Interactive Visualization of Netflow Traffic

Thomas C. Eskridge, Marco Carvalho
Fitzroy Nebhard, Hari Thotempudi
Harris Institute for Assured Information
Florida Institute of Technology
Melbourne, Florida
mcarvalho,teskridge@fit.edu

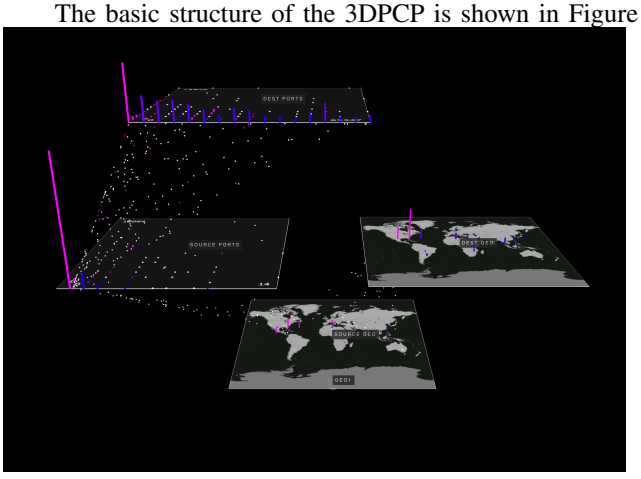
Peter J. Polack
Georgia Institute of Technology
Atlanta, GA
pjpolack@gmail.com

***Abstract*—We introduce a novel tool for visualizing netflow traffic on enterprise networks called the 3D Parallel Coordinate Planes (3DPCP). This tool provides operators with the ability to manipulate the visual flow of network traffic information by arranging two-dimensional planes along a vertical time axis in 3D space. The output of any plane may be split into two or more streams by adding additional planes and connecting them using a stream filter. The visualization system augments general situation awareness with the ability to isolate traffic meeting specific criteria and connect it to mission requirements. Particularly useful criteria can be collapsed into software agents that can tag the incoming netflow traffic data stream, enriching the visual representation of flows.**

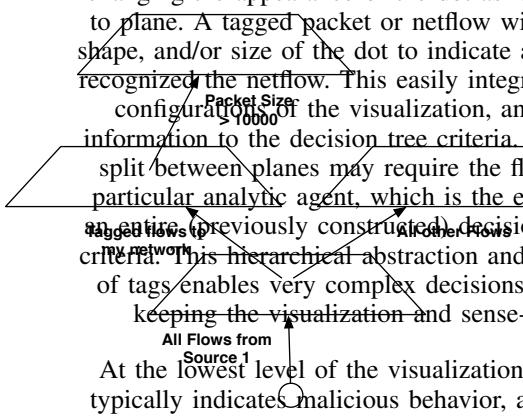
I. Introduction

A key problem in computer network defense is understanding the current state of the network, including what attacks are happening, what impacts the attack will have on current and future mission plans, if and how the attacks are changing, and what options are available to improve the situation [1]. In this paper, we present a visualization technique called 3D Parallel Coordinate Planes which enables operators to construct a visual representation of the activity on their networks, to explore hypotheses about threat behavior, and to understand how current network activities affect currently active and planned missions.

The basic structure of the 3DPCP is shown in Figure I.



and is illustrated in Figure 2. Each plane shown in the figure is a representation of some aspect of the incoming data, and is also a node in a decision tree. In Figure 2, the top plane shows the geo-located source IP for each event (i.e., packet or netflow) in the data stream. Below that, the second level plane changes the representation from geolocation to two aspects of the communication event, the packet size vs. bits per packet. The decision criteria then separates the packets between two planes, the one on the left that is used for IP addresses in our enterprise, and the one on the right that is for all other addresses. The fourth level planes split the data coming into our enterprise into those source IPs that are on a blacklist, and those that are not. The dots between the planes represent the packets or netflows in the incoming data, indicating a network connection between the source and destination. The purpose of the visualization is to allow the analyst to construct a visual structure that amplifies the differences between malicious and legitimate traffic, and presents a view of the important distinctions to be made in the network, as the analyst understands it.



By manipulating the criteria for the split, the investigator can configure the display to answer a number of important questions about the data. Particularly useful configurations of planes and filters may be stored for reuse, sharing with colleagues, or used in briefings to present the results of the investigation. As part of this effort, we propose to extract the decision tree component of the display, and construct an analytic software agent that can monitor incoming traffic and tag incoming packets or netflows with classification information. This type of tag is indicated to the operator by changing the appearance of the dot as it transits from plane to plane. A tagged packet or netflow will change the color, shape, and/or size of the dot to indicate an analytic agent has recognized the netflow. This easily integrates into subsequent configurations of the visualization, and adds additional information to the decision tree criteria. For example, a new split between planes may require the flow be tagged by a particular analytic agent, which is the equivalent of adding an entire (previously constructed) decision tree into the split criteria. This hierarchical abstraction and semantic grounding of tags enables very complex decisions to be made, while keeping the visualization and sense-making simple.

At the lowest level of the visualization, one of the planes typically indicates malicious behavior, and the other planes indicate different types of legitimate behavior. To create an analytic agent, the operator selects the malicious plane and, through a menu, initiates the agent construction. The agent construction algorithm looks up the path through the decision tree that would route a packet from the top plane to the malicious plane, and codifies this.

The name ”3D Parallel Coordinate Plane” is meant to evoke notions standard, two-dimensional parallel coordinate graphs. These graphs have proven to be very useful in analyzing large amounts of multivariate data. This proposal extends the parallel coordinates idea of using lines to connect values on linear scales, to time-referenced, annotated, and animated dots connecting values on a two-dimensional scale (the planes). By constructing such a representation in 3D space, the user can manipulate the location of planes to maximize their informativeness, and arrange them to highlight interesting separations in the data.

The MissionStack is a new interface concept that is designed to flexibly and efficiently display a range of mission and process dependence information and how the defenses and responses to attacks affect these dependencies, and ultimately, mission success. To do this, we visualize the interaction of mission steps and required information with the operation and performance of the underlying network. By visually inspecting these interactions, operators gain insight in to operations that impede or improve the chances for mission success.s

The mission stack is comprised of a number of layers, that show the mission and its interactions with the network at multiple levels of abstraction (see Figure 1).

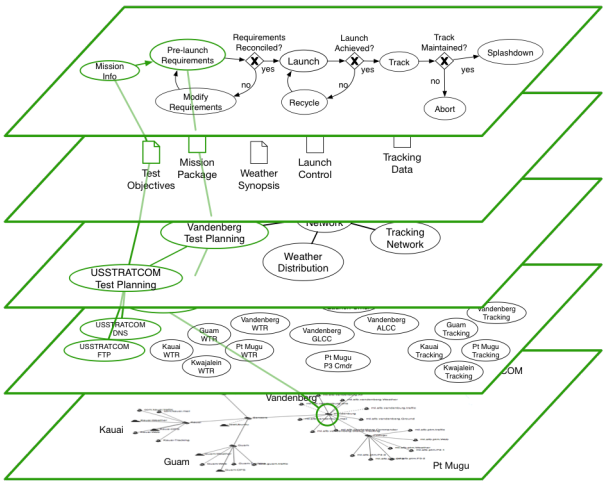


Fig. 1. Overview of MissionStack Interface Concept

In this example, there are five planes defined, and the vertical lines connecting them denote the existence of a mission dependency between an item on the top-most mission plane, and the elements beneath it, which are necessary to complete the mission element. We illustrate the concept with the example of a distributed backup service, where clients connect to servers in order to back up their locally stored files.

In this example, clients connect to a regionally located satellite server to begin the transfer process. After this initial connection, the next step to to authenticate with the backup company’s authorization server at its main site. After authentication, the client sends the satellite server a manifest of files to be backed up. Then, the satellite server pulls the files identified in the manifest from the client. These steps are illustated in the process flow diagram shown in Figure **??**.

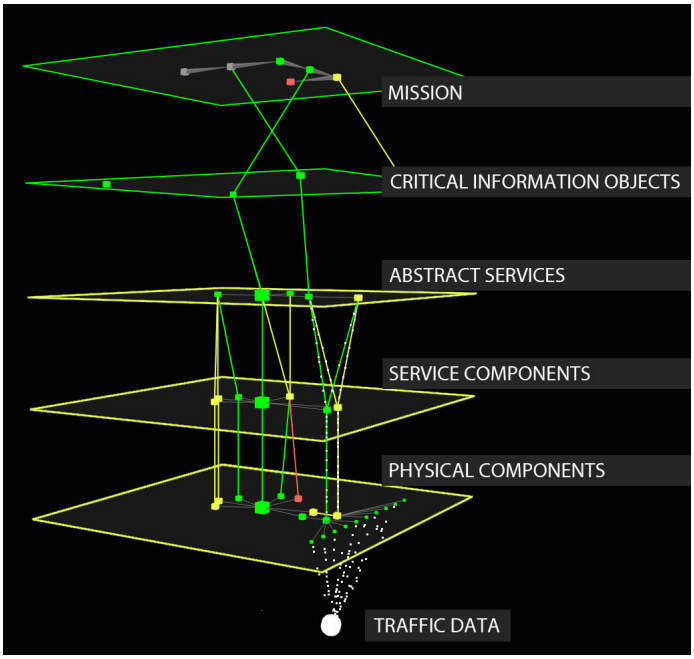


Fig. 2. 3-D Parallel Coordinates

There are a number of different types of planes that could be defined for us within the mission stack, including:

- Mission Plane: This plane shows the mission goals, processes, and process dependencies at the level of detail necessary to determine the level of risk for mission failure.
- Critical Information Objects: Particular data or information items are needed to complete individual mission steps within the overall mission plan. This plane identifies critical information objects, and relates them with the network elements that create them and the mission steps that require them.
- Abstract Services: This plane shows a high-level view of the services necessary to support critical object creation.
- Service Components: This plane shows a lower-resolution view of the services needed to support critical object creation.
- Physical Components: Physical components are the actual hardware and network components needed to create the critical information objects and accomplish the mission.

A key aspect of the MissionStack visualization is the visual representation of dependencies both within an abstraction level (i.e., within a plane) and between abstraction levels.

In the next sections, we provide more detail about each plane in the example scenario.

II. Mission Plane

A principal goal of the MissionStack visualization is to graphically depict the mission status and how the attacks on

the network - and the associated introduction of defenses and other responses meant to mitigate the threat - affect the success of the mission. In most mission-driven implementations of the MissionStack, we expect that one view of the overall mission steps and their internal dependence on each other will be of central interest.

We have looked at different methodologies for defining the representation of the mission that appears in this plane, from custom representations to standardized BPMN representations. It is key for this representation to show the mission at an abstract level, so that it is easily and immediately understood. Details of the mission can be pushed down to other planes, where they can be better tracked closer to the personnel responsible for them.

III. Critical Information Objects

The critical information objects are keys data that are processed and pushed through the mission representation in the Mission Plane. These objects are the drivers that determine the completeness of mission steps, and form the triggers that cause new mission steps to commence.

In many cases (such as in this example) the dependency between information objects is simply temporal, and is represented by the left-to-right ordering of the objects in the plane. Other, more complicated dependencies can be described in the plane using containment to show part-of relationships, or arrow links to indicate additional dependencies.

IV. Abstract Services

The Abstract Services Plane shows the services necessary to create the critical information objects - and therefore, the services needed to accomplish the mission - at an abstract level. This level is an important part of the process, because it is often the level at which network defenders think of the network environment. The nodes represented at this level indicate the key competencies and relationships between groups or components necessary to complete a mission step. In the example shown in Figure **??**, there is a key relationship between "USSTRATCOM Test planning" and "Vandenberg Test Planning" that will be related to object in planes both above and below it. However, this level is important because it is at a sufficiently abstract level as to permit several different implementations at lower levels. The flexibility permits operators significant latitude in determining how the service will be implemented, without requiring changes to the upper levels of the MissionStack. This feature keeps the upper levels consistent and slowly changing, which makes them excellent representations of the context of the mission.

V. Service Components

The Service Component level shows the components necessary to implement the abstract services presented above. Although the elements represented at level are more concrete than those indicated at the Abstract Services level, the elements at this level do not represent specific hardware. The elements represented at this level are the services and network features selected to implement one possible solution to the requirements for the Abstract Services. Other solutions may also be possible, and one of the extensions planned for the elements represented at this level is to show these alternatives and compare their simulated performance against the performance of the selected solution. This type of comparison could be very useful in quickly determining options in the face of attacks or network failures.

VI. Physical Components

The lowest level in this example are the physical components used to implement the services represented at the Service Component level. For example, the "USSTRATCOM FTP" service represented at the Service Components plane may be protected by a Operating System Diversity Defense, where the OS underlying the FTP service is changed every 2 minutes. In this case, the "USSTRATCOM FTP" element would have connections to the set of computers used as part of the OS Diversity Defense.

This level also indicates the routers, firewalls, and other network services (i.e., DNS, DHCP) that are used as part of normal operations. If any of these services are attacked or brought down, the dependencies of higher elements on them will be visually represented in the Mission Stack.

VII. Visualizing Dependencies

As the flow of information proceeds from the start of the mission, the links between planes indicate the status of mission-related dependencies and the links within the planes indicate process-related dependencies.

If network attacks or equipment failures occur, conditions will be triggered that change the state of the dependency lines drawn to indicate both the error and the level of risk to the mission of that error. For example, a particular Vandenberg FTP server failing may add slight risk to the "Vandenberg FTP Service", but since that service is utilizing multiple servers to provide FTP services, there is little risk to the overall mission because of that failure.

VIII. Summary and Future Work

Our work on the MissionStack is proceeding in several areas:

- Plane Definition: We are working to better define the semantics of the planes, and to determine new planes to add to the stack.
- Dependency Specification: We are working on methods and representations for specifying dependencies between planes, and to determine when those dependencies are at risk.
- User interaction: We are investigating and prototyping different methods of user interaction with the MissionStack, including gestural and hierarchical controls.
- Embedding in larger interfaces: We are investigating how the MissionStack may play a role in larger interfaces, which we describe below.

A. Integration in Larger Interfaces

Figure 3 shows an interface concept that uses one main MissionStack to denote current operations, with a set of small multiples of MissionStacks represented performance in the past, and projecting performance into the future. With this type of interface, the dependencies represented as lines between planes are extended to show which mission steps are at risk, and at one point of time in the future that risk changes from acceptable to unacceptable.

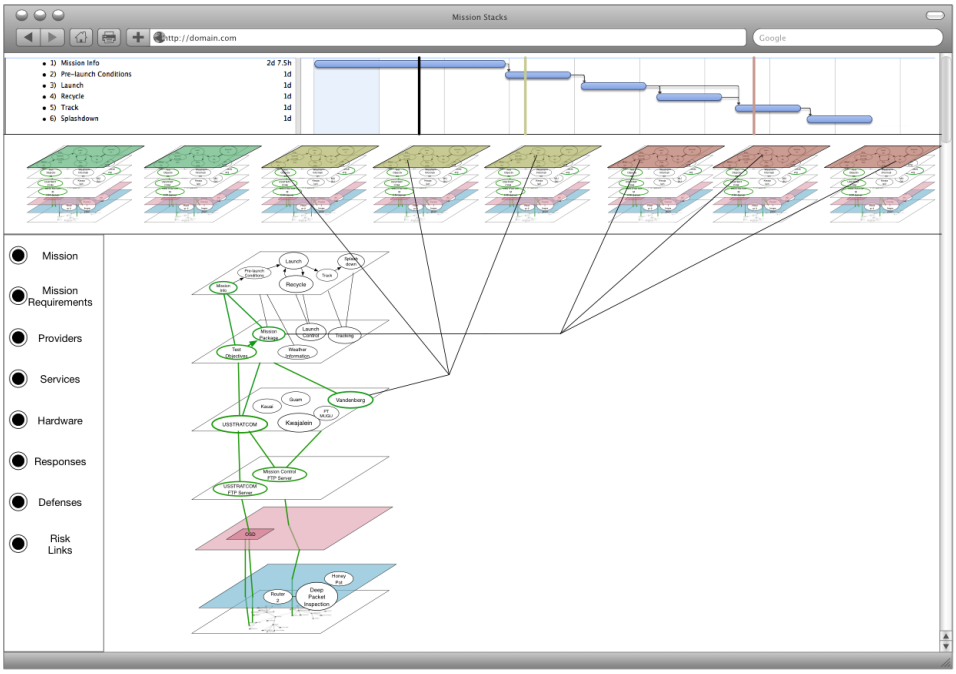


Fig. 3. Envisioned interface shows stack states past, present, and in the predicted future.

ACKNOWLEDGMENT

This research project is sponsored by the U.S. Department of Defense. Any opinions, findings and conclusions or recommendations presented in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense.

REFERENCES

[1] S. Jajodia, P. Liu, V. Swarup, and C. Wang, editors. *Cyber Situational Awareness - Issues and Research*, volume 46 of *Advances in Information Security*. Springer, 2010.