Large Scale Network Situational Awareness Via 3D Gaming Technology

Matthew Hubbell and Jeremy Kepner

MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02420

***Abstract* - Obtaining situational awareness of network activity across an enterprise presents unique visualization challenges. IT analysts are required to quickly gather and correlate large volumes of disparate data to identify the existence of anomalous behavior. This paper will show how the MIT Lincoln Laboratory LLGrid Team has approached obtaining network situational awareness utilizing the Unity 3D video game engine. We have developed a 3D environment of the physical plant in the format of a networked multi player First Person Shooter (FPS) to demonstrate a virtual depiction of the current state of the network and the machines operating on the network. Within the game or virtual world an analyst or player can gather critical information on all network assets as well as perform physical system actions on machines in question. 3D gaming technology provides tools to create an environment that is both visually familiar to the player as well display immense amounts of system data in a meaningful and easy to absorb format. Our prototype system was able to monitor and display 5000 assets in ~10% of the time of our network time window.**

I. Introduction

Achieving network Situational Awareness (SA) is a goal of network administrators and analysts in most enterprises. A generally accepted definition of situational awareness is [1] "…the perception of the elements in the environment within a volume of space and time, the comprehension of their meaning, and the projection of their status in the near future." Analysts vigilantly pour over thousands of logs daily in an attempt to comprehend the activity in the network.

According to Amico [2], achieving SA requires the analyst to proceed through three stages: perception, comprehension, and projection. The challenges traversing through these stages are: ingesting large volumes of data, sustaining real time response, and presentation in a comprehensible format. Effective presentation of the current state allows the analyst to more rapidly identify anomalous behavior and devise an appropriate response.

Our approach to the perception challenge leverages technology utilized in the 3D gaming industry. The video game medium provides a platform for users to immerse themselves in a world in which the player is able to absorb a tremendous amount of environmental information rapidly and sustain this for a long duration of time.

Traditional network SA tools such as Snort [3], Nagios [4], or OpenNMS [5] present volumes of logs and graphs of data in a variety of forms. Over time, the stream of logs and scatter plots, bar charts, pie charts, and graphs, lose much of their meaning via information saturation. The 3D environment enables a diverse suite of tools to creatively depict asset behavior, state, and location. These tools allow us to create a virtual world that accurately represents the physical. Accurate geo-location of the assets enables the player to seamlessly identify the location of network assets operating on the network. The player is then able to perform actions to obtain pertinent information about the assets of interest enabling accurate SA.

The visualization of SA has been the subject of an increasing body of research. Drapers work identifies the benefits of using their VisAlert intrusion detection technology to elevate the comprehension of the available information [6]. They identify visualization as the key to enhance SA decision making. Other efforts to address the visualization include Glanfield's work using OverFlow to create FloVis [7], a visualization tool for investigating network traffic. The FloVis framework allows for analysts to customize their analysis-environment to enable better insight into identifying abnormal behavior. These and other visualization efforts have provided many examples of tools and frameworks to provide visual insight to SA. This works seeks to improve upon these tools by providing an environment that is more natural to the operator by leveraging the inherent advantages of 3D gaming technology.

This paper will present an overview of how we utilized 3D gaming technology to visualize the answers to the critical questions to achieve actionable network situational awareness: who is on the network, what they are doing, and where they are located. The following sections will include our approach, information flows, the capabilities afforded by Unity3D, and performance results [8].

II. Approach

Our Unity3D network SA gaming environment is setup in the format of a networked multiplayer First Person Shooter (FPS) utilizing an authoritative game server. We chose the

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Unity3D gaming engine because of its flexibility in both scripting and platform. The Unity3D development environment allows for programming in Java, C#, and Boo. Game binaries can be generated to play on Windows, MAC, iPhone, Android, and a Web Player allowing for diversity in distribution platforms.

The format of a FPS was chosen to represent the view of the player in the real world. This allows for a comfortable viewpoint enabling a sustained attention over a longer duration of time. In addition, the Unity3D environment provides the capability to easily allow for a 3rd Person perspective depending on users goals. Enabling alternative perspectives is achieved by simply changing the position of the player camera behind the operator avatar. This subtle change can allow for a broader perspective. This type of change allows for the different simultaneous players to personalize their environment.

Different players also require different levels of access to information available within the game. We have implemented tiered user accounts for managers, operators, and analysts allowing for secure connections and a variation in privileges. Managerial accounts provide a view of the overall state whereas an analyst or operator account provides access to granular details and is able to perform system actions against network assets. An analyst or operator has the ability to walk down a virtual hallway and identify which computers are actively on the network and identify the user, IP, MAC, OS, and connection time simply by performing a mouseover action.

The 3D environment significantly reduces the learning curve for new users as they are able to enter a virtual world and begin identifying systems of interest immediately. The gaming environment is one that is comfortable to younger analysts and operators due to the amount of experience they have immersed in virtual environments. As reported by Jane McGonigal [9], "…a 21-year-old has spent 10,000 hours gaming, close to about the same amount of time spent in school from 5th to 12th grade and are wired to respond to the stimuli this environment presents." A time investment of this magnitude identifies the gaming platform as an accepted and desirable format to expose to the user.

III. Data Flow

The data flow (figure 1) driving the game is a collection of output correlated with a Network Access Control (NAC) device query. To first geo-locate the assets properly within the environment, we extracted the X and Y coordinates of the nearly 10,000+ network faceplates from existing building CAD drawings. The CAD drawings of the floor plans for each building were imported into the 3D environment and used as the texture for a 2D plane that represents the respective floor. In total there are 142 individual floors rendered within the virtual world.

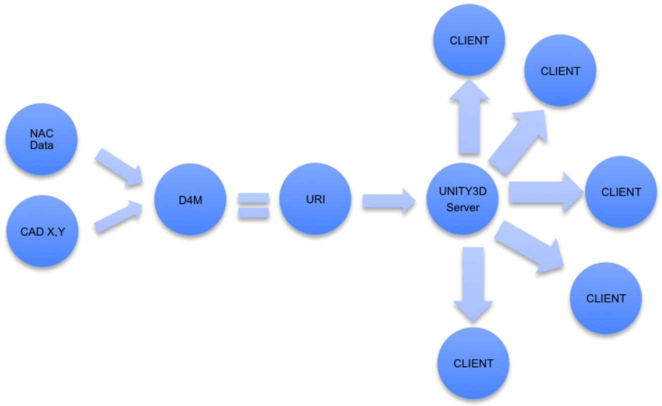


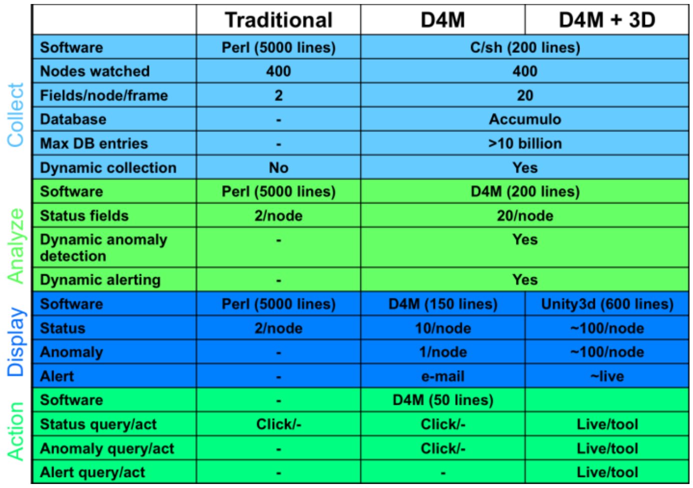
Fig. 1 Data Flow. NAC Data and CAD X,Y coordinates are fed into D4M to create a URI which is fed into the Unity3D Game Server and updates are distributed to connected clients.

The second phase was to gather information from the NAC appliance to determine who is currently connected to the network. The output ranges from as few as 500 to greater than 5000 active connections depending upon the time of day. This database query provides us with the User, IP Address, MAC Address, OS, network port, and connection time. The network port and user activity operating on that port is then associated with the proper faceplate to identify the relative X, Y coordinates to be plotted within the virtual space. This correlation is accomplished by ingesting the X, Y coordinate data as well as all NAC connection records into a Dynamically Distributed Dimensional Data Model (D4M) [10] database running in MATLAB [11]. This data is then processed to produce a dynamic URI (universal resource identifier) that is ingested into the Unity3D game server. The authoritative game server then identifies new events that have taken place and distributes an environmental update of the network state via a series of remote procedure calls (RPC's). This process ensures all connected clients are viewing the latest information. The authoritative model ensures all updates are distributed from a single source maintaining data integrity throughout the system. The amount of code necessary to perform the data analysis and 3D display is only a few hundred lines of code versus the thousands necessary in traditional web based models.

IV. Results

We have put the 3D gaming model of monitoring and system management into place with the MIT Lincoln Laboratory 900 CPU on-demand interactive TX2500 HPC system operations as well as in a Lincoln Laboratory Network SA effort [12]. In the past, we have explored many of the existing tools and have developed customized tools to aggregate logs and visually depict SA. These tools have required significant effort to integrate and have resulted in disappointing levels of actionable information. As shown in Table 1, our previous experiences with traditional monitoring system designs were involved a significant amount of code and provided limited dynamic capability. The combination of a high performance database, D4M, and a 3D gaming environment comprises a small, agile footprint and allows for a dynamic, action-based monitoring and management tool.

TABLE 1. Comparison of capabilities and efforts associated with three different SA environments we have implemented. The "Traditional" environment used perl scripts to populate web pages. The "D4M" environments used D4M to generate GUI plots and graphs. "D4M + 3D" combines the D4M analysis system and the Unity3D SA environment.

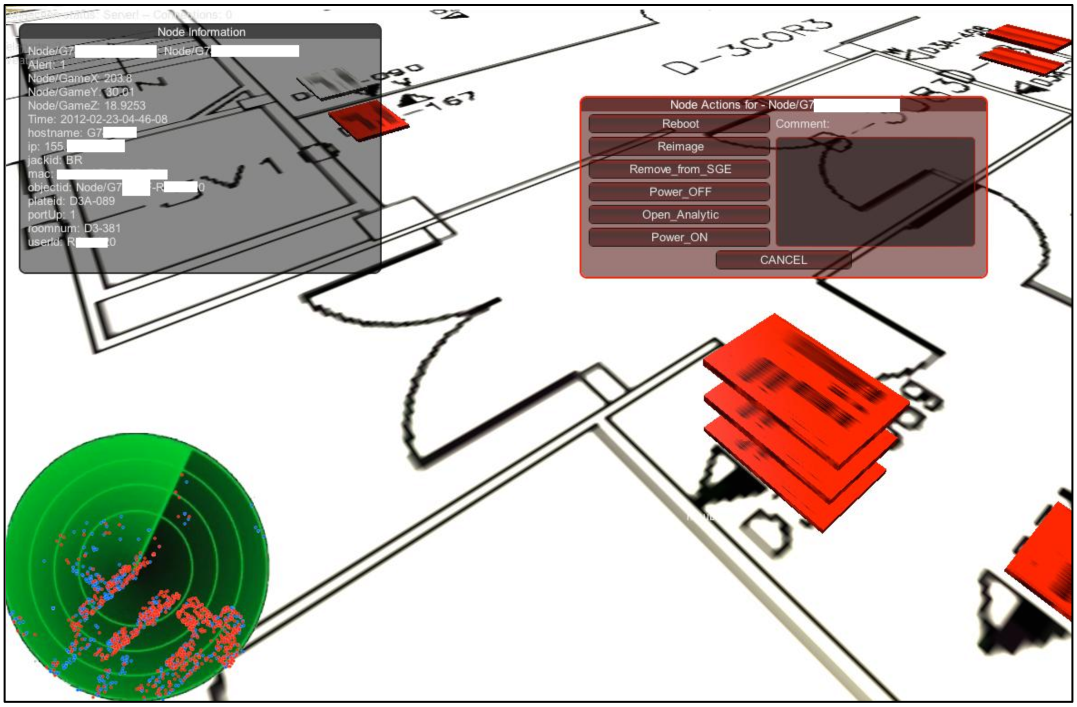


		Traditional	D4M	D4M + 3D
Collect	Software	Perl (5000 lines)	C/sh (200 lines)	
	Nodes watched	400	400	
	Fields/node/frame	2	20	
	Database	-	Accumulo	
	Max DB entries	-	>10 billion	
	Dynamic collection	No	Yes	
Analyze	Software	Perl (5000 lines)	D4M (200 lines)	
	Status fields	2/node	20/node	
	Dynamic anomaly detection	-	Yes	
	Dynamic alerting	-	Yes	
Display	Software	Perl (5000 lines)	D4M (150 lines)	Unity3d (600 lines)
	Status	2/node	10/node	~100/node
	Anomaly	-	1/node	~100/node
	Alert	-	e-mail	~live
Action	Software	-	D4M (50 lines)	
	Status query/act	Click/-	Click/-	Live/tool
	Anomaly query/act	-	Click/-	Live/tool
	Alert query/act	-	-	Live/tool

The gaming environment has capabilities that would be difficult to replicate in another environment. For example, resolving multiple nodes on the same port. The gaming environment allows us to simply look for collision events; a natural operation in gaming, rather than scripting to find like kind ports. The identification of a collision allows us to stack the assets and identify local switches or many Virtual Machine instances as seen in Figure 2.

Once an asset has been identified as being of interest, the player can then perform a variety of actions against the asset by simply clicking on the object (e.g., taking a machine offline). Clicking this desired action will log the event request by recording the users who initiated the action and the state of the machine prior to request. The game will then create a dynamic URI indicating the desired action to be taken. The URI is picked up by a monitoring process that polling for these specific messages. Once the event has been identified and authenticated the game server sends a command to the switch to turn off the switch port, thereby limiting the ability of an adverse event to spread. Another action available to the operator or analyst is the ability to push system images or patches to systems not meeting configuration requirements. An important in-game action is to further investigate a network asset by opening a targeted web page that connects to a traditional tool set to provide a forensic capability. This is consistent with McGuiness and Foy's requirements for SA tools to provide the ability to request additional data to allow for deeper analysis enabling a more informed decision [13,14].

Within the gaming environment, we have implemented a chat/command window that allows for players to communicate within the game. As shown in Figure 3, this window allows the user to inspect assets without navigating to the asset of interest. For example, a player can search for assets by specific IP address, MAC address, user name, location, or any collected identifiable attribute. The filter window allows the player to filter results on alert status or state. This gives the operator, analyst, or manager the flexibility to navigate the environment in a casual or targeted manner.

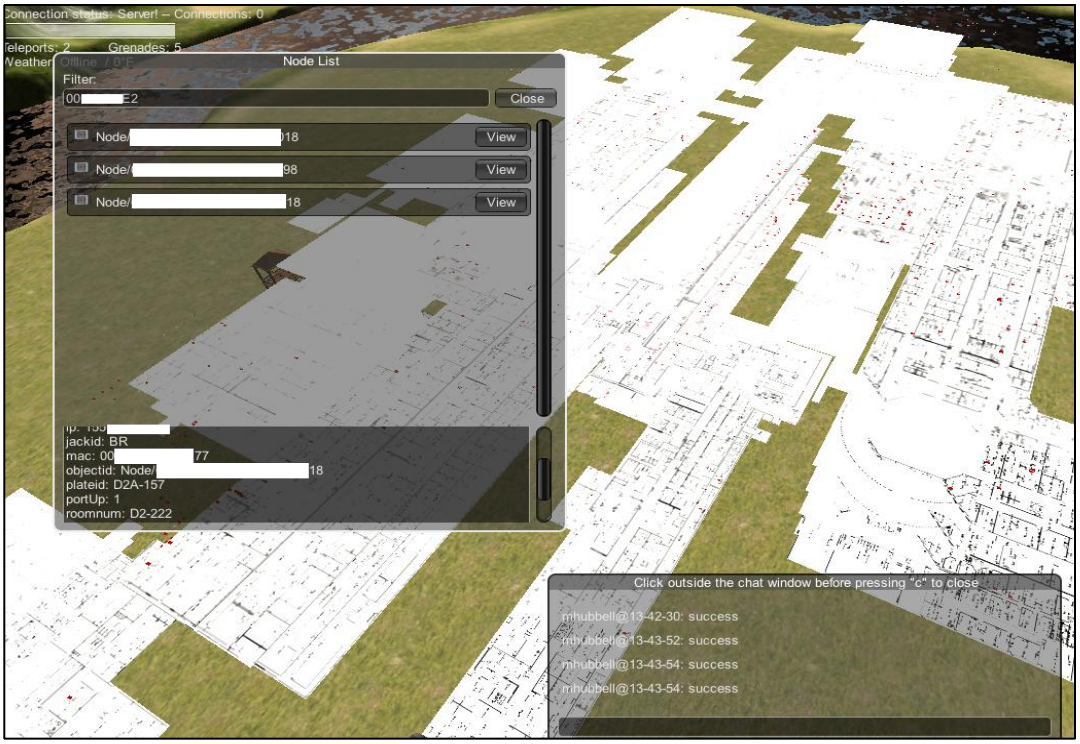

Connection status: Server! – Connections: 0

Node Information

Node/G7[redacted]: Node/G7[redacted]
Alert: 1
Node/GameX: 203.8
Node/GameY: 30.01
Node/GameZ: 18.9253
Time: 2012-02-23-04-46-08
hostname: G7[redacted]
ip: 155.[redacted]
jackid: BR
mac: [redacted]
objectid: Node/G7[redacted]-R[redacted]0
plateid: D3A-089
portUp: 1
roomnum: D3-381
userId: R[redacted]0

Node Actions for - Node/G7[redacted]

Fig. 2 Operator view. Background is floor plan showing walls, doors, offices, and port locations. Lower left shows wide field radar of all assets. Upper lefts shows information on a specific node on the network. Upper right shows actions that can be performed on this node. Lower right are several nodes on the same port that have been automatically stacked by the 3D environment.


Connection status: Server! – Connections: 0
Teleports: 2   Grenades: 5
Weather: Offline / 0°F

Node List

Filter: 00[redacted]E2

Node/[redacted]018
Node/[redacted]98
Node/[redacted]18

ip: 155[redacted]
jackid: BR
mac: 00[redacted]77
objectid: Node/[redacted]18
plateid: D2A-157
portUp: 1
roomnum: D2-222

Click outside the chat window before pressing "c" to close

mhubbell@13-42-30: success
mhubbell@13-43-52: success
mhubbell@13-43-54: success
mhubbell@13-43-54: success

Fig. 3 Manager view. Background shows floor plans for complex. Lower right shows command window that allows filters to be applied to select specific nodes on the network. Upper right shows filter results.

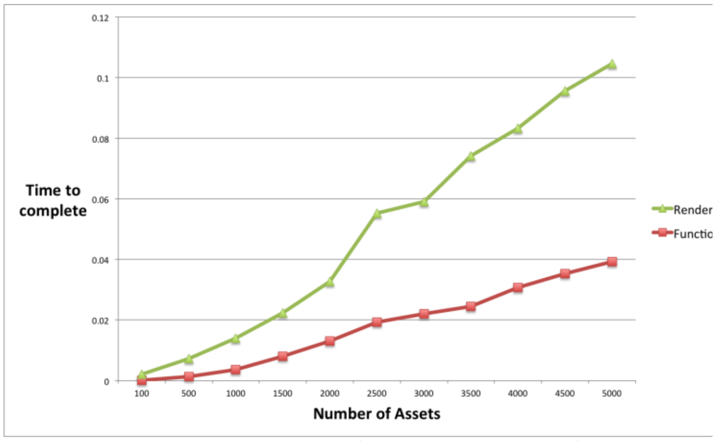


Fig. 4 Render time and asset function versus number of assets. Time is shown as fraction of the total time window of the system.

The scalability of our SA environment is shown in Figure 4. The time is broken up into two components. Function time is to execute the update function that reads in new data. Render time is the time required by the gaming environment to render the environmental updates. As the number of nodes increase these times increase. Even for 5000 nodes on a network these times are a small fraction of the monitoring time window.

V. Conclusion

We have demonstrated the benefits a 3D gaming technology provides in overcoming the challenge of visualizing network SA. One advantage is the ability to present information in a variety of ways where the player can derive the most valuable information by glancing at an asset. If an asset presents an alert triggered by an event, the asset can easily perform semantically rich notifications by a change of color, explosions, and other animations. The 3D gaming platform and monitoring model is a unique approach to a growing challenge. As more resources are dedicated to solving the network SA problem the amount of data collected will increase as well as make it more complicated to derive valuable information.

An area of further research will be to increase the performance on larger networks. There are a variety of vectors to approach this issue, which include occlusions strategies, proximity or radius detection, or streaming updates to create a more gradual process.

REFERENCES

1. Endsley, M. (1999). *Situation awareness and human error: designing to support human performance*. Proceedings of the High Systems Surety Conference, Albuquerque, NM.

2. Anita D'Amico, Kirsten Whitley, Daniel Tesone, Brianne O'Brien and Emilie Roth, *Analysts Achieving Network Defense Situational Awareness: A Cognitive Task Analysis of Information Assurance*, Proceedings of the Human Factors and Ergonomics Society Annual Meeting 2005 49: 229.

3. www.snort.org

4. www.nagios.org

5. www.opennms.org

6. G. Draper, Y. Livnat, and R. Riesenfeld, "*A Visual Query Language for Correlation Discovery and Management*," Proc. Second Ann. Visual and Iconic Language Conf. (VaIL '08), pp. 14-23, 2008.

7. Glanfield, J.; Brooks, S.; Taylor, T.; Paterson, D.; Smith, C.; Gates, C.; McHugh, J.; , "Over flow: An overview visualization for network analysis," Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on , vol., no., pp.11-19, 11-11 Oct. 2009

8. www.unity3d.com

9. McGonical, Jane, "Reality Is Broken: Why Games Make Us Better and How They Can Change the World", Pub. Date: 1/20/2011 Publisher: Penguin Group (USA) Incorporated

10. J. Kepner, W. Arcand, W. Bergeron, N. Bliss, R. Bond, C. Byun, G. Condon, K. Gregson, M. Hubbell, J. Kurz, A. McCabe, P. Michaleas, A. Prout, A. Reuther, A. Rosa & C. Yee, *Dynamic Distributed Dimensional Data Model (D4M) Database and Computation System*, ICASSP (International Conference on Accoustics, Speech, and Signal Processing), Special session on Signal and Information Processing for "Big Data"

11. www.mathworks.com

12. A. Reuther, B. Arcand, T. Currie, A. Funk, J. Kepner, H. Kim, M. Hubbell, A. McCabe, and P. Michaleas, "TX-2500 – An Interactive, On-Demand Rapid-Prototyping HPC System," *High Perfomance Embedded Computing (HPEC) Workshop*, Lexington, MA 18-20 September 2007.

13. B. McGuinness and J. L. Foy. *A subjective measure of SA: The crew awareness rating scale (cars)*. In Proceedings of the first human performance, situation awareness, and automation conference, Savannah, Georgia, USA, October 2000.

14. George P. Tadda and John S. Salerno, Air Force Research Laboratory Rome NY - S. Jajodia et al., (eds.), *Network Situational Awareness, 15 Advances in Information Security* 46, DOI 10.1007/978-1-4419-0140-8 2, Springer Science+Business Media, LLC 2010]