Info Vis

Article

Information Visualization
2015, Vol. 14(1) 62–75

DOI: 10.1177/1473871613488573
ivi.sagepub.com
SAGE

Multi-aspect visual analytics on large-scale high-dimensional cyber security data

Victor Y Chen[1], Ahmad M Razip[2], Sungahn Ko[2], Cheryl Z Qian[3] and David S Ebert[2]

Abstract

In this article, we present a visual analytics system, SemanticPrism, which aims to analyze large-scale high-dimensional cyber security datasets containing logs of a million computers. SemanticPrism visualizes the data from three different perspectives: spatiotemporal distribution, overall temporal trends, and pixel-based IP (Internet Protocol) address blocks. With each perspective, we use semantic zooming to present more detailed information. The interlinked visualizations and multiple levels of detail allow us to detect unexpected changes taking place in different dimensions of the data and to identify potential anomalies in the network. After comparing our approach to other submissions, we outline potential paths for future improvement.

Keywords

Interactive visual analytics, semantic zooming, pixel oriented, multivariate visualization, geospatial analysis, interaction design

Introduction

We designed and developed a visual analytics (VA) system "SemanticPrism" to address the large-scale, high-dimensional cyber situation awareness problem arisen by the VAST 2012 Mini-Challenge 1.[1] The challenge is a "big data" problem. A large enterprise network, named the Bank of Money with approximately 1 million machines, generated approximately 160 million multidimensional data logs (e.g. geographic location, time, activity, policy, machine class/function, and number of network connections) in 2 days. For proper analysis, the analyst must be able to see and compare all of these different dimensions at multiple granularities (e.g. enterprise to individual machines in individual offices). To meet these requirements, we developed the VA system SemanticPrism to visually analyze the given data from three perspectives: spatiotemporal distribution of machines and their health, overall temporal trends, and pixel-based IP blocks. All these visualizations are interlinked and provide 2–4 levels of semantic zooming. The analyst can not only grasp the overall situation of the enterprise network, but also drill down to read more detailed information of regions, offices, and even the level of individual computers. With SemanticPrism's comprehensive visualizations and interaction, we were able to discover all anomalies hidden within the large dataset and won the award of "Outstanding Integrated Analysis and Visualization."

While designing the SemanticPrism, rather than simply solving this particular challenge, we tried to explore a more general approach to face real-life large-scale high-dimensional datasets. In this article, we

[1]Department of Computer Graphics Technology, Purdue University, West Lafayette, IN, USA
[2]Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
[3]Department of Art and Design, Purdue University, West Lafayette, IN, USA

Corresponding author:
Victor Y Chen, Department of Computer Graphics Technology, Purdue University, West Lafayette, IN 47907, USA.
Email: victorchen@purdue.edu

review related literature and then discuss the design considerations of SemanticPrism in terms of scalability, dynamic situation awareness, visualization, and novelty. Furthermore, we compare our approach to other submissions and outline potential paths for future improvement.

Previous research

Previous study has explored a variety of approaches to handling the problems "big data" create. As "big data" are often complex, multidimensional, and multivariate, many studies have discussed different methods of visualizing large datasets. Fua et al.,[2] for example, described the use of hierarchical parallel coordinates to visualize large multivariate datasets. Keim[3] and Oelke et al.[4] have shown the use of pixel-based visualization to fit the huge data space into a small screen space. Keim et al.[5] also developed a hybrid technique that is scalable with "big data" visualization. Some approaches also include using multiple linked views to visualize "big data."[6] A summary of recent visualization techniques of large multivariate datasets is available by Keim et al.[7] Looking beyond the academic community, a number of commercial VA systems that process big data already exist within the marketplace and have been evaluated.[8]

We believe that analyzing this high-dimensional data requires multiple linked views from different perspectives and different levels of granularities and detail. Zoomable user interfaces (ZUIs)[9] allow users to work with a large virtual space and navigate through it by zooming. Semantic zoom[10] lets the user see different representations of the data at different zoom levels. Weaver[11] stated that "semantic zoom is a form of details on demand that lets the user see different amounts of detail in a view by zooming in and out." This method has been widely used to provide smooth analysis experiences through interaction (such as in malicious network objects[12] and relational data[13]). We utilize semantic zoom in this system to visualize the different properties of big data at different granularities.

As the number of connected machines and the possibility of network attacks increased, many experts have developed various network monitoring tools that include a number of different visualization techniques. One of the more popular methods of visualizing network data is to use graph-oriented visualizations[14] where machines are mapped to nodes and links connecting those nodes with different characteristics, such as thickness and color, represent relations among nodes. Boschetti et al.,[15] for example, implemented a graph-oriented approach to monitor network traces and detect anomaly. Iliofotou et al.[16] proposed the use of traffic dispersion graphs (TDGs) as a way to monitor and analyze network traffic by modeling the social behavior of hosts. In many systems, different types of node placement algorithms have been used. Among those, the force-directed graph drawing method[17] and bipartite algorithms[18,19] have been widely adapted.[20,21]

A different visualization approach is the pixel-based visualization approach, and it provides some advantages over the traditional graph-oriented visualization of the computer network. Oelke et al.[4] and Keim[3] introduced the pixel-based (pixel-oriented) visualization technique to maximize the screen space for visualizing large amount of data. In this technique, the entire visualization space is equally divided into squares or rectangles, called pixels, where each data element is assigned. Then, a predefined color map is applied to represent the range of the data attributes. The pixel-based visualization technique has been widely used in various applications and research in which the datasets are very large and multivariate. Borgo et al.[22] presented how the usability of the pixel-based visualization varies over different tasks and block resolutions. Oelke et al.[4] studied visual boosting techniques for pixel-based visualization such as halos and distortion. Ziegler et al.[23] presented how the pixel-based visualization helps analysts gain insight for long-term investments. Ko et al.[24] demonstrated how sales pixel matrices can be used for analyzing competitive advantages of companies. Panse et al.[25] discussed the effectiveness of the technique in PixelMap when the datasets consist of very large number of points. In our study, we employ the pixel-based visualization method to explore the IP address space of the challenge data.

According to MacEachren and Kraak,[26] geospatial datasets are fundamentally different from other kinds of information in at least three ways: structured spatial variables, meaningful location names, and emergent behaviors. To visualize the special geospatial aspects of the data, the popular technique with many geographical information systems (GIS) is applied in order to plot points on a geographical map. This technique has been employed in a variety of domains, such as crime mapping,[27] public health,[28] and social science.[29] Other techniques to visualize large spatial datasets include PixelMaps,[5] which is designed to combine both clustering and pixel-based visualization to plot points on the map. This technique copes well with dense geographic data and prevents data point overlaps. In this study, we used the point-plotting technique that has more expressive power of identifying individual offices inside a region.

SemanticPrism system

Although the data provided for this challenge are artificially generated, this challenge simulates a real scenario. The SemanticPrism system was designed from the beginning, not only to solve this particular

challenge, but rather explore a general approach to achieve cyber situation awareness in a real-life scenario while facing large-scale multidimensional datasets.

Dataset and tasks

The data from the challenge include a geographical map (an image file), a KML (Keyhole Markup Language, an XML notation for expressing geographic data) data file to define regions, and two large spreadsheet tables. One table contains basic information of all computers, including its IP address, business unit, facility, latitude and longitude, machine class (server, automated teller machine (ATM), or workstation), and machine function. The other table contains 160 million records of computer status logs. Each record contains information of a computer's IP address, number of connections (NOCs), policy status, activity flag, and the log timestamp. The policy violation status is a discrete data measurement of the health status from normal to severe (labeled from 1 to 5 to indicate severity). The activity flag categorizes the machines by different types of activities (labeled 1–5). The NOC log is a discrete data value (range from 1 to 100 in the current dataset). The data provide a 2-day snapshot of the health status of all computers in the whole organization with 15-min intervals (192 total time periods).

We first brainstormed several fundamental tasks and their consequent data queries based on the need of cyber security situational awareness: (1) See the geospatial distribution of computers and tell whether there exists any spatiotemporal status pattern. This task requires the system to query computer logs grouped by offices and time periods. (2) Visualize the trends of computer status at different granularities from the overall network to individual machines. The system has to count the numbers of computers of different statuses over time. Also, the NOCs need to be aggregated to get the maximum and average values. (3) Study the spatiotemporal pattern of status over the IP address space. The challenge data do not provide the structure of the network. In the Internet, computers' IP addresses can be classified as Class A–C based on their four 8-bit numbers. Such classes partially reflect the network structure. Computers within the same block (especially a Class C block) are likely to be in one subnetwork. Data also need to be aggregated based on Class C IP blocks. (4) Investigate individual computers through their log history. It requires searching the full history logs of a computer through its office or IP block. The data should be indexed by the computers' offices and IP blocks to speed up data query.

Data transformation and aggregation

Our first challenge was to transform the large-scale data and make it efficient for interactive analysis. Even when using a MySQL database, the direct querying of such a large dataset remains inefficient and can take hours to provide an aggregation number (e.g. the total number of computers of a given status). To speed up the data query and enable a responsive system performance, we created additional indices and aggregated data into new tables. The process to treat the data as a data cube, precompute the aggregation values along necessary dimensions, and store the aggregated values into several tables is in line with the online analytical processing (OLAP) approach.[30] Precomputed aggregated values include the number of computers for each policy status and activity and the maximum and average NOC at a given time for each Class C IP blocks. Querying and processing the data for a group of time series curves only take a small fraction of 1 s; so, most interactions in the system can produce instant results.

For this challenge, the data provided are a static file, meaning aggregation is only done once. In a real-life implementation, such an aggregation and preprocessing could be performed while collecting data on the fly.

System structure and development platform

Although the raw data from the challenge only cover 2 days and are 8 GB in size, in essence, these data are streamed and can eventually become truly big data as time goes on. In order to correctly and effectively manage these big data, SemanticPrism uses a client–server architecture designed as a web application. Clients in the front end visualize the data but do not retain a copy of the whole dataset.

As a prototyping system for research purposes, we built the system with Adobe Flash, PHP, and MySQL. The client-side application uses Flash that is currently supported by most web browsers and is an efficient platform for an application with rich interactions and dynamic graphics.[31] The web server runs PHP to process data requests. The communication between Flash and PHP is done through action message format (AMF). The PHP web application is hosted in a shared server. The MySQL database server resides in an Intel Xeon 3.0-GHz server, with 64 GB of memory. The Flash client can run smoothly on a notebook computer (Intel 2.6 GHz Core 2 Duo CPU with 4GB RAM).

Visualization and interaction design

The choice of visualization and interaction design should be based on the nature of data and the problems faced. We wanted SemanticPrism to run on a notebook computer to allow maximum freedom of working location. With limited screen space, the analyst should be able to navigate through different dimensions of data, drill down to investigate details, become aware of significant changes, and identify anomalies. To

enable exploration of the data from different data dimensions, we chose to use a multiple linked views approach: different types of information are visualized using the geospatial map, time series curves, and pixel-oriented visualization views. Each of the views has multiple visualizations to present different levels of details. We chose semantic zoom as the basic interaction technique to navigate through these visualizations.

Geospatial–temporal visualizations

The default view of SemanticPrism is a geographic visualization with a time slider designed to visualize the computer status at a given time (Figure 1).

Offices around BankWorld are marked on the map as square dots. Different icons are used to distinguish types of offices: small squares represent regular branch offices, squares with one boundary line represent the regional headquarters, and squares with two boundary lines represent headquarters and data centers. The map provides an overview of the most critical information at the current time. Different colors are applied onto the squares to indicate the maximum policy violation status of the computers within the office at that particular time. The colors, varying from yellow to orange to dark red, represent the maximum policy violation status (from 1 to 5) for all computers in the office. The reason the most severe computer health is shown, instead of the average health status, is to draw the analyst's attention to a problem the very first moment the problem arises. This color setting is consistent in representing policy status across different visualizations and functions in SemanticPrism. If there is no log from an office at a certain time, it means that all computers are off-line. In this case, the office is represented by the black color.

To update the statuses of all offices to a different time period, the analyst can drag and slide the time slider to a new time mark along the bar (Figure 1). Additionally, the analyst can input the desired time (period number, ranging from 1 to 192) in the time input slot or use the time step forward/backward button to advance to the next time slot or roll back to the previous slot.

Layers on the map

The SemanticPrism map (Figure 1) uses several layers to stack different dimensions of information together. The analyst can selectively turn on or off these layers.

The dataset's Bank of Money is a global organization spanning eight time zones of BankWorld. In order to present the different time zones and local times for the distributed offices, we provide an overlay time zone layer. It is a half-transparent layer with vertical strips of gray shades. Time zones within the early morning or late night are in darker shades to hint there is less sunlight. Although at night machines tend to be less active, we also wanted to draw the analysts' attention to the fact that some crucial attacks might take place during such time periods.

When there are many offices in a relatively small area, the area can be cluttered with many dots of offices. To improve that condition, we created two layers to highlight offices containing computers with a selected policy status or activity status. The analyst can select a policy from the policy drop-down menu to turn on the policy layer. With one policy selected, any office with a computer that belongs to this selected policy status will be highlighted as a blinking red/blue square. The size of the blinking square reflects the number of computers with the selected policy status. The blinking effect is good at drawing the analyst's attention even if the dot is small. Similarly, the analyst can turn on the activity layer through the activity menu (Figure 1(b)). The system uses orange/green blinking squares to highlight all offices that have computers with the selected activity. Correspondingly, the sizes of the squares reflect the numbers of computers that are involved in such an activity. As time progresses, the change in size and location of blinking squares indicates to the analyst the trends of the policy status and activity.

If both blinking layers are turned on, the blinking squares become messy and hard to read. To effectively reduce the visual clutter and visualize the policy status and activities together, we use a Kernel density estimation (KDE)[32] heat map as an alternative method to visualize the geospatial distribution of a selected policy status or an activity. The heat map is computed based on the density of computers matching the selected status in an area using a clustering algorithm. The heat map uses blue shades: the darker the shade, the more computers there are that match the selected status. This layer is stacked under the blinking layer, as shown in Figure 1(a). With the combination of the two layers (KDE and blinking), the analyst can read the policy status and activity at the same time.

Zoom and navigate

Through the right side's navigation panel, the analyst can zoom in/out and pan to navigate in the map. A red square shade (right corner of Figure 2(b) and (c)) is used in the panel to indicate current visible area of the map. While zooming in, the space among office dots increases, which can potentially be used to display more information.

In SemanticPrism, the analyst can drill down and investigate the data at different levels of detail (Figure 2) through semantic zooming.[10] Depending

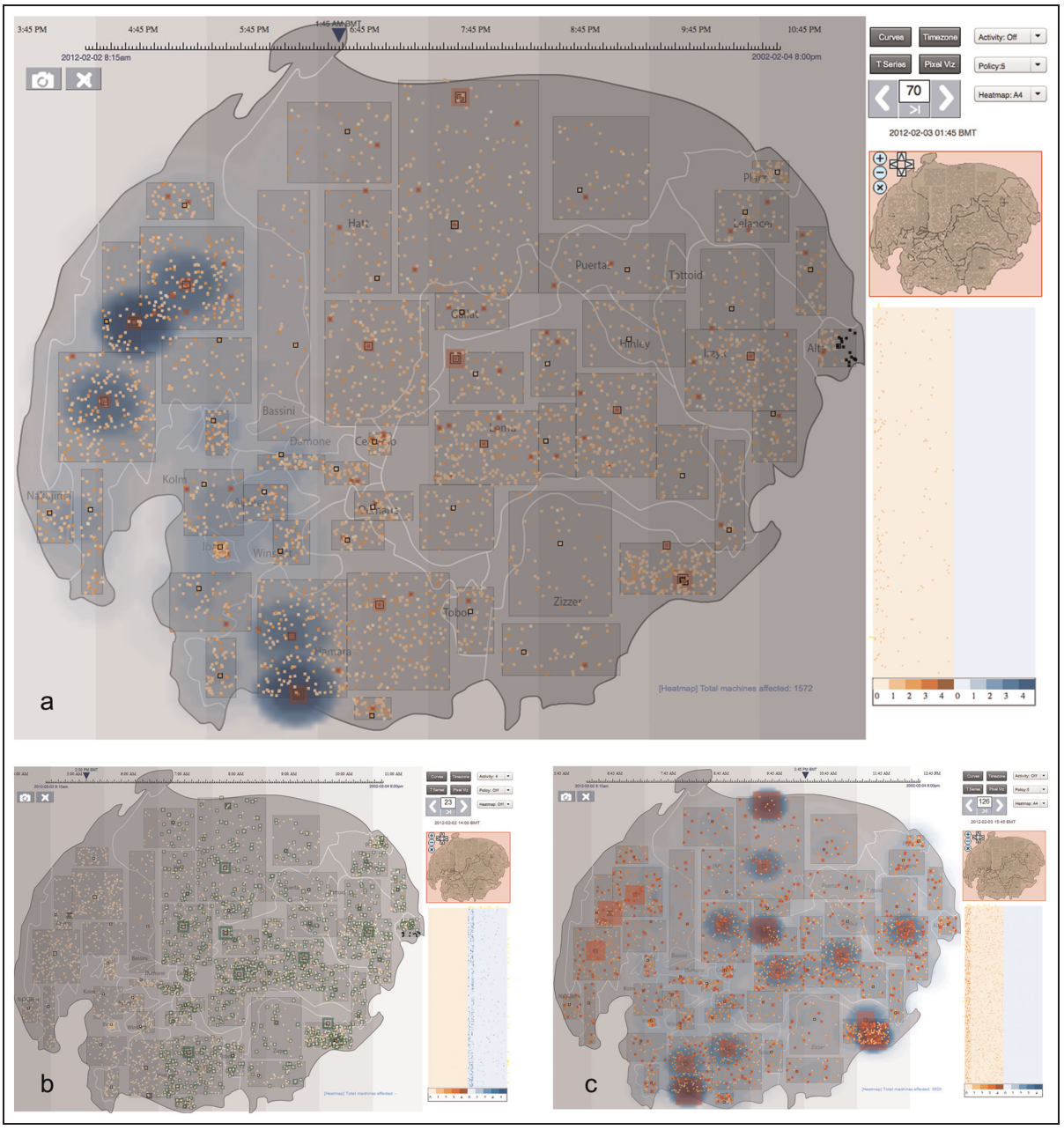

Figure 1. SemanticPrism map view: (a) the policy layer with the activity heat map at 1:45 a.m. BMT, 2 March 2012; (b) with only the activity layer on at 2 p.m. BMT, 2 February 2012; and (c) the policy layer with the activity heat map at 3:45 p.m. BMT, 2 March 2012).

on the size of available space, an office is dynamically visualized in one of the following four levels:

Level 1 allows the analyst to visualize an office as an individual dot when using the default full map view or when the space is still quite dense after zooming in (Figure 2(a)). In the full map view, the offices at some areas appear so dense that the square dots may overlap. While zooming in, the squares are also enlarged, but at a smaller ratio (square root of the screen ratio), to make the spaces among the offices larger. Thus, the density decreases and provides better readability to distinguish these little office dots.

Level 2 (Figure 2(b)) uses a horizontal color bar to show the percentage of computers with different policy statuses, including those that are currently off-line. In this visualization, the analyst may misread those policy statuses with only a very small percentage of computers and assume that they do not exist. To avoid this problem, we used vertical squares to mark if

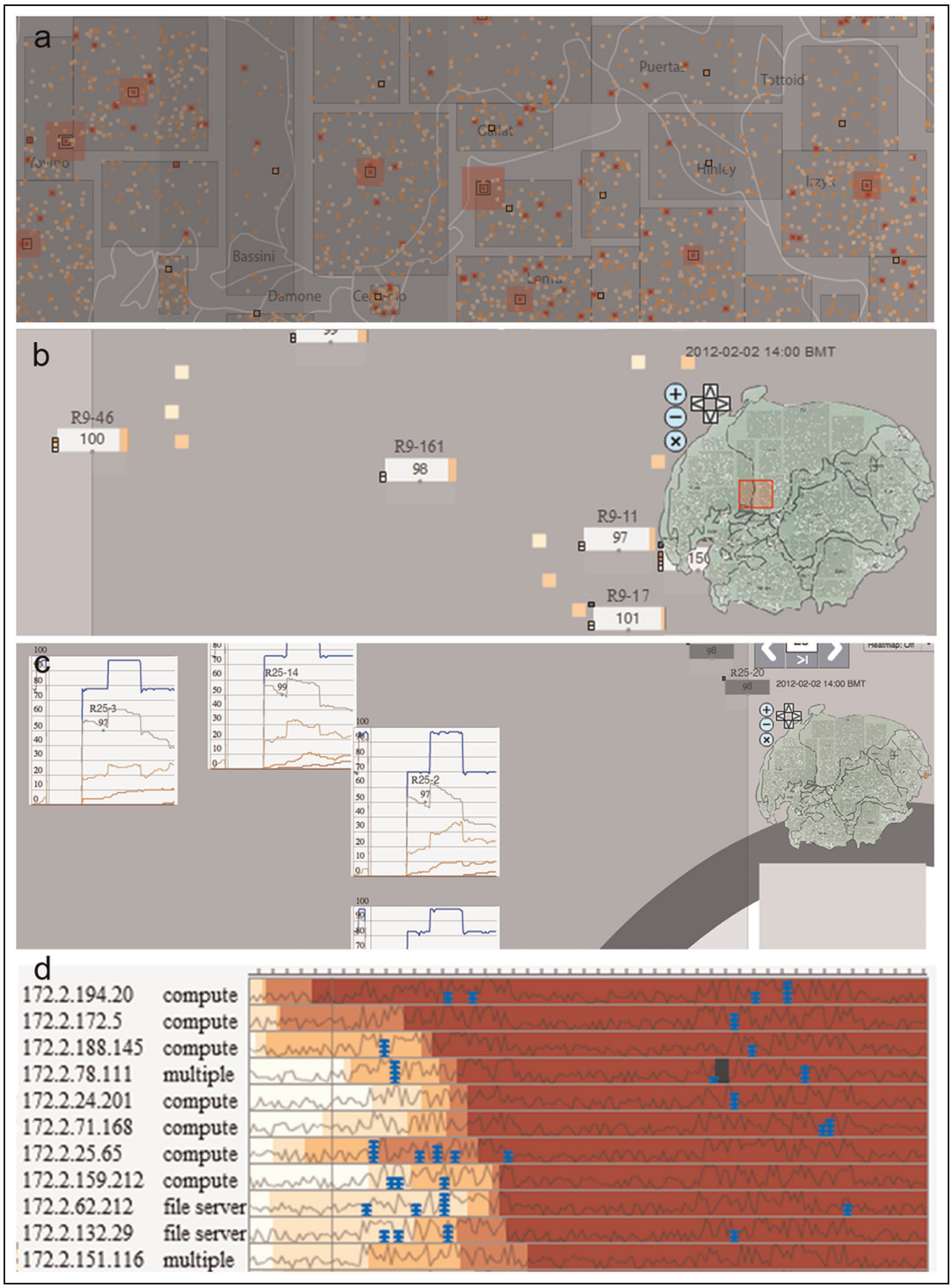

Figure 2. Four levels of semantic zooming on the map. (a) level 1 - offices as dots (b) level 2 - offices as bars to show percentages of problematic computers (c) level 3 - curves to show trends of problems (d) level 4 - history of every computer in one office.

computers with certain policy statuses exist. Also, the analyst can see the office name and total number of computers in the office. At this level, the size of the bar remains the same during zoom in until the space is big enough to show Level 3 details.

Level 3 (Figure 2(c)) indicates the growth curves of all policies in the office where the *x*-axis presents the temporal direction, and the *y*-axis shows the number of computers. This graph contains six different curves, displaying numbers from Policy 1 to Policy 5, and the

total numbers of online computers. The standard policy colors are used to distinguish different curves. The size of the graph (200 pixels × 200 pixels) remains the same when zooming in.

Level 4 illustrates the history status of each individual computer within the office (Figure 2(d)). The history of a computer's policy status is visualized as shades of a red bar. The curve in the middle of the bar shows the NOCs. The computer's activities are visualized as blue bars with stacked horizontal lines. The number of lines represents the activity number. Activity 1 (normal status) is omitted. The analyst can use this visualization to read the finest details of a specific computer in a specific office.

Zooming in/out and panning change the screen dramatically. When the analyst's eyes are focused on one area and there is a sudden change in the visualization, change blindness[33] might take place. The analyst may lose his focus. To avoid that, we integrated animation to permit a more gradual zoom in/out and allow saccadic eye movements[34] to catch up with the changes. Zooming out creates a reverse effect of zooming in. Detailed views will be shrunk until the offices become square dots.

Apart from using the navigation panel, the analyst can directly interact with the map to pan and zoom in/out. Scrolling the mouse's middle wheel zoom in/out of the map. A left mouse drag pans the view. Clicking on an office will directly open Level 4 details of an office. Clicking on the boundary of a region will open the pixel-based visualization of all offices within that region. These offices are laid out in a rectangular array to let the analyst see all offices simultaneously (Figure 4(b)).

Time series curves. While designing the system's information query process, we followed Shneiderman's information-seeking mantra:[35] "overview first, zoom and filter, then detail-on-demand." The SemanticPrism time series curves (Figure 3) provide an overview of the growth trends of policy statuses, activities, and NOCs over the given time period. The default curve view (Figure 3(a)) presents the growth of policy statuses and activities of one class of computers. The analyst can choose to use either a linear or logarithmic scale to draw the curve. The linear scale can intuitively show the overall growth trend. But because of the large number of overall computers, it is hard to read the curve at the early phase of an attack when there are only a few computers affected. The logarithmic scale (Figure 3(b)) boosts the small numbers by adjusting the curves to help the analyst to catch that first moment when a computer is violating a policy.

The time series curve visualization can also be "zoomed in" conceptually. The analyst can narrow down by applying a combination of filters to select certain computer class, computer functions, activities, policy statuses, and NOC to visualize the trends of affected computers.

In SemanticPrism, the user can also create multiple panels (Figure 3(c)), with each containing curves generated by different filters. The analyst can then compare different curves side by side to investigate further.

Pixel-based visualizations. The classification of IP addresses can partially reflect the organization's network structure. Within these data, we also noted that computers within a single office with one class (server/workstation/ATM) belong to the same level Class C. To visualize such an IP address space, we incorporated a pixel-based visualization of IP blocks (Figure 4) to analyze computers in a more detailed classification.

By default, the pixel-based visualization contains one panel showing the selected policy status and activity (the red/blue square on the right side of Figure 1). The selection is done through the same drop-down menu of highlighting policy status and activity. The analyst can expand it to show five panels. Each panel shows the number of computers within an IP block that are affected by each activity and policy (Figure 4(a)). In each of these five panels, the red side shows policy status and the blue side is for activity. Each pixel represents a group of computers in a particular Class C block. The x-axis consists of the IP's Class B block (ranging from 172.1 to 172.56), and the y-axis consists of the values of Class C blocks (ranging from 0 to 255). The colors of the pixels encode the number of computers that carry the selected policy status or activity flags in the C block.

The IP block pixel-based visualization has three levels of semantic zooming. Hovering the mouse pointer over the inside area of the panel will evoke a zoom-in lens to show enlarged pixels. Clicking of a pixel brings up the time series curve of that C block. Clicking on the bottom x-axis of the panel will evoke the system to show the time series curves of all C blocks within the selected B block. The user can choose to see the curves of policy statuses (Figure 4(b)), activities, or NOCs (Figure 4(c)). The analyst can also see Level 4 individual computer histories in the C block (similar to Figure 2(d)) through clicking. The visualization enables the small approach to multiple comparison[36] where analysts can easily investigate the differences of trends in multiple data. For example, in Figure 4, it is easy to notice that some C blocks have an abnormal spike in the middle of the visualization, which is different than the regular on–off office hours. This leads us to investigate abnormal network connections at night.

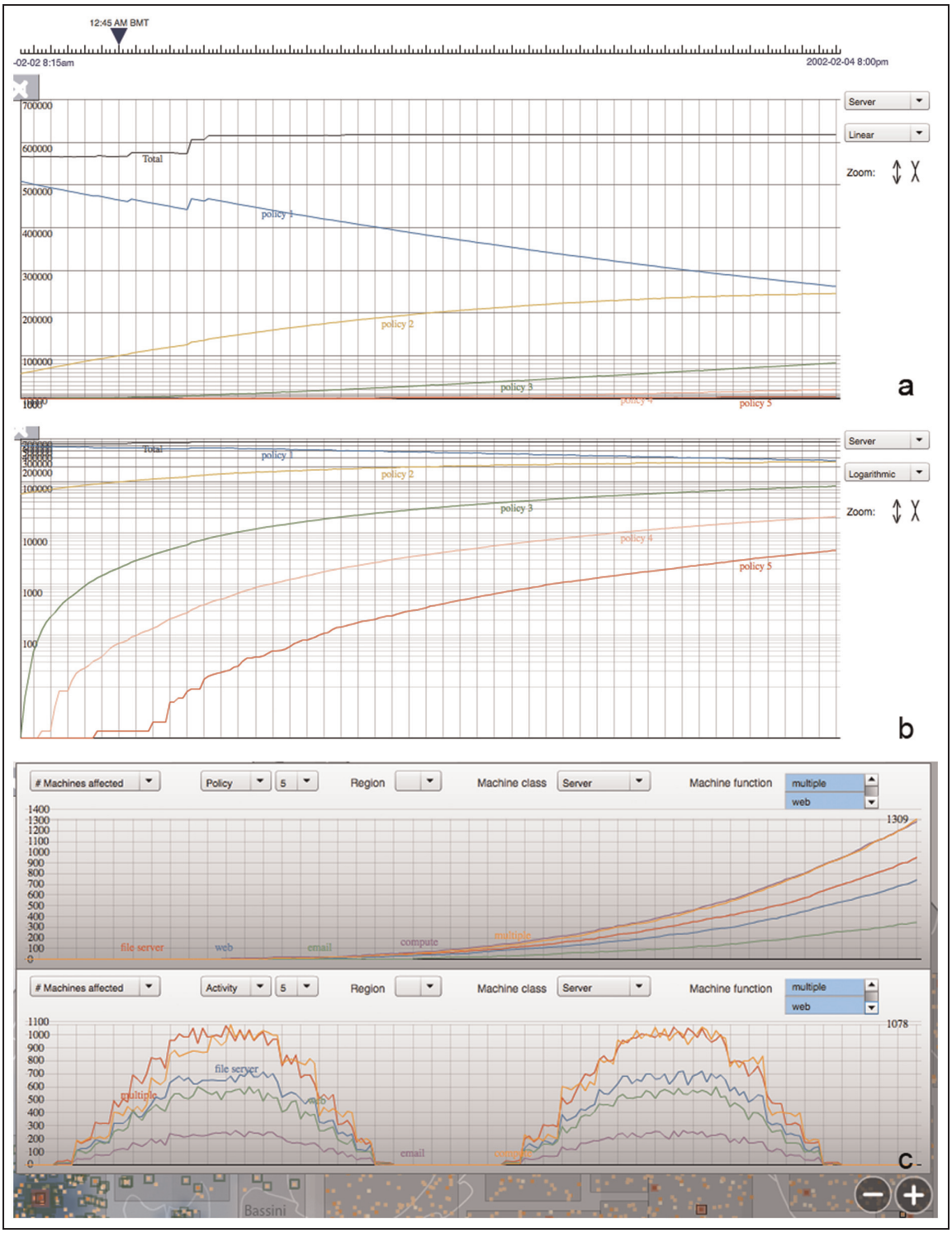

Figure 3. Time series curves in SemanticPrism: (a) linear curve of servers, (b) curves in logarithmic, and (c) dynamic panels to show the curves of policy statuses and activities of selected types of computers.

Situational awareness analysis

In a real-life context, it is essential that immediate actions be taken to prevent the expansion of damage. In SemanticPrism, each of the three visualizations has been used either individually or collectively to support situational awareness.

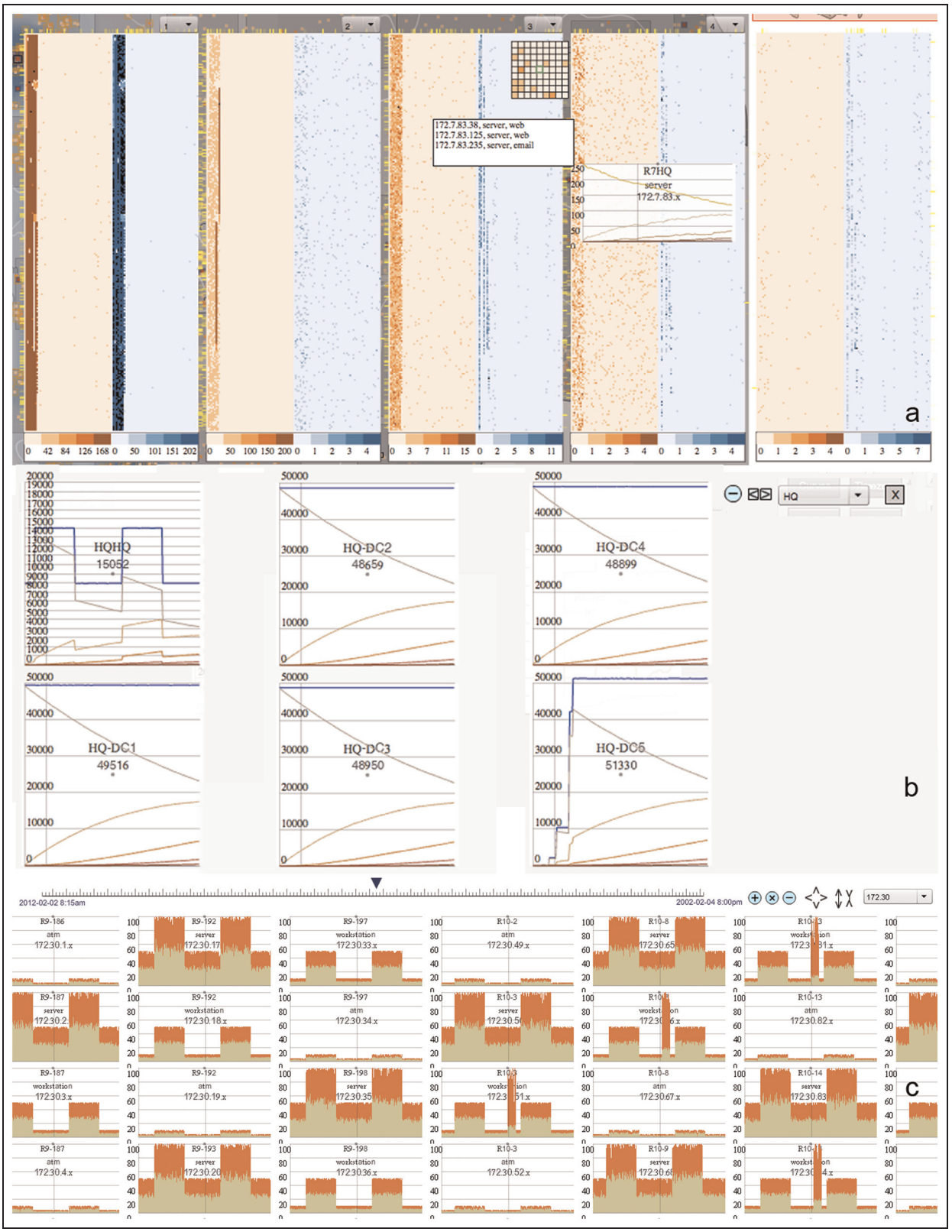

Figure 4. Pixel-based visualizations of IP blocks: (a) five default policy/activity pixel–based visualization panels, (b) offices in a region (Headquarter), and (c) zoom in to show the NOC graphs of all Class C blocks in one Class B IP block.

Detection of problematic computers at an early stage

It is critical to detect the first occurrence of a certain activity or policy violation. The time series curve view accurately presents the statuses over time, including both previous and future growth trends, though it does not pinpoint the location. With the help of the time slider, both the map view and the IP block pixel view can hint to the analyst when and where the first computer fell into policy status 4 or 5. The map view then locates the office, a perspective that is necessary if the prevention action has to be taken on-site. With both the map view and IP pixel view, the analyst can drill down to find the particular computer (IP address), a step that is necessary for remote access and repair. The quickest way to find the accurate occurrence time and location of the problematic computers is to use both the curve view and the map view: anchor the exact time of the first occurrence from the curve view and then switch to the map to find the exact office.

Although it does not appear in the 2012 challenge data, one potential threat worth monitoring is that of abnormal activities happening at off-office hours (e.g. invalid logins or intrusions). Such anomalies can be easily seen from the map by turning on the activity highlight layer and the time zone layer. As time goes on, we can clearly observe how offices become more idle when their local time slots reach the off-working time at 6 p.m. and then become active again at 8 a.m. (Figure 1(b)). The curve view can also help to monitor abnormal activities (Figure 3(c), bottom). Such activities are normally rare. The logarithmic method is useful to show even one instance of occurrence.

Overall trend of policy violations and activities

The curve view shows the growth trend of policy statuses and activities over the time period. By looking at the curves of different classes or regions, the analyst can clearly tell the growth or working pattern of the computers (Figure 3(c)). However, the analyst cannot see the spatial pattern of the spread. With the map view, as time passes, we can see that there are more and more blinking squares, and squares getting larger and larger, which indicate that there are more and more computers under high-risk policies (Figure 1(c)). With the pixel-based visualization, we can see that more and more IP blocks are affected (Figure 4(a)).

We assumed that the IP pixel view could indicate to the analyst the spread patterns of policy violation over the network (e.g. computers within the same IP block or neighboring blocks get affected first), but we have not found any good evidence for this within the current dataset.

Outages in Region 25

Among our three visualizations, the map view accurately shows that there were multiple office outages in Region 25 (black dots on the right side of the map in Figure 1(a)). With the time slider, the analyst can see the development of the outage range and how the offices then recovered (e.g. black dots in Figure 1(c)). Since this was caused by a hurricane, the accurate geospatial locations of these affected offices are useful to indicate the natural disaster's affecting range.

The IP block visualization is not effective in discovering this outage because many computers are turned off at night. The number of hurricane-affected computers does not appear to be distinct or large, so this outage has not been reflected clearly on the time series curve view.

Addition of servers to Headquarter Datacenter 5

The combined use of the time series curve view and pixel-based visualization has helped us to find that many servers were added to the Headquarter Datacenter 5. We can observe a major jump in the number of servers in the curve view (the black total curve in Figure 3(a)). However, SemanticPrism cannot directly link these data change to the individual offices. We have to manually examine offices through the pixel-based visualization. In the zoomed-in view of offices by regions, we can see that there is a significant jump of the number of servers within Datacenter 5 (Figure 4(b)).

Abnormal NOCs at night

Apart from finding the addition of new servers, the pixel-based visualization of all Class C IP blocks also helped to identify another anomaly: abnormal NOC at night. The map view did not provide a way to visualize NOC. Most computers are turned off regularly at night, so the change of NOC is not obvious on the time series curves.

Using the three visualizations to analyze the data, we came to realize that anomalies were usually identified by a combination of different visualizations. No visualization method is really universal. As different components of a system, these visualizations cover the weaknesses of each other and should be used as elements of a tool kit to detect problems in large-scale data.

Feature discussions

Considering the limitation of individual visualizations and the strength of their integration, we started to review the overall pros and cons of SemanticPrism. The following discussions are based on the comments

from reviewers of our submission and external questions raised from the VAST workshop presentation and system demonstration.

Scalability

The SemanticPrism has the potential to attack much larger data. The scale of these challenge data can be expanded in several dimensions. The first expandable dimension is time: the data can be extended to months or even years. Other possible data expansions include the addition of more offices, more computers, or more types of activity or policy statuses within the logs.

SemanticPrism's map view only displays the current status. To fit the enlarged time span, the timeline bar might be edited so that the analyst can zoom in/out and navigate semantically. The number of offices will significantly affect the map view as a result of the higher density of office dots. Theoretically, our map can only present a limited number of offices effectively. Since our system uses blinking effects to show alerts, the analyst can still easily see the problematic area even if the map has been fully covered by offices. The heat map layer is fully scalable (O(1)) since it permits the computation and visualization of the relative density of both offices and computers. Our semantic zoom technique can also work with higher office density. While zooming in, the office dots are enlarged at the square root of the screen ratio, which makes their proportions within the screen space much smaller and generates more spaces among the offices. Thus, even if there are many offices close to each other, the analyst still can distinguish each individual office by zooming in further.

The IP block pixel-based visualization will expand (O(n)) when there are more computers (more IP blocks). Since one C block contains up to 254 IP addresses, the number of IP blocks is much smaller (~1/255) than the number of computers.

The scale of time series curves follows the temporal scale as O(n) since the *x*-axis is the time dimension. We can overcome the time span problem by compressing the X direction and use interactions to zoom in and slide along the X direction to see details of the curve.

At last, the system can be converted to handle streaming data. We can change the server side component to only query data for the current moment (or within a certain period of time range), which will help to solve the large time span problem.

SemanticPrism employs layers to permit recognition of multiple properties. If there are an increased number of properties, more layers can be added, but the efficiency of readability may decrease. To improve the analysis, in the future, we might adopt a measurement that integrates multiple properties, such as the "concern level assessment (CLA),"[37] to combine multiple individual layers into one comprehensive layer.

Query data on demand

With SemanticPrism, data are stored in an external database. Only when it is necessary, the client-side Flash application can send a request to the server to fetch a small amount of data. Therefore, a significant size increase of the overall dataset will not necessarily slow down the performance of SemanticPrism. While one is working in the map view and zooming in, only the specific offices within the display area will be checked to see whether they should be expanded to display the next level of details. At Level 2 zooming, the maximum number of offices that will be displayed simultaneously in the screen is 600 (each office needs a space of 30 pixels $\times$ 60 pixels), with each office requiring just six integers for the number of computers under all policy statuses. At Level 3 zooming, the maximum number of offices in one screen is 30 (each office needs a space of 200 pixels $\times$ 200 pixels), with each office fetching data for its time series curves (an array of 192 periods with 6 flags = 1152 integers).

In the pixel-based visualization, if the screen space cannot show all of the items, the user can pan the screen to read the rest. Similarly, the client-side Flash only queries data for those elements that are within the display area.

Visualization design

One of our reviewers indicated that SemanticPrism did not invent any new visualization. The map with multiple layers, the time series curves, and the pixel-based visualization are all very common methods. However, other reviewers stated that these common visualizations are among the most suitable ways to identify anomalies in this challenge. We chose these visualizations because they are very intuitive and the analyst can easily adopt and identify the problems without taking much training.

In this challenge, one main task is to detect anomalies (e.g. virus infection) as quickly as possible. The percentage of problematic computers can be extremely low. Some traditional visualization techniques that can present both the quantity (e.g. use area or length to represent number of problem computers) and the quality (e.g. use colors to represent the policy statuses) may be inefficient to alert the analyst the problem since the size/length of the graph is too small to notice. Therefore, we used several ways to boost the visualizations. In the time series curves, the logarithmic way of drawing the curve boosts small numbers. In the geospatial map, the blinking dots draw the analyst's attention even when the dots are tiny. At Level 2 of office

details, we used separate icons to indicate the existence of different policy statuses, as well as using length to represent the percentage of each policy status.

Adobe Flash as the development platform

We would like to spend more time exploring the visualization and interaction design, rather than devoting too much time on coding in the limited competition period. We choose Flash as the platform due to its rich support on graphics, animation, and interaction. Although it is not popular with the development of scientific tools, we found it to fit well with quickly developing functional prototypical systems for research purposes.

Our VAST 2011 challenge submission[38] was also built upon Flash. It was the only submission that used animation to vividly demonstrate the flow of people in different locations. We spent most of our time designing the visualization and interaction. The implementation is relatively simple and straightforward.

Applications built by Flash's ActionScript 3 are compiled and run in the Flash player's virtual machine. The performance is acceptable for our systems. When searching for anomalies with SemanticPrism, most of our interactions are smooth and responsive, and its response time largely falls in the appropriate time limit suggested by Shneiderman.[39] Thanks to our client–server structure and the query-on-demand design, the client-side Flash does not need to handle extremely large amount of data. Most tasks take much less than the acceptable 2-s response time,[39] and many interactions, such as a semantic zoom in the map view, can respond instantly.

Inspirations from other submissions

In the VAST 2012 challenge workshop, we had an opportunity to see other award-winning solutions. All of them are creative and inspiring. This is actually one of the most valuable components in the challenge for us.

Dudas et al.[40] presented a solution that integrates OLAP operations[30] into VA. They used a matrix to display multiple histograms simultaneously. The analyst can perform OLAP operations (drill down, roll up, slicing, and dicing) to manipulate the matrix of curves. When compared to our time series curves, their solution appears superior in two ways. First, the matrix uses two-dimensional (2D) array, which can show five dimensions (column, row, *x*, *y*, and stack) of information and concurrently display many histograms. Second, the OLAP operations allow the analyst to generate many curves with simple interactions. In our system, the analyst has to manually select and combine filters to generate the curves. To generate curves containing exactly the same amount of information, our system needs more interactions.

Kachkaev et al.'s[41] solution used a single line to visualize the status change of an office by time. Colors of the pixel in the line present the maximum NOC, the maximum policy status, or the activity flag. These single lines are then grouped into regions. This approach inspired us to note that another level of semantic zooming can be added in our pixel-based visualization. Our first zooming level uses one pixel to display a C block at one time (Figure 4(a)). Then, the next level directly jumps to a 2D histogram (Figure 4(b) or (c)). Kachkaev's one-dimensional (1D) method can be used in between our single-pixel view and 2D curve. At our current second level of curves, we can only display curves for all offices in one region or all C-level IP blocks in one B-level block. This 1D method is compact enough to place a more detailed temporal overview of many regions/C blocks into one screen.

Choudhury et al.'s[37] submission used a machine-inferred variable "CLA," which contains inference rules that embody adductive inferences from parameters including machine class and function, policy status, activity flag, NOC, and time of the day, to compute the concern level of the computers. Our system visualizes different parameters separately. Although our integration of multiple visualizations allows the analyst to see multiple parameters at once, a comprehensive understanding of the combination of parameters is difficult.

SemanticPrism uses simple nodes in the map to show offices. To avoid overlapping, the nodes' sizes are tiny and identical and sometimes hard to read. Pabst's[42] system rearranged office locations to align with grids at the overview. Only when zooming in are the offices redrawn in full precision. Such an arrangement is superior for more effectively using the screen real estate, because the office nodes are bigger and easier to read.

Conclusions and future development

Developed as a VA system to solve 2012 VAST challenge, SemanticPrism has successfully detected all the anomalies. As Cook et al.[43] pointed out, among all the 2012 challenge submissions, traditional visualizations were well applied, although not many new visualization technologies were invented. Our visualization techniques are traditional and popular, so users can understand them well without training. The integration of visualizations is innovative and effective. The three main techniques covered the weaknesses of each other and had been used as a tool kit to detect problems. For large data with many dimensions, the data may have different characteristics at each dimension, and using a single visualization technique will be hard to fully represent the data. The query-on-demand data

handling behind semantic zoom view made it possible to create this lightweight client-side application to solve complicated VA tasks in large-scale datasets. Our system design executes Shneiderman's information-seeking mantra pretty well: the "overview to detail levels" not only exists within each type of visualization, but also links across different types. The time series curves act as an overview. The analyst filters and drills down from either the pixel-based visualization or the geospatial map. However, currently, SemanticPrism lacks the capability to automatically indicate potential anomalies and suggest appropriate zooming areas. Detection of anomalies in the "overview" curves simply relies on the analyst's visual judgment. The analyst also has to identify the suitable zooming level and search around. Early stage problems may be ignored when the signs in the curves are too subtle. Having the system to suggest zoom level and indicate anomalies in concerned areas automatically is another potential research direction for SemanticPrism.

After solving the VAST 2012 challenge 1, we started to consider whether the SemanticPrism, as a VA tool, may grow healthily to solve realistic cyber security problems. The underlying design principle may be generalized to help other large-scale high-dimensional data domains. From our standpoint, we can see that it may develop in two directions. First, SemanticPrism may allow us to find a better, natural way to integrate different visualization approaches (linking and hinting at each other), which may eventually lead to new types of visualization and interaction techniques that are more efficient for high-dimensional data analytics. Second, this tool could enhance the current client–server structure in order to allow it to solve other more complex geo-temporal VA problems in real-time large-scale datasets.

Acknowledgements

The authors would like to thank Yinghuan Peng, Abish Malik, Sohaib Ghani, and Steve Visser for their valuable help and thoughtful feedback.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Funding

This study was supported in part by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001.

References

1. VAST. *VAST challenge 2012: challenge descriptions*. Available at: http://www.vacommunity.org/tiki-index.php?page=VAST%20Challenge%202012%3A%20Challenge%20Descriptions.
2. Fua Y-H, Ward MO and Rundensteiner EA. Hierarchical parallel coordinates for exploration of large datasets. In: *Proceedings of the conference on Visualization '99: celebrating ten years*, 24–29 October 1999, pp.43–50. Los Alamitos, CA: IEEE Computer Society Press.
3. Keim DA. Designing pixel-oriented visualization techniques: theory and applications. *IEEE T Vis Comput Gr* 2000; 6(1): 59–78.
4. Oelke D, Janetzko H, Simon S, et al. Visual boosting in pixel-based visualizations. *Computer Graphics Forum* 2011;30(3):871–880.
5. Keim DA, Panse C, Sips M, et al. Pixelmaps: A new visual data mining approach for analyzing large spatial data sets. In: *Proceedings of the Third IEEE International Conference on Data Mining*. 19–22 November 2003, pp. 565–568. Melbourne, FL: IEEE Computer Society Press.
6. Guo D, Chen J, MacEachren AM, et al. A visualization system for space-time and multivariate patterns (VIS-STAMP). *IEEE T Vis Comput Gr* 2006; 12(6): 1461–1474.
7. Keim DA, Panse C and Sips M. Information visualization: Scope, techniques and opportunities for geovisualization. In: *Exploring Geovisualization* (eds J Dykes, A Maceachren and M Kraak), Oxford, UK.: Elsevier; 2005, pp.23 -52.
8. Zhang L, Stoffel A, Behrisch M, et al. Visual analytics for the big data era—a comparative review of state-of-the-art commercial systems. In: *Proceedings of IEEE symposium on visual analytics science and technology*, 14–19 October 2012, pp. 173–182. Seattle, WA: IEEE Computer Society Press.
9. Bederson BB and Hollan JD. Pad++: A zooming graphical interface for exploring alternate interface physics. In: *Proceedings of the 7th annual ACM symposium on User interface software and technology*, 2–4 November 1994, pp. 17–26. Marina Del Rey, CA: ACM Press.
10. Perlin K and Fox D. Pad: An alternative approach to the computer interface. In: *Proceedings of the 20th annual conference on computer graphics and interactive techniques*, 2-6 August 1993, pp. 57–64. Anaheim, CA: ACM Press.
11. Weaver C. Building highly-coordinated visualizations in improvise. In: *IEEE Symposium on Information Visualization (INFOVIS 2004)*. 10–12 October 2004, pp.159–166. Austin, TX: IEEE Computer Society Press.
12. Conti G, Grizzard J, Ahamad M, et al. Visual exploration of malicious network objects using semantic zoom, interactive encoding and dynamic queries. In: *Visualization for Computer Security, 2005.(VizSEC 05)*, IEEE Workshop on. 26 October 2005, pp. 83–90. Minneapolis, MN: IEEE Computer Society Press.
13. Woodruff A, Olston C, Aiken A, et al. DataSplash: a direct manipulation environment for programming semantic zoom visualizations of tabular data. *J Visual Lang Comput* 2001; 12(5): 551–571.
14. Becker RA, Eick SG and Wilks AR. Visualizing network data. *IEEE T Vis Comput Gr* 1995; 1(1): 16–28.
15. Boschetti A, Salgarelli L, Muelder C, et al. TVi: A visual querying system for network monitoring and anomaly

detection. In: *Proceedings of the 8th International Symposium on Visualization for Cyber Security*. 20 July 2011, pp. 1. Pittsburgh, PA: ACM Press.
16. Iliofotou M, Pappu P, Faloutsos M, et al. Network monitoring using traffic dispersion graphs (tdgs). In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 24–26 October 2007, pp. 315–320. San Diego, CA: ACM Press.
17. Eades P. A heuristic for graph drawing. *Congr Numer* 1984; 42: 149–160.
18. Ball R, Fink GA and North C. Home-centric visualization of network traffic for security administration. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. 25–29 October 2004, pp. 55–64. Washington, DC: ACM Press
19. Yin X, Yurcik W, Treaster M, et al. VisFlowConnect: Netflow visualizations of link relationships for security situational awareness. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, 25–29 October 2004 pp. 26–34. Washington, DC: ACM Press.
20. Muelder C, Ma KL, Bartoletti T. A visualization methodology for characterization of network scans. In: *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*, 26 October 2005, pp. 29–38. Minneapolis, MN: IEEE Computer Society Press.
21. Mansmann F, Meier L and Keim D. Graph-based monitoring of host behavior for network security. In: *Visualization for Computer Security, 2007.(VizSEC 07). IEEE Workshop on*. 29 October 2007, pp. 187–202. Sacramento, CA: Springer.
22. Borgo R, Proctor K, Chen M, et al. Evaluating the impact of task demands and block resolution on the effectiveness of pixel-based visualization. *IEEE T Vis Comput Gr* 2010; 16(6): 963–972.
23. Ziegler H, Nietzschmann T and Keim DA. Visual analytics on the financial market: Pixel-based analysis and comparison of long-term investments. In: *Information Visualisation, 2008. IV'08. 12th International Conference*. 8–11 July 2008, pp. 287–295. London, UK: IEEE Computer Society Press.
24. Ko S, Maciejewski R, Jang Y, et al. MarketAnalyzer: An interactive visual analytics system for analyzing competitive advantage using point of sale data. *Computer Graphics Forum*. 2012;31(3):1245–1254.
25. Panse C, Sips M, Keim D, et al. Visualization of geospatial point sets via global shape transformation and local pixel placement. *IEEE T Vis Comput Gr* 2006; 12(5): 749–756.
26. MacEachren AM and Kraak MJ. Research challenges in geovisualization. *Cartogr Geogr Inf Sci* 2001; 28(1): 3–12.
27. Malik A, Maciejewski R, Collins TF, et al. Visual analytics law enforcement toolkit. In: *Technologies for Homeland Security (HST)*, 2010 IEEE International Conference on, 8–10 November 2010, pp. 222–228. Waltham, MA: IEEE Xplore.
28. Sopan A, Noh ASI, Karol S, et al. Community health map: A geospatial and multivariate data visualization tool for public health datasets. *Government Information Quarterly* 2012; 29(2):223–234.
29. Podnar H, Gschwender A, Workman R, et al. Geospatial visualization of student population using Google™ Maps. *J Comput Sci Coll* 2006; 21(6): 175–181.
30. Codd EF, Codd SB, Salley CT. *Providing OLAP (On-Line Analytical Processing) to User Analysts: An IT Mandate*. White Paper, EF Codd & Associates, 1993.
31. ADOBE. Adobe - Rich Internet applications. Available at: http://www.adobe.com/resources/business/rich_internet_apps/ (accessed 30 April 2013).
32. Epanechnikov VA. Non-parametric estimation of a multivariate probability density. *Theor Probab Appl* 1969; 14(1): 153–158.
33. Simon HA. *The sciences of the artificial*. 3rd ed. Cambridge, MA: MIT Press, 1996.
34. Hoffman JE and Subramaniam B. The role of visual attention in saccadic eye movements. *Atten Percept Psycho* 1995; 57(6): 787–795.
35. Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In: *Proceedings of IEEE symposium on visual languages*, 3–6 September 1996, pp. 336–343. Boulder, CO: IEEE Xplore.
36. Javed W, Elmqvist N. Stack zooming for multi-focus interaction in time-series data visualization. In: *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, 2–5 March 2010, pp. 33–40. Taipei, Taiwan: IEEE Xplore.
37. Choudury S, Kodagoda N, Nguyen P, et al. M-Sieve: A visualisation tool for supporting network security analysts. In: *Vast 2012 MC1 Award: "Subject Matter Expert's Award*, 14–19 October 2012, pp. 165–166. Seattle, WA: IEEE Society.
38. Chen YV, Qian ZC and Zhang L. Mobile analymator: Animating data changes on mobile devices. In: *Proceedings of IEEE Conference on Visual Analytics Sciences and Technology*, 23–28 October 2011, pp. 311–312. Providence, RI: IEEE Society.
39. Shneiderman B. *Designing the user interface: strategies for effective human-computer interaction*. Boston, MA: Addison Wesley, 1986.
40. Dudas L, Fekete Z, Gobolos-Szabo J, et al. OWLAP - using OLAP approach in anomaly detection. In: *IEEE Conference on Visual Analytics Science and Technology*. 14–19 October 2012, pp. 167–168. Seattle, WA: IEEE Society.
41. Kachkaev A, Dillingham I, Beecham R, et al. Monitoring the health of computer networks with visualization. In: *IEEE conference on visual analytics science and technology*, 2012, pp. 169–170. Seattle, WA: IEEE Society.
42. Robert P. Business forensics HQ. VAST challenge MC1 award: "Good Visualization". In: *IEEE Conference on Visual Analytics Science and Technology*. 14–19 October 2012, pp. 161–162. Seattle, WA: IEEE Society.
43. Cook KA, Grinstein G, Whiting M, et al. VAST challenge 2012: Visual analytics for big data. In: *IEEE Conference on Visual Analytics Science and Technology*. 14–19 October 2012, pp. 151–155. Seattle, WA: IEEE Society.