improvements consist mainly of features that were scheduled to be implemented in the beta development phase and are considered to

be items whose development time and scope are fairly well constrained. Their development is considered feasible and relatively straightforward. The list of incremental feature improvements is as follows:

- Node grouping based on a system in which nodes could collapse or hide inside a “parent” node. Such a system would allow the user to collapse and expand hierarchies at will, a feature that exists in other visualization tools and potentially acts as a graphical resource saving option by reducing the level of detail of data at long view distances. One early concept for cyber data visualization suggested a “galaxy”-like view in which a large quantity of data would be presented in hierarchical form, with only a small number of nodes expanded and visible at any one time. This feature should be feasible to implement without great difficulty.
- Data input formats beyond CSV format. We originally intended JSON to be another accepted data input format. JSON is more suitable for relational and hierarchical data and is in some cases the native output data format for some tools. We also considered XML and GML input parsers as useful features. Generally, we found it possible, if not always elegant, to convert data into CSV format. Therefore, this feature was not implemented but remains on the list of features for future development.
- Improved color palette options for coloring nodes and edges, including color wheel and saved palettes. While the color-selecting user interface panel we developed was functional and easily implemented, it was unfamiliar to users; therefore, a more conventional color-selecting methodology using a color wheel was planned, and we intended for users to save their color palettes for repeated application across multiple data sets.
- Comprehensive user interface design additions and improvements. The following features and updates were planned:
 - An option to control the amount of nodes displayed at one time
 - Changeable icon sprites and edge textures
 - Rebindable hotkey mappings
 - An always-visible color mapping legend
 - A second pass on the exiting filtering user interface panel design

The extended vision objectives were major features whose implementation details and schedule were unknown at the end of the ARL Network Security Branch development period:

- Bidirectional interaction between the display environment and the data source. Early on in project development, we envisioned it would be possible to modify data in/on the data source (e.g., a file, a database, a remote database) by interacting with that data in Vids. The largest challenge was that the desired flexibility in input data formats and sources precluded a one-size-fits-all solution, making display to data interaction very logistically challenging (i.e., it could possibly be implemented, but would be very complex). In retrospect, we believe that attempting to maintain input data flexibility may have been a mistake (see Lessons Learned for additional insight).
- A data collection mechanism for research studies integrated into Vids. We intended to use Vids as the platform for a study on user interaction with the 3-D visualizations. However, since this project stage was not reached, and little preliminary work was done considering what metrics would be tracked, we consider this mechanism to be a feature whose implementation details and overall feasibility are unknown.

8. Conclusion

The Vids project aimed to bridge the gap between existing visualization technologies and the anticipated future of VR and AR capabilities. A number of cybersecurity data visualization methods were envisioned, developed, and evaluated by analysts, informing future design efforts. Lessons were learned on the design and development of a cybersecurity data visualization platform, and the code and development activities have been transitioned for incorporation into the VRDAE tool. Overall, the Vids project provided a step forward toward creating a cybersecurity visualization capability that could greatly benefit cyber defender's situational awareness and ability to recognize trends and patterns in data.

9. References

- Alberts D, Kott A, Rivera B, Chan K, Scott L, Hobbs R, Leung A, Dron W, Chadha R. Network science experimentation vision. Adelphi Laboratory Center (MD): Army Research Laboratory (US). 2015 Sep. Report No.: ARL-TR-7451.
- Ferebee D, Dasgupta D. Security visualization survey. Proceedings of the 12th Colloquium for Information Systems Security Education; 2008 June 2–4; Dallas, TX, University of Texas.
- Filho JA, Rey MF, Freitas C, Nedel LP. Immersive visualization of abstract information: an evaluation on dimensionally-reduced data scatterplots. IEEE Conference on Virtual Reality and 3D User Interfaces; 2018 Mar; Reutlingen, Germany. IEEE. p. 483–490.
- Hu Y. Efficient, high-quality force directed graph drawing. *Mathematica J.* 2006;10(1):37–71.
- Kullman K, Ben-Asher N. Operator impressions of 3D visualizations for cybersecurity analysts. Proceedings of the 18th European Conference on Cyber Warfare and Security; 2019a July; Coimbra, Portugal.
- Kullman K, Cowley J, Ben-Asker N. Enhancing cyber defense situational awareness using 3D visualizations. Proceedings of 13th International Conference on Cyber Warfare and Security ICCWS 2018; 2018 Mar; Washington, DC, National Defense University.
- Kullman K, Ryan M, Trossbach L. VR/MR supporting the future of defensive cyber operations. Proceedings of the 14th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems; 2019b Sep 16–19; Tallinn, Estonia.
- Marriot K, Schreiber F, Dwyer T, Klein K, Riche NH, Itoh T, Stuerzlinger W, Thomas BH. Immersive analytics. Gewerbestrasse (Switzerland): Springer; 2018. (Lecture Notes in Computer Science book series; vol. 11190.)
- Payer G, Trossbach T. The application of virtual reality for cyber information visualization and investigation. In: Blowers M, editor. Cham (Switzerland): Springer; 2015. *Evolution of Cyber Technologies and Operations to 2035*. p. 71–90. (Advances in Information Security; vol. 63.)
- Peckman E. How Unity built the world’s most popular game engine. *TechCrunch*; 2019 Oct 17 [accessed 2020 Jan 30]. <https://techcrunch.com/2019/10/17/how-unity-built-the-worlds-most-popular-game-engine/>.

- Robertson A. Unity officially releases its new game engine: Unity 5. The Verge. 2015 Mar 3 [accessed 2020 Jan 30]. <https://www.theverge.com/2015/3/3/8142099/unity-5-engine-release>.
- Sarikaya A, Gleicher M. Scatterplots: tasks, data, and designs. TVCG. 2018;24(1):402–412.
- Shearer G, Edwards J. Vids: Version 2.0 alpha visualization engine. Defense Technical Information Center. 2018 Apr [accessed 2020 Feb 3]. <https://apps.dtic.mil/dtic/tr/fulltext/u2/1050618.pdf>.
- Shiravi H, Shiravi A, Ghorbani AA. A survey of visualization systems for network security. TVCG. 2011 Aug;18(8) 1313–1329.
- Sicat R, Li J, Choi J, Cordeil M, Jeong W-K, Bach B, Pfister H. DXR: a toolkit for building immersive data visualizations. IEEE TVCG. 2018;25(1):715–725.
- Trossbach LC Jr, Robinson PE, inventors; ICF International, assignee. United States patent US 9,142,102. 2015 Jan 8.
- Zage DM, Zage WM. Intrusion detection system visualization of network alerts. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2010 July (unpublished). <https://apps.dtic.mil/dtic/tr/fulltext/u2/a532723.pdf>.

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
3-D	three-dimensional
AR	augmented reality
ARL	Army Research Laboratory
CA2X2	Computer Aided Analysis, Exercise, Experimentation
CCDC	US Army Combat Capabilities Development Command
C5ISR	Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance, and Reconnaissance
CSSPWG	Cybersecurity Services Provider Working Group
CSV	comma-separated value
FRB	Federal Reserve Bank
GML	Geography Markup Language
IDS	Intrusion Detection System
IP	Internet Protocol
JSON	JavaScript Object Notation
M&S COE	Modelling and Simulation Center of Excellence
MR	mixed reality
NATO	North Atlantic Treaty Organization
NIRT	National Incident Response Team
TAB	Technical Advisory Board
VIDS	Visual Intrusion Detection System
VR	virtual reality
VRDAE	Virtual Reality Data Analysis Environment
XML	Extensible Markup Language

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 CCDC ARL
(PDF) FCDD RLD CL
TECH LIB

2 CCDC ARL
(PDF) FCDD RLC ND
G SHEARER
J EDWARDS