SitaVis - Interactive Situation Awareness Visualization of large datasets

Francis C. B. Williams† William J. Faithfull‡ Jonathan C. Roberts‡

School of Computer Science
Bangor University

ABSTRACT
We present SitaVis, a visualization and situational awareness tool for the analysis of the health of a computer network. Network datasets are large and therefore we have developed an interactive framework that enables the dynamic exploration and interactive analysis of the data using aggregation techniques and Microsoft's XNA framework. Machine health data is queried and analysed through visual and direct manipulation of our visualizations. SitaVis includes stacked area graphs, dense pixel plots and geographic representations of the data, and offers situational awareness of large geographically dispersed networks. This paper describes the development and use of the SitaVis that was applied to the VAST 2012 mini-challenge 1, cyber situation awareness data.

Keywords: Visual Analytics, VAST 2012, XNA runtime environment, information visualization, Network Security data

1 INTRODUCTION
The VAST 2012 challenge was to investigate the cyber situational awareness of a fictitious international bank called the ‘Bank of Money’. Two datasets were provided for the first challenge, the first contained meta-data that describes the computer network of the bank. This consisted of nearly 1 million individual machines divided over 4,056 geographically dispersed facilities which included datacenters, regional offices and branches. The second dataset contained the output of status updates for all computers on the banks network for two days. The status of each machine is stored every 15 minutes and provides details about deviations from corporate policies and the current activity for each individual machine. When stored as comma-separated-values the main dataset is approximately 8GB.

Our aim was to visually analyze this data interactively. Consequently a significant portion of our development time was directed to develop an infrastructure that would allow the system to dynamically query the data in real time. Another challenge is that there are many events occurring at the same spatial location. Visually we could have a problem of overlapping points, so we chose to aggregate the data (and visualize summary information) and allow users to interactively drill-down into the data, to discover details about the data.

SitaVis offers a visual overview of the network with the ability to drill down into the data using direct interaction techniques. We plot the facilities by location and allow the user to pan and zoom over the data, inspect facilities by hovering over them, analyze temporally the changes in both policy status and activity flag through the use of interactive stacked area graphs and filter the data using a menu system

2 DEVELOPMENT PROCESS
The development of SitaVis was split into three distinct phases, database development, development of the visualization framework, and the development of the visualizations for SitaVis.

2.1 Database Development
The implementation of a suitable data storage mechanism was vital to this project primarily due the size of the data involved. We used a MySQL relational database, not only because the dataset consisted of relational data that suited the structure, but also due to time constraints, since we had developed other interactive visualization tools using this methodology our development time was reduced.

The data features were pre-analyzed and separated into a relational structure that consisted of seven tables (4 main tables and 3 meta tables). The main tables were ‘facilities’ and ‘machines’ which described the individual facilities and computers respectively, and ‘machineStatus’ and ‘facilityStatus’ which described the individual and aggregated status updates for the machines. Several optimizations were applied to the database, including (1) columns that were likely to be regularly queried (including time stamps or status descriptions) were mapped against integers (for fast comparison), (2) the query cache of the DBMS was increased to a total of 21GB, and (3) queries were threaded.

2.2 Visualization Framework
To make the system interactive we realized that we needed to optimize the rendering and re-draw operations. On previous projects we have developed visual analytics tools using Java, Processing.org, OpenGL and Cuda. For SitaVis we wanted a development environment that was quick to develop and fast to run. Microsoft’s XNA framework is designed for game development and enables code reuse across different target platforms. It provides the low-level access that was required to develop visualizations that render at a fast speed. The challenge though is that XNA is focused on 3D graphics, and therefore offers no existing architecture for the creation of 2D graphics such as polygons, splines fills and strokes. Consequently, a rudimentary 2D library was developed (using XNA in C#) specifically for this project that contained classes for polygons, user interface elements, lines, strokes and graphs that supported the development of SitaVis.

2.3 Visualizations
Several visualizations were developed that are coordinated together: including stacked area graphs to show distributions over time, a geographic display and a ‘reticle’ visualization that integrates histograms and dense pixel plots. It is often difficult to view the whole network in such visualizations [2], and therefore in SitaVis we use several techniques. We aggregate the data, summarize the information on the map, and also use stacked area graphs to provide a summary over time. SitaVis is shown in Figure 1 and highlights several facilities which include datacenters and regional headquarters in red. This indicates that these facilities have at least one machine registering a policy status at level 5 (possible virus infection or questionable files found).
The **Stacked Area Graphs** (shown in Figure 1b and c, and Figure 3) can be used to summarize the whole data, or specific regions. When looking at the complete VAST 2012 challenge data, it is clear to see that there is a general worsening trend towards more critical policy status', and the periodicity of days and nights can be viewed. However, it is more useful when applied to individual regions. In particular, it is possible to identify that there is an unusually high quantity of offline machines in region 25 (Figure 3), which is possibly due to a large power cut in this region.

![Figure 3: Policy status counts by status type as a stacked area graph. Time on the x-axis, count of each policy status on the y-axis.](image)

The **Reticule visualization** is a novel view that integrates dense-pixel plots with histograms (Figure 2). The reticle view acts as a telescope on the data. It is triangular in form with dense-pixels of individual machines listed by order, and a histogram that demonstrates the distribution of policy status. The reticle is used to display the status of an individual facility and shows the status of all machines in each of the three categories (Server, Workstation, ATM). We found the dense-pixel plots useful to display an overview; other systems such as PortVis [3] visualize the data in a similar way. Other visualization methods for network security are reviewed by Shiravi et al. [4].

The **map view** plots all the facilities. We pre-calculate statistical information of the data on the facilities and use this summary information to provide the color of the facility. For instance, we can choose to color the locations on the map by the ‘worst machine status’, such that if merely one machine of a datacenter has been compromised then the color is set to red. It is therefore quick to eye-ball the map to locate more serious events.

### 3 Conclusions

SitaVis provides a situation-awareness of the VAST 2012 challenge data. It was designed to produce a holistic view of the whole data, whilst offering the user context and still allowing for real-time interaction. The use of the map view and temporal stack graphs enable users to see an overview of the network at a glance. Users can drill down into specific datacenters and facilities through zooming and querying. The reticle display provides an overview of all machines at a facility and helps visualize different types of attacks. The data is displayed at interactive speeds on our Core i7 3.4Ghz machine with 24GB of RAM and an Nvidia GTX 580 graphics card. This has been possible with the use of XNA and the optimizations of the MySQL database.

There are many ideas for future development. We are investigating GPU programming to develop faster data processing and the use of HPC facilities to provide faster analysis. Finally, it would be beneficial to provide comparative visualizations [1] on the data of two (or more) facilities, and also investigate and compare the data temporally, such to ascertain how events progress geographically across the network.

### References


