EG UK Theory and Practice of Computer Graphics (2013), pp. 1–7
Silvester Czanner, Wen Tang (Editors)

NetVis: A network traffic visualization tool

SUBMISSION ID

AFFILIATION

Abstract

Computer network traffic visualizations deliver improved understanding of traffic on a network to an observer. Many existing tools opt for graph or plot-based visualizations to detect patterns or outliers in the data, but they still largely provide a segmented view of any data feed. In this paper, we present a novel network traffic visualization framework that makes use of a variety of complementary visualizations to obtain better situational awareness. Our proposed solution NetVis provides six different visualizations that work in tandem to provide situational awareness of the network traffic passing through. Three of the visualizations are based on existing literature (parallel coordinate plots, flowscan and spinning cube of potential doom), whereas the three remaining visualizations are novel to this paper. Our results show that it is possible to use the tool to detect unusual activity and cyber attacks on a network. The framework is written in a way that allows future visualizations to be straightforwardly added.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—K.6.5 [Management of Computing and Information Systems]: Security and Protection—

1. Introduction

Analysing and monitoring network traffic is an important part of network security. Since the amounts of data involved in this task are extraordinarily large, it can be hard to make effective use of them. Visualizations offer a solution: they give a compact representation of data to a human analyst who can use them to detect patterns in the network [War12]. An effective visualization of network traffic makes it easy to identify outliers while making clear the general patterns of network usage, and it will facilitate detection of intrusions and malicious activity.

In this paper, we present a set of visualization techniques that work in tandem to improve awareness of activities in ongoing network traffic passing through an analysts systems. They have been implemented in a robust application to provide an analysis framework for network activity in an interactive manner, enabling a deeper and more straightforward understanding of the data. Two of the visualizations are based on existing literature (parallel coordinate plots [Ins85] and spinning cube of potential doom [**?**]), whereas the four remaining visualizations are novel to this paper. These are referred to as: Attribute Distribution, Traffic Volume, Heat Map and Activity Groups.

Time series data in the form of packet captures are processed in real-time and simultaneously rendered in multiple connected visualizations. The user can switch between the available representations of the data and change both the visualization's layout and the amount and type of the data displayed. The software is designed to make it easy to spot irregular activity and investigate it from multiple perspectives.

The framework aims to provide an environment in which situational awareness can more effectively be obtained. It is therefore flexible to the demands of the specific situation. New visualizations will in a natural way complement the existing ones: The underlying data processing engine provides a standard data basis which is shared by all displays.

The tools emphasize exceptions, show comparisons, and answer a wide variety of questions in a concise fashion. We want the user to generate good hypotheses in response to the visualized information. To achieve this, aids are given to the user to avoid cluttering displays with irrelevant data. In particular, we have implemented a filtering system which can be adjusted in response to changing situations.

Though the application's main purpose is to monitor current activity in the network, the same architecture can be used to forensically analyse recorded data. The system simulates the data records as if they represent a live network.

This exploits the important dimension of time and makes it easier to understand recorded activity.

2. Related Work

To this day, most visualization approaches provide a segmented view of a dataset. These provide different perspectives on the same data. Some of these may favour graph-based representations, port-scanning activity characteristics, network traffic patterns, payload characteristics or event-log forensics. Conti [Con07] and Marty [Mar09] provide a detailed discussion on this topic.

Some examples of available tools include the open source Rumint [Con] and Wireshark [Ger] for traffic forensics. Many tools for analysing network usage patterns exist [BBL*10, KKB04, Lau04, LSC10]. Other approaches include geographical-based representations of malware activity, such as Sony Rootkit Global Spread [Con07], and on city-level visability by Yu et al. [YLRB10]. The potential wider impact of attacks on network assets was visualised by Chu et al. [CIL*10]. Intrusion-detection event-correlations have also been visualized by Rasmussen et al. [RER*10] and Yelizarov et al. [YG09].

The commercial SecureScope tool [Sec] addresses business impact of attacks by mapping clusters of potentially malicious network activity to business role or organisational units (such as *Human Resources*) and geographical location (such as the *New York Office*). Similarly, the commercial Arcsight tool [Arc] provides mapping between event alerts, source IPs and business role. Another software application is the Tenable 3D Tool [Ten], which can visualise topology based on vulnerability scans and change the visible features of machines accordingly. These features encode information about vulnerabilities, missing patches, open ports, firewalls, intrusion detection system alerts, netflow, etc.

While many tools exist to enable situational awareness, few works have discussed how multiple visualization techniques can complement each other and work in tandem to deliver an improved situational awareness from network activity data. This is the main problem NetVis addresses.

3. NetVis Architecture

NetVis is a Java application which uses OpenGL to draw visualizations and Swing to provide a graphical user interface for displaying further information and allowing the user to customise the visualizations.

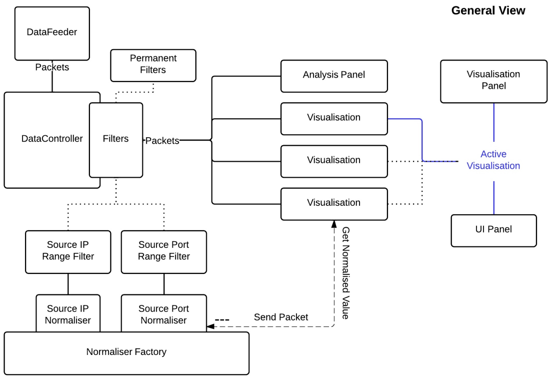
In designing the application, we have focused on maintaining extensibility and keeping a modular programming style. It is simple to add support for additional input data formats, or to develop a new visualization that utilises the same data processing engine.

Packet data is transferred to the application as a CSV packet capture file. NetVis uses a file format that can be directly obtained from the packet analyser Wireshark and other similar applications. NetVis will process each packet and analyse its fields which includes a time stamp, information about the packet's source (IP, hardware (MAC) address, port), information about the packet's destination, communication protocol, and the packet length in bytes. The current version of NetVis does not process the content of packets.

A convenient feature of the data input system is a time control system. The CSV file is processed as if its packets would arrise in a live network. The user can choose to speed up or slow down the speed with which packets are fed into the application, can pause processing, or skip towards the end of the data record. This is helpful in analysing the data since critical time intervals can be analysed in more detail.

The application is set up in such a way that it is also easy to use the activity of a live network as its input source.

The input data is processed in a data controller which supplies the visualizations and the user interface with packet data. If the user has chosen to apply filters to the data, the data controller only directs the filtered data stream to the rest of the application, so that all parts share a common data source.



NetVis includes a data filtering system that allows users to select a subset of processed packets which exhibit features which the user specified to be of interest.

The application supports two types of filters: filters that the user explicitly defines, and filters applied on-the-fly from within the visualizations. Both types are applied to all visualizations and information displayed, and they can be adjusted at any time without losing prior data.

There are a variety of filter controls. First, there is a menu for filtering by transit protocol. By default, all protocols are selected and therefore included. Protocols are sorted into menus by protocol family, appearing in multiple places where appropriate. Second, there is a control to select the range of ports the application uses, which defaults to the maximum port range. The source and destination ports can be set separately, enabling a user to view all data entering/exiting a port as desired. Third, there are IP and MAC address filters. These work on a blacklist/whitelist system, allowing a user to only view packets to or from a particular set of addresses, or to ignore packets going to or from a different set. This enables a user to, for instance, ignore traffic from sources they know are irrelevant or focus only on an address that is causing concern.

The packet attributes that are processed by NetVis do not share a common scale, nor are they necessarily orderable. It is, however, desirable to map some of these properties into a shared representation which allows a more intuitive grasp of the distribution of attributes. We achieve this by processing the packets in a 'normalising' system. This system keeps track of the used values and maps them into the interval $[0,1]$. In this way all normalised packets can be displayed in relation to a number axis. This representation is used in three different visualizations.

Each normaliser is able to create a temporary filter in the application that will filter its corresponding attribute on a certain range. Since the normalising class is in control of its filter this creates a zooming effect based on the range of the filter: the lower bound is normalised to 0, the upper bound to 1.

4. NetVis Visualizations

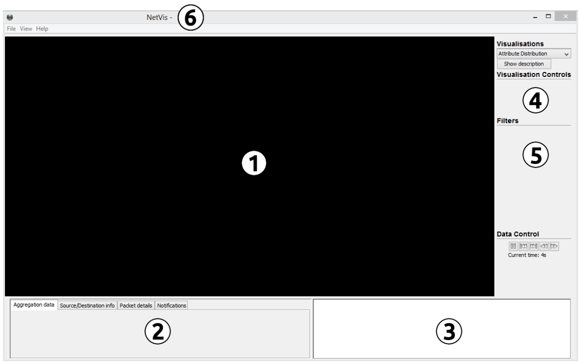
The user interface of the application focuses on the displayed visualization, which takes up the majority of the window and can be maximised to fill the available space. Other panels occupy the right and bottom sections of the window.

The right panel is aimed at providing the user with options to adapt the shown data. Choice of visualization is presented to the user by a list in the top of the right panel, followed underneath by options specific to that visualisation, and then by general filters to refine the processed data. If a recorded data source is active, time controls are displayed on the right panel to allow the user to adjust the speed at which data is processed and visualised by pausing, doubling or halving the current data rate.

The bottom section shows fine and aggregate details of the processed data. To fit the large amount of data available into such a limited space, a tabbed pane is used on the left 'Analysis Panel' to categorise distinct types of data. The right section of this panel is a 'Context Pane', which shows further detail of selected data on demand. The left 'Analysis' panel and the right 'Context' panel are separated by a split pane, so the user can modify the proportion of each they wish to see.

Window tools, an exclusive-mode full screen option and the facility to select a new data source are all accessed by the main menu bar. Messages to the user come in the form of warning dialogues for user or program errors, notifications in the bottom 'Analysis' panel for program notifications, and text messages in the bottom 'Context' panel for help messages and suggestions.

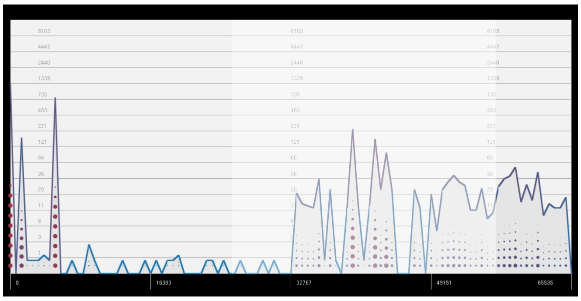
Below is a diagram of the GUI with the most important components labelled.



Key

1. Visualization
2. Analysis Panel
3. Context Panel
4. Visualization-specific options
5. General data filters
6. Data source identifier

4.1. Attribute Distribution

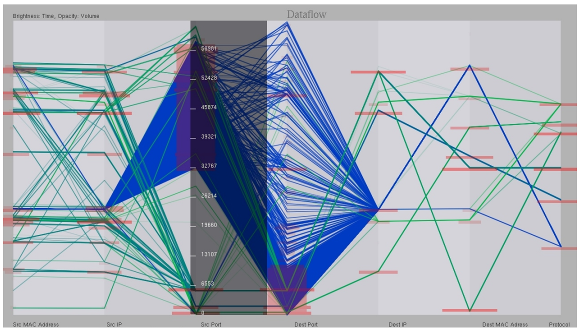


The most common need in the analysis of network traffic is information on the distribution of certain features of the packets. For the graph constructed here, the user selects such a feature (e.g., ports or destination IPs). A logarithmic line graph is drawn and (as with all visualizations in the application) updated in real time. A logarithm is applied to assist in detecting spikes.

A second layer of the visualization is presented simultaneously: circles under the line graph indicate the actual distribution (without a logarithm applied) through their area. Furthermore, all lines and circles are colour-coded to give extra signals of the volume of traffic. Red elements are under heavier load.

This visualisation makes active use of the normalisers described above. It also allows the user to click-and-drag along interesting data to zoom into this area. Internally this means that a filter is applied and all visualizations show only the selected packets so that they can be analysed in more detail.

4.2. Data Flow



The Data Flow visualization applies the idea of parallel coordinates proposed by [Ins85] to network traffic. Following the principles in [FM] the visualization shows the ‘flow’ of each packet, representing each as a line through the parallel coordinates. This provides an informative view of the whole traffic in the network. It visualises the distribution of various packet attributes while also giving an intuitive insight into relationships and correlations between distinct aspects of the packets.

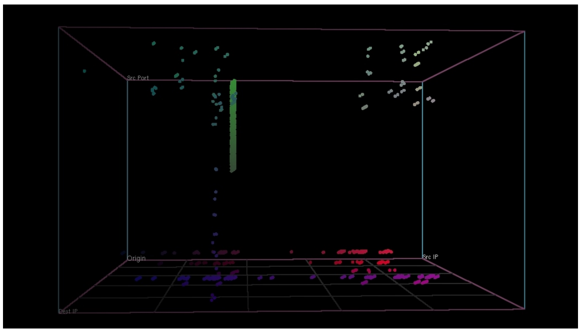
Lines representing packets are coloured based on their value in some coordinate. They fade out based on how old the packet is, thereby placing focus on newer developments in the network.

A common criticism of parallel coordinate plots is that they make it hard to interpret data that is uniformly distributed between coordinates or that is very concentrated around few values [Mar09]. Our implementation addresses these concerns using animation. Lines representing new packets are randomly perturbed and move slightly. This makes it easier to spot whether a line represents one or more packets. The animation allows the user to distinguish between single packets and concentrated groups of packets with the same characteristics.

Suppose, for example, that multiple requests to a server come from a single MAC address through the same ports and using the same protocol. A non-animated visualization would represent all these requests as a single line. A user would erroneously interpret this as a single request. Adding randomness makes the pack of requests more visible without significantly impacting the accuracy of the data representation.

The Data Flow visualization is closely tied to the previously described Attribute Distribution visualization. If a coordinate shows an unusual distribution (say a uniform use of ports in a certain range), then a single click on the packets in this coordinate will show the Attribute Distribution log plot which allows direct access to filtering option. Hovering over packets also displays further contextual information.

4.3. Spinning Cube

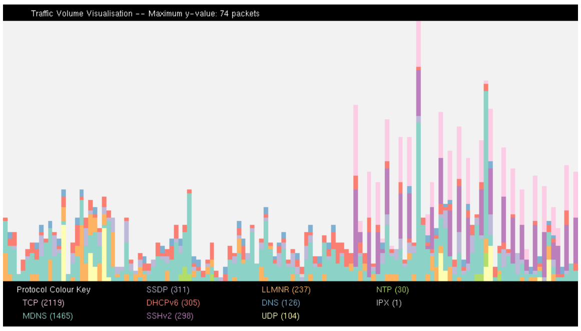


The Spinning Cube is a three-dimensional scatter plot that displays packets as dots inside a rotating cube. The position of the dot is determined by its packet’s attributes. This could be the combination of source port, destination port, and source IP but also any other choice of packet attributes.

The visualization is an implementation of an existing visualization tool known as the Spinning Cube of Potential Doom [Lau04], but it offers additional control for the user. In particular, the axes can be chosen to represent arbitrary packet attributes.

To ensure good distinguishability of individual packets, we use the same principle of randomness as in the Data Flow visualization. Every point is animated and perturbed. In this way, multiple packets with the same attributes do not occupy the same pixel in the cube. It is thus easier to spot the packet density of an area in the cube.

4.4. Traffic Volume

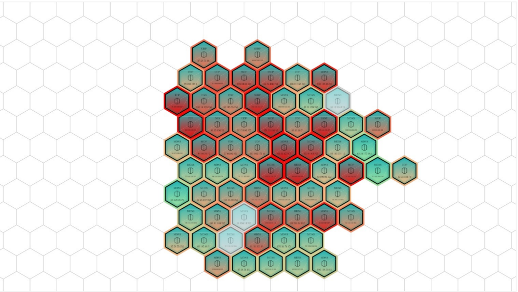


The Traffic Volume visualization is inspired by the ‘FlowScan’ graph [Plo00]. The objective is to allow users to see at a glance if a particular protocol is being exploited in the network. It is realised as a stacked bar chart which displays the volume of data arriving in each time interval. Each column is segmented into sections with heights proportional to the total number of packets transmitted for each protocol. In addition, the column segments are colour-coded and can be cross-referenced with a protocol key underneath the visualization.

To help distinguish between different protocols, colours are selected from a palette which provides colours from a qualitative colour scheme. The increase in both column height and colour proportion should draw attention to any protocol which becomes overburdened. For instance, if a network saturated with TCP traffic suddenly becomes flooded with DNS requests, the drastic change in colour will alert the user to the change in circumstances.

A control panel is provided in the right panel for adjusting the number of time intervals displayed at once on the x-axis. The y-axis scales automatically to fit the current data.

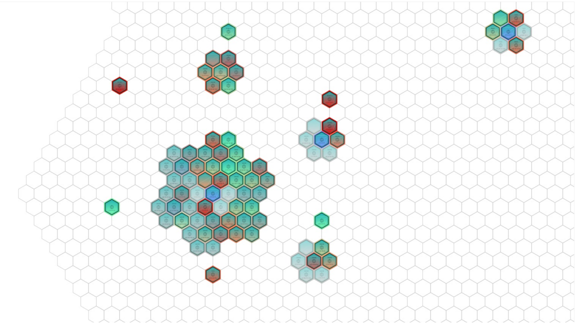
4.5. Heat Map



The Heat Map visualization uses a hexagonal map to display a compact cluster of nodes. Each machine found to be communicating in the analysed network will be represented as a node. The activity of a specific machine in the network is represented by the colour of the corresponding hexagon. As more and more traffic arises, nodes representing the machines that send and reveive data “heat up” and, as the time passes, “cool down”. The hexagonal grid is filled in a compact manner - new nodes are place outside of the already displayed nodes. The relative position of the map elements in this visualization does not play a role by default. There is a way to sort the nodes so that they form a radial gradient - the more “heated” will be sorted to the middle.

This visualization helps a user find machines that are most active in the network. It gives a quick overview of the number of the clients and how many of them are currently active. Additionaly every node presents some information, such as the machine’s most commonly used protocol. This lets users do some basic data selection choices easily. For example: if some other visualization is being obscured by too much data from one specific machine, or too much data of some protocol, the user can look at the heatmap and find out what filters to apply to make the overview of the situation more clear.

4.6. Activity Groups



This visualization also makes use of the hexagonal grid. In the attempt to give a best possible insight it uses two visual factors: size and proximity.

As the traffic in the network rises, the nodes representing machines are put on the map. All the machines sending packets to one specific device are grouped together. As more machines communicate with a specific address, more nodes appear around the hexagon representing their destination.

The colour of the communicating nodes represents how much data they send. More heavily used destinations will thus have more orange and red nodes around them, and destinations used by a lot of machines will have a greater number of nodes around them. Node placement is automatic. The procedure ensures that there is enough space around a centre and that no two nodes overlap.

This visualization helps detect the machines that perform a server role in the network. It also allows a user to quickly identify what type of service the device is providing. Each node specifies the most commonly used protocol, thus revealing the role of the server node.

5. Discussion

The main determinants of traffic in a network are the network’s topology, the flow of the traffic, and the type of the traffic. The graphical visualizations presented here ensure that the available data can be interpreted from all three of these perspectives. Moreover, better insight into the activity in the network can be gained by combining these perspectives.

Each visualization fulfills a specific purpose. The Heat Map visualization provides a very quick impression of overall network load and size. The Activity Groups visualization provides the same information on a server-by-sever basis, providing more data at the expense of simplicity. The Data Flow visualization provides the ability to see what the traffic of the network currently ‘looks’ like, and the Spinning Cube and Attribute Distribution visualisations allow the user to detect patterns in the attributes of the packets.

The visualizations developed here differ in their approach to handling the inherent multi-dimensionality of traffic data.

An effective visualization framework needs to make clear how traffic changes as time progresses, and needs to show interesting developments in diverse attributes such as port use, source and destination machines, protocol use or traffic volume. Showing all this information simultaneously risks obstructing the simplicity needed for ready understanding. NetVis uses both multi-dimensional systems (such as parallel coordinates) to give an overview and lower-dimensional visualizations to provide more detail. This encourages an understanding of network activity that is both broad and detailed.

Results: (couldn't find a section)

pic: portscan detected

Caption: The analyst can clearly see that a port scan has occured.

pic: portscan identify

Caption: By visualising the distribution of source IP's he can distinguish a spike in the network traffic for a small IP range.

pic: portscan explore

Caption: Applying a range filter isolates the port scan which he can explore further in all the visualizations.

SSH Attack:

pic: ssh_suspicious:

Caption: The analyst observes some irregular activity in the traffic volume visualization.

pic: ssh_explore

Caption: After exploring with a couple of filters he finds out that a single IP is responsible for all the irregularities. A quick look at the analysis panel provides information about the attacker and the victim. By isolating the packets from the suspicious IP he can determine the type of attack.

Results conclusion: An analyst can train for detecting attacks by using packet captures. After learning the shape of different attacks in each visualization it becomes extremely easy for him to detect an ongoing attack. By using a different filter combinations he can isolate the attack an learn more about it.

5.1. User Workflow

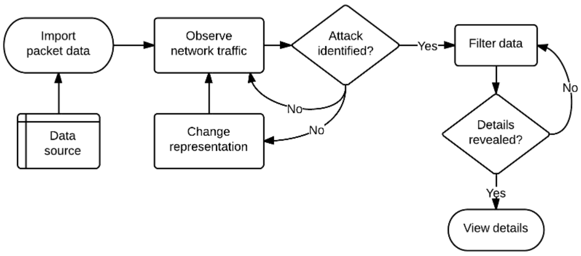
From the perspective of an analyst examining a network, a typical workflow would begin by considering all activity from a broad overview, using any number of visualizations. Once a network attack has been identified, it should be easy for the analyst to "zoom in" to the data of interest by applying successive filters to the source and changing the representation to suit the particular attack. Finally, when the exceptional behaviour has been singled out, the analyst may explore details of the intrusion.

This workflow illustrates the visual design framework suggested by Ben Shneiderman [Shn96]:

'Overview first, zoom and filter, then details-on-demand.'

To assist with the process of singling out suspicious activity, certain visualizations allow filtering by mouse interaction. Where appropriate, the analyst may click or drag their mouse cursor over sections of the active visualization to create a new filter or to switch to a different representation of the selected data. Navigation between complementing visualizations in this fashion is highly intuitive and allows the analyst to tweak the active filters quickly and effectively.

The following is a graphical representation of the typical workflow expected of an analyst studying a network using NetVis.



5.2. Advantages

Existing network visualization solutions are typically single-purpose programs which present given data in one specific way (see for instance the catalogue in [Mar09]). The analyst is expected to choose in advance which kind of visualization will provide the most insight, and is then limited to what the specific application is able to show. Existing applications also require different input file formats which makes simultaneous use complicated.

The program presented here takes a single data input stream and simultaneously visualizes it in different presentations. Since these visualizations are built upon a common base, they are able to complement and inform each other. This improves the user's ability to investigate anomalies.

Network visualizations have to process and display a large amount of data. Often, this can cause a visualization to become cluttered. An obvious solution is to provide filters which allow the analyst to focus on phenomena of interest. The filters we provide apply application-wide and can be defined in an intuitive manner from within the visualizations themselves in a "click-and-zoom" fashion. This is a significant improvement over existing workflows.

The framework developed here can be used both for real-time monitoring and for historical analysis, which makes it applicable to a wide variety of use cases. Furthermore, the application can be used as a learning tool: by observing archived data captures, new users can familiarise themselves with the general patterns of network usage and can see how suspicious patterns show up in a visualization.

5.3. Limitations

The tools in our framework support only the analysis of time series data of packet captures. A multi-layer approach might help inform the user more concretely about the state of the network. In particular, information from intrusion detection systems, routers, and firewalls could be incorporated. The application in its current state does not analyse the content of packets.

5.4. Future Work

Our main avenue of future work is a more powerful data preprocessing system. Our current visualizations focus on giving the user a complete view on the network while allowing interactive filtering. However, if statistical inference is applied to the data and historical observations are incorporated, visualizations can be more effective in pointing out suspicious patterns and anomalies.

Future work should need to validate the usefulness of our application by testing it on other expert users in live environments instead of laboratory settings only.

6. Conclusion

We have presented a network visualization tool to analyse and monitor network traffic. Instead of presenting single perspectives (visualizations) of the data very well (but with a limited scope), our framework delivers a variety of visualizations that complement each other and work in tandem to help an analyst obtain improved awareness of ongoing network activity through user interaction. We overviewed its suite of visualizations and demonstrated its usefulness with example scenarios. In the future we intend to add more visualizations and further improve the tool's usability.

References

[Arc] ARCSIGHT: Arcsight enterprise security manager. http://www.arcsight.com. 2

[BBL*10] BEST D., BOHN S., LOVE D., WYNNE A., PIKE W.: Real-time visualization of network behaviors for situational awareness. In *Proceedings of VIZSEC 2010* (2010), ACM. 2

[CIL*10] CHU M., INGOLS K., LIPPMANN R., WEBSTER S., BOYER S.: Visualizing attack graphs, reachability, and trust relationships with navigator. In *Proceedings of VIZSEC 2010* (2010), ACM. 2

[Con] CONTI G.: Rumint. http://www.rumint.org. 2

[Con07] CONTI G.: *Security Data Visualization*. No Starch Press, San Francisco, CA, 2007. 2

[FM] FLIGG K., MAX G.: Network security visualization. 4

[Ger] GERALD COMBS: Wireshark. http://www.wireshark.org. 2

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91. 1, 4

[KKB04] KIM H., KANG I., BAHK S.: Real-time visualizaton of network attacks on high-speed links. *IEEE Network 18 Issue 5* (2004), 30–39. 2

[Lau04] LAU S.: The spinning cube of potential doom. *Communications of the ACM 47*, 6 (2004), 25–26. 2, 4

[LSC10] LIAO Q., STRIEGEL A., CHAWLA N.: Visualizing graph dynamics and similarity for enterprise network security and management. In *Proceedings of VIZSEC 2010* (2010), ACM. 2

[Mac87] MACKINLAY J. D.: *Automatic design of graphical presentations*. Tech. rep., Stanford Univ., CA (USA), 1987.

[Mar09] MARTY R.: *Applied security visualization*. Addison-Wesley, 2009. 2, 4, 6

[Plo00] PLONKA D.: Flowscan: A network traffic flow reporting and visualization tool. In *USENIX LISA* (2000), pp. 305–317. 4

[RER*10] RASMUSSEN J., EHRLICH K., ROSS S., KIRK S., GRUEN D., PATTERSON J.: Nimble cybersecurity incident management through visualization and defensible recommendations. In *Proceedings of VIZSEC 2010* (2010), ACM. 2

[S*46] STEVENS S. S., ET AL.: On the theory of scales of measurement, 1946.

[Sec] SECUREDECISIONS: Securescope. http://www.securescope.com. 2

[Shn96] SHNEIDERMAN B.: *The eyes have it: A task by data type taxonomy for information visualizations*. IEEE Symposium, 1996. 6

[Ten] TENABLE: Tenable 3d tool. http://www.tenable.com. 2

[War12] WARE C.: *Information visualization: perception for design*. Morgan Kaufmann Pub, 2012. 1

[YG09] YELIZAROV A., GAMAYUNOV D.: Visualization of complex attacks and state of attacked network. In *Proceedings of VIZSEC 2009* (2009), ACM. 2

[YLRB10] YU T., LIPPMANN R., RIORDAN J., BOYER S.: Ember: A global perspective on extreme malicious behaviour. In *Proceedings of VIZSEC 2010* (2010), ACM. 2