Animated Commentator Enhanced Network Monitoring and Visualization Application for Cyber Security Competition

Jie Yan
Department of Computer Science
Bowie State University
Bowie MD USA
jyan@bowiestate.edu

Claude Turner
Department of Computer Science
Norfolk State University
Norfolk VA USA
cturner@nsu.edu

Dwight Richards
College of Staten Island
City University of New York
Staten Island NY USA
dwight.richards@csi.cuny.edu

ABSTRACT

This paper presents an animated commentator enhanced application for visualization, animated commentary and broadcast of cyber security competitions. The visualization application seeks to engage the audience/spectators at cyber security competitions by presenting information pertinent to understanding the real-time events occurring on networks of competing team as they unfold. This paper focuses on the network monitoring and visualization Application(NMVA) component and the animated commentator component. It also discusses how the two components interoperate. The role of NMVA is to monitor critical services and attacks on blue team networks. NMVA consists of five component types: components for monitoring network traffic, components for reporting various network messages, data storage components and a visualization component. The animated commentator component aims to engage the audience/spectators using a video game like approach.

KEYWORDS

Cybersecurity Competition, Visualization Application, Animated Commentator

1 Introduction

This article discusses a visualization and broadcast application targeted to improving spectators' ability to understand and make sense of cybersecurity competitions. the original design of the application [1] incorporated a variety of techniques, including user profiling, real-time network security visualization, live video and audio monitoring, animation, computer graphics, animated commentary and human commentary. This work focuses specifically on its efforts to achieve real-time (or near real-time) network security visualization and animated commentary.

Generally, the audience at a cyber-defense competition is unable to make sense of the competition because the infrastructure to enable them to extract, visualize and comprehend events as they unfold are often not readily available to it; or even when available, might be unusable. Unlike other competitions, such as chess, where you can see which moves players make, or a basketball game where you see where the ball is and easily understand the scoring, the activities in a cyber defense competition take place on networks and computers. Hence, the strategic moves in the competition cannot be easily seen without the use of specialized analysis and visualization tools, and even with the use of such tools, only expert users can usually make sense of the output produced by the tools.

2 Background

2.1 Cyber Defense Competitions

Hoffman et al. [2] provides a useful summary of cybersecurity competitions. The authors describe the aim of a cybersecurity competition as "to provide a venue for practical education in the implementation of all strategies, tools, techniques, and best practices employed to protect the confidentiality, integrity, authenticity, and availability of designated information and information services." They posit that cybersecurity exercises "fulfill the same role as capstone projects in traditional engineering programs." That is, they enable students to synthesize and integrate knowledge acquired through course work and other learning experiences into a project usually conducted in a workplace. Jacobson and Evans [3] informs us that competitions also increase awareness and understanding of security exploits, tools, and countermeasures in the rapidly changing network security environment.

2.2 Network Visualization Application

A variety of network visualization application has been proposed in the literature. Papadopoulos et al. [4] provide a description of how spectral techniques may be utilized to discern the nature of packets flowing over a network. For example, spectral analysis can be used to differentiate a DoS (single-source) attack from a DDoS (multi- sources) attack. Baxley et al. [5] presented a tool that uses animation to visualize network attacks on LANs. The tool is primarily meant for teaching the concepts of network attacks in the context of a LAN. Ball et al. [6] described VISUAL (Visual Information Security Utility for Administration Live), a tool that allows users to quickly spot communication patterns between internal hosts and external hosts. Kazemi et al. [7] introduced the tool, IPsecLite, which demonstrates the functionality of IP Security (IPsec) standard. Wang et al. [8] proposed RBACvisual, a user-level visualization tool designed to facilitate the study and teaching of the role-based access control (RBAC) model. Yu et al. [9] proposed a visualization approach to address Domain Name

System (DNS) security challenges, such as distributed denial of service (DDoS) and cache poisoning attacks. The authors propose a methodology to identify, detect, classify, and analyze abnormal DNS querying behaviors by leveraging visualization and human visual perception capabilities.

2.3 Intelligent Tutoring Systems

Researchers face the problem of how to create an effective user interface to provide the user with a believable experience. The idea is to create a system that will use intelligent and fully animated agents to engage its users in natural face – to – face conversational interaction. Malekzadeh et al. [26] provide comprehensive work on the research on ITS's. The competition environment offers novelty as an educational environment.

3 Network Monitoring and Visualization Component

The Network Monitoring and Visualization Application (NMVA) component is a Node.js [10] application, which uses D3 [11] for dynamic generation of graphical units. It leverages several open source components for collecting and processing network data. Results are then sent to a visualization component for interpretation by a scoring algorithm and presentation methods. More specifically, NMVA consists of the following components: (1) **Monitoring Systems:** Consists of Nagios, Snort, and Linux Auditd kernel facility; (2) **Message Reporting Systems:** Consists of a Syslog-ng server and associated clients; (3) **Data Storage:** Consists of MySQL and Redis server, (4) **Animation Reporting Component:** Process and verbally report the events captured in the MySQL database (It is not discussed in-depth in this work.), (5) **Visualization component:** Renders visualization and score of captured events to the browser. It utilizes Node.js and D3.

Figure 1 illustrates NMVA's functionality and the interaction of its core visualization component with its open source external sub-systems. The sub-system, *Node.js*, operates as a Web server and binds together the activities of the Redis server, MySQL server and the Web socket server sub-systems. *Monitoring* of the competition blue team host is carried out by Linux Auditd kernel facility [12], Snort intrusion detection and prevention system (IDPS), and Nagios subsystems. Reporting is carried out by Syslog-ng server and clients. The visualization host receives log messages from the syslog-ng server, updates the visualization views and sends the messages to the MYSQL server.

Blue teams in the competition network are monitored by Auditd, Snort, and Nagios (monitors). Syslog-ng clients on the blue team machines watch log files generated by the respective monitors for events of interest. These events of interest are sent to a centralized syslog-ng server. The syslog-ng server in turn sends the data to a Redis server, also running in the administrative domain. Redis [13] is an in-memory database utilized by NMVA to listen for messages submitted to channels marked by keys. Received messages are then transmitted to clients (browsers) over a web-socket connection. Specifically, they are sent to the core NMVA visualization app, which processes each message, scores it, and displays it appropriately in a browser depending on its source or data type.

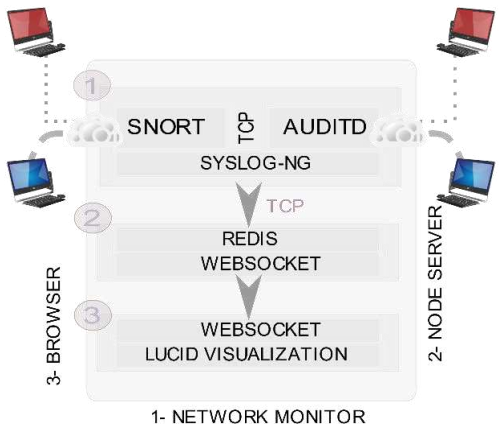

Figure 1: NMVA Components

Linux Auditd is a kernel auditing facility. Auditd may be configured to monitor user space activities and report changes to a file's content or attributes, including permission and execution modes. *Redis* is essentially an in-memory database that supports performance add-on features, including support for persistence storage (disk storage), replication in master-master and master-slave modes, publish and subscribe (pub/sub) communication channels and its own type of transaction. NMVA makes use of Redis pub/sub feature to listen for messages submitted to channels marked by keys. Received message are then transmitted to visualization clients over the web-socket connection.

Syslog-ng Open Source Edition application is a flexible and scalable centralized logging solution. It can be configured to operate in client, relay or server mode, which is determined by its prescribed role. NMVA adapts the server mode, where several syslog-ng clients send log data to a single syslog-ng server. The server runs on a computer (or virtual machine) assigned to the administrative domain. Each blue team host needs to run a syslog-ng client and be configured to send specific log file data to the single syslog-ng server. The main task performed by syslog-ng is connecting a source of log information to a specific destination or multiple destinations. The flexibility is derived from the many sophisticated processing that can be progressively performed on the data stream as it makes its way from source to destination. In particular, syslog-ng supports the application of built-in and/or custom parsers, filters, re-write rules, etc., which ultimately transforms the data to almost any desired custom format.

Blue teams are expected to establish various network services on their host machines and defend them against nefarious attacks launched by the red team. How well a blue team is able to continuously maintain the network services and defend itself against attacks is reflected in the team's score. Clearly, there is a need to monitor and record the status changes of the services running on the blue team computers. One proven solution is Nagios. Nagios is a tool that allows for centralized monitoring of the statuses of hosts and the services running on the same host machines. The main requirement is that all the blue team hosts are

reachable from the computer running the Nagios service. In the NMVA environment, Nagios is used to monitor pre-defined services running on the blue team host computers. Services can be in one of four states:

1. OK - The service is up and running
2. CRITICAL - The service is down
3. WARNING - The service is in an unstable state.
4. BUP - The service is in a "bring-up" state (not yet started)

In NMVA initial setup, the Nagios application is used to monitor the following three network services running on all of the blue team computers: SSH, NTP, and MYSQL. Adding additional services is a relatively straightforward process. A single Nagios server is used to remotely monitor the services running on all of the individual blue team computers. To accomplish this, the Nagios server needs to be configured with the necessary information to identify the blue team hosts and the services that are being monitored on them.

The initial version of NMVA also offers monitoring for the following three types of attacks: DENIAL-OF-SERVICE-ATTACK: A denial of service attack is launched against a blue team host that is detected by Snort; SHADOW-FILE-COMPROMISED: A prohibited user or process alters one or more attributes of the password shadow file. On centos 7 this file is located at /etc/shadow; SSHD-DAEMON-COMPROMISED: An unauthorized user or process escalates privilege and proceeds to shut down or to reconfigure the daemon.

Snort is a signature-based intrusion detection and prevention system (IDPS). It is effective in detecting an attack with a known signature. Once a signature is known, a rule can be written to detect the attack characterized by the signature. Hence, Snort is sometimes referred to as a rule-based IDPS. In the initial iteration of NMVA, Snort is used to detect denial of service (DoS) attacks; specifically, ping flooding DoS attack. This attack is characterized by flooding a network node with an excessive number of ICMP packets to overwhelm that node and prevent it from carrying out normal services; hence denial of service to users. NMVA ping flooding rule is based on the packet rate. If the packet rate exceeds a given rate, x, an alert is sent to the /var/log/snort/alert file. NMVA then detects the alert through its syslog-ng client, which has been preconfigured to monitor the alert file. The syslog-ng client then parses the detected alert and forward the pertinent information to the syslog-ng server. Snort runs on a single server monitoring all the blue team via its rule. Syslog-ng server then monitors the single alert file and sends the information to the Redis server.

NMVA has a graphical user interface (Fig. 2) for its SQLite competition database that can be used to configure: Scheduled tournaments (competition table), Targeted host name and IP (hosts table), Server name and IP (servers table), Available network services (service table), Players (players table), Teams (teams table). More specifically, NMVA run-time configuration parameters are stored in a SQlite3 database. The database contains the following tables; one table for each configuration page plus a few other tables for storing miscellaneous run-time data: (1) **hosts** - stores the name and IP address of blue hosts, hosts targeted for attacks, (2) **services** - stores service-name and port of services running on each blue host, (3) **servers** - stores IP address and network-name of critical network servers, including MySQL and Redis servers, (4) **teams** - stores team information, (5) **players** - stores each player's personal data, (6) **competitions** - stores information about a competition, such as the date and venue.

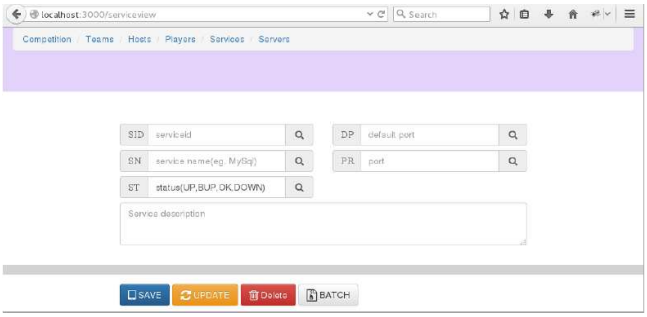

Figure 2: GUI for Exercise Configuration

NMVA has a graphical user interface (Fig. 2) for its SQLite competition database that can be used to configure: Scheduled tournaments (competition table), Targeted host name and IP (hosts table), Server name and IP (servers table), Available network services (service table), Players (players table), Teams (teams table). More specifically, NMVA run-time configuration parameters are stored in a SQlite3 database. The database contains the following tables; one table for each configuration page plus a few other tables for storing miscellaneous run-time data: (1) **hosts** - stores the name and IP address of blue hosts, hosts targeted for attacks, (2) **services** - stores service-name and port of services running on each blue host, (3) **servers** - stores IP address and network-name of critical network servers, including MySQL and Redis servers, (4) **teams** - stores team information, (5) **players** - stores each player's personal data, (6) **competitions** - stores information about a competition, such as the date and venue.

5 Scoring

NMVA scoring is derived from the statuses on the blue team hosts. In the case of DoS, a host could be in one of two states: NORMAL or DOS. This state is monitored and reported by Snort. A host is said to be in the DOS state if the traffic rate (packet arrival rate) exceeds a prescribed value, x; otherwise, it is in the NORMAL state. Critical services are monitored by Nagios and may be in one of the following four states: UP, DOWN, WARNING, or BUP (bring up). File integrity is monitored by Auditd and take one of the following two states: SECURED or COMPROMISED.

The scoring procedure is inextricably linked to the events taking place on the network. Teams start with a preset maximum number of points. As the exercise proceeds, a predefined number of points are deducted for specific events. The winner of the exercise is the team with highest score at the end of the exercise.

Scoring decisions are made based on packets meeting one of two conditions: (1) The status of a monitored event changes from a desired state to an undesired state; e.g., UP → DOWN, NORMAL → DOS and SECURED → COMPROMISED, or (2) A blue team host remains in an undesired state for a time exceeding a preconfigured penalty time limit. Packets not meeting either of the two conditions are considered non-events and simply dropped.

There are several key scoring configuration parameters. PENALTYTIMELIMIT: A negative penalty triggered if a monitored resource remains in an undesired state for a duration exceeding this limit; PENALTY: The score that is subtracted when PENALTYTIMELIMIT is exceeded; BUPTL: Remaining in the BRING-UP (BUP) state for more than BUPTL seconds triggers a BUPPENALTY penalty; BUPPENALTY: A negative score to be added to a team's score when a member host remains in BUP state for a duration exceeding BUPTL; RATETHRESHOLD: Packet arrival rates above this value indicates a DDOS attack on a host; Accumulative PENALTY is accessed proportional to multiples of PENALTYTIMELIMIT as defined above.

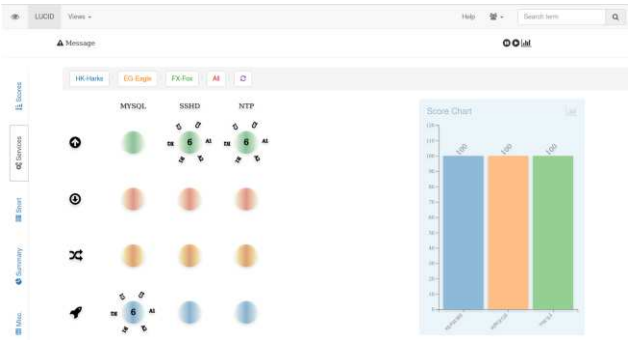

Figure 3: GUI for Cyber Security Competition Configuration

6 Visualization

Visualization is included for denial of service (DoS) and the three services just mentioned. The GUI for NMVA is depicted in Figure 3. It is interactive and has several vertical tabs that enable viewing of different kinds of data. For example, data related to critical services are viewable through the service tab, while denial of service (DoS) data received through Snort is viewable through the Snort tab. The current view in Fig. 6 depicts the state and associated scores of critical services. It shows that for six hosts (A1, A2, B1, B2, C1 and C2), MySQL is in the BUP state (blue), while SSHD and NTP are in the UP state (green) for the same hosts. The Score Chart that is available on every tab provides the updated score for the blue teams. The current view shows all three teams (HK-Harks, EG-Eagles and FX-Fox) have the same score: 100. Information for a specific team may be obtained by selecting the appropriate horizontal tab. Figure 4 provides the example of a DoS attack. (When the Snort tab is selected, this graph would replace the critical service related information that appears in the main pane of Fig. 3.) It is a simple line graph indicating the packet arrival rate. A red dotted line, indicating the threshold for denial of service, is superimposed over the packet rate graph.

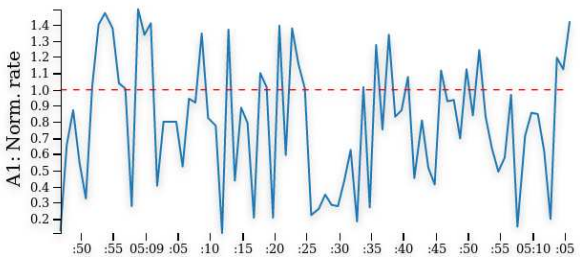

Figure 4: Denial of Service Visualization

The color scheme used in the Services tab pane indicates the state of a critical service. The color, green, indicates that the service is up, while red indicates that it is down. The example in Figure 3 shows that there are three critical services that are being monitored: MySQL, SSH and NTP. Two other states are also shown: Warning (yellow) and Bring-up (blue). The symbols consisting of letter-number pairs surrounding a circle are abbreviations for names of blue hosts in the state. C1, for example, means that the indicated critical service for host one on team C is in the given state. The number in the middle of the circle indicates the number of hosts in the state for the given critical service. The example illustrated in Fig. 3 suggests that hosts A1, A2, B1, B2, C1 and C2 are in the up state. The state information for the services in Fig. 3 are also illustrated by symbols in the left column.

7 Animated Commentator

The commentator component in the framework aims to engage the spectators in a similar fashion [14] as with video games. The commentator focuses on the visual and auditory component in terms of a virtual environment. Figure 5 illustrates the models obtained through a third-party resource for the commentator interface. We also integrated a variety of models of different gender, age and ethical background in the system to accommodate user's needs.

The core of the architecture consists of two components: (1) An integrated hardware and software interface for acquiring motion data, (2) An animation system to combine face, body and hand tags for the agent's behavioral models.

7.1 Motion Generation

For the agent to simulate human commentator behavior, initially, we used video footage of various contact sports commentators. Sample footage of commentator came from NFL roundtable discussions, super bowl commentators, and Apollo Robbins' TED talks. Each video file is roughly 9 minutes in length. These contain commentator motion data that can be applied to the animated agent. The videos used in this project were sourced from YouTube.com. To further analyze behaviors of commentators from non-contact sports, especially from cybersecurity events, we collected data from the Maryland Cyber Challenge. In 2014, we attended one such event in which we interviewed (and recorded) one of the sponsors of the competition. The interview lasted 30 minutes, and from that, we created frame-by-frame shots to aid in the analysis of facial and body gestures. Again in 2015, we performed a similar data

collection, which provided us with data one hour and thirty minutes in duration.

Figure 5: Different views and appearances of the animated commentator

We observed several motions/gestures across all the collected data. To annotate the footage, we used a tool called ELAN [15]. With ELAN, we created three tiers to represent the different commentator behaviors we analyzed. Further to that we added another tier to represent the dialog phase of the interaction between the interviewer, spectators, and event sponsors. Figure 6 shows the interface of ELAN and the different tiers which allow the user to analyze different media corpora by adding descriptive tags that are either be time-aligned to the media or it can refer to other existing annotations.

From the analyzed video clips, we observed typical animation pattern for a commentator with individual personalities. We represent this animation pattern by using tagged meta-animation sequence. Based on behaviors of interest in the expression and gestural behaviors, we developed tagged animations sequences that correspond to the tier of behavior. To capture the requisite gestures and facial expressions used by sports commentators several hardware devices and software applications.

The markerless capture of the actor's facial and body gestures is based on the point cloud capability of the Kinect v2 camera system. Brekel uses proprietary technology to map information from the point cloud to a virtual skeleton of the actor. This data is used to drive the behavior of the virtual model as seen in figure 7.

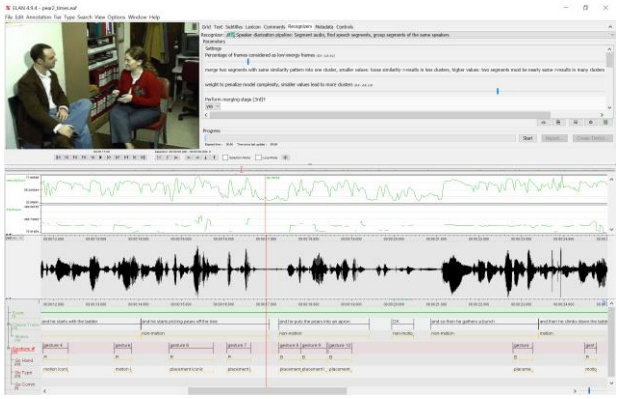

Figure 6: Annotated video clip using ELAN and the associated annotation tiers.

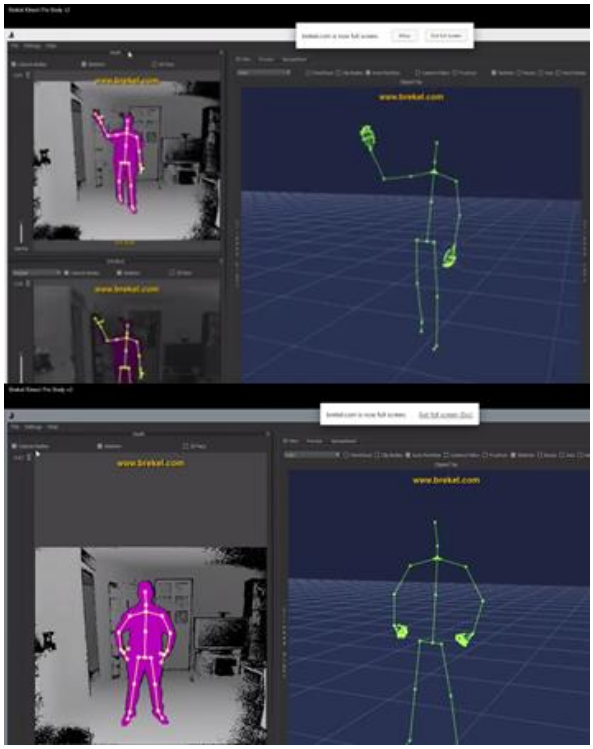

Figure 7: Actor motion in Brekel interface

7.2 Speech Generation

The animated agent is required to give commentary on the cybersecurity competition, and to give this commentary while maintaining human-like expressions, hand gestures, and

movements. The facial expression demonstrated by the agent is dependent on sentences being narrated by the agent. Having this work seamlessly for the different scenarios that may arise based on the activities in the competition involves two parts. Firstly, a database that provides the appropriate triggers for the agent to speak based on different conditions in the form of text, and secondly, a system that takes this text and converts it into speech while using the appropriate facial expressions.

For our system, the events happening during the competition, mainly the services that are up or down (up indicating this it is currently still running, and down indicating that the service has been compromised and shut down), cause data to be stored in the MySQL database. This data contains keywords that the agent needs to narrate. We created scripts that query this database whenever the virtual agent needs to announce information to the spectators.

The second part of the system is the text-to-speech (TTS) system. This component uses several third-party systems to provide speech functionality and to ensure that the text spoken by the agent also creates appropriate lip movement and facial expressions to generate a full and realistic virtual commentator. Because of the different advantages they provide, two separate text-to-speech software application are utilized. MaryTTS [16,17] is an open-source, multilingual TTS platform written in Java. The result of the query, i.e., the text, is inputted into this software to generate the necessary phonemes contained in each word. A phoneme is the basic unit of sound in any word in a specified language as seen in figure 8. The phoneme is used to determine the way the agent's lips move. Phenomena, such as accent, the shape of the mouth, language, all affect the way the mouth and lips move. This phoneme list is generated and then used to facilitate the mouth shapes of the agent when pronouncing each word. The second application is CereVoice [18,19], which is also a TTS application. We use CereVoice because it produces a much more natural sounding synthesized speech than MaryTTS. It also takes in the same text as input and is responsible for producing the final audio that will be heard by the spectators. To tie the model's speech capabilities to its motion, we use the Lip Sync pro system [20] to provide a window for synchronizing phonemes generated, emotions and gestures to dialogue in an audio file, and a component for playing back these dialogue clips on a character using totally customizable poses.

7.3 Animated Commentator

The current interface as seen in figure 9, contains the current virtual environment for one of the virtual commentators created for the system and two NPCs whose jobs are to add a level on immersion in the environment. The current scene is a 3D mock-up of the bridge of the starship Enterprise from the popular science fiction show Star Trek™: The Next Generation. On initialization of the system, the agent checks if it has established a connection to the MySQL database that houses the data collected from the network monitoring system.

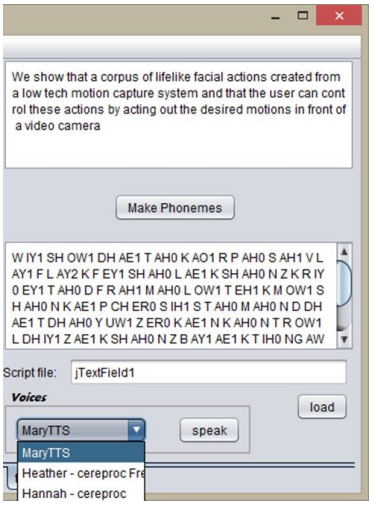

Figure 8: Translation of agent script to phoneme equivalent.

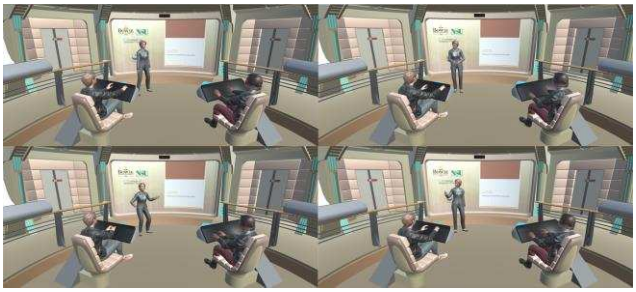

Figure 9: System interface with virtual commentator and non-playable characters (NPC) expressing different animation cycles for different behaviors.

Throughout the competition, the system probs the database for certain tags that it uses to inform its behavior. It is in this behavior that the visualization system aids in making sense of the competition events to the layman. As the spectators interface with the visualization system, the virtual commentator provides them with play-by-play view of the events of the competition – how the competitors are doing, and what they are doing. The aim is to help spectators learn to recognize their own vulnerabilities by watching competitors perform the task and then watching the correct solution path for the case. Active monitoring of the competitor's problem-solving steps would lead to a change in the learner's internal standards toward becoming more accurate in future self-assessments of performance[22].

8 Preliminary Evaluation and Analysis of Results

8.1 Network Topology

The network topology for the competition environment is depicted in Fig. 10. It shows red hosts, blue hosts, the visualization host,

MySQL-server and a syslog-ng server. For generalization sake, servers are shown on separate physical hosts and, collectively, they comprise the administrative domain. The IP-address of the MySQL-server must be configured using an SQLite3 interface prior to launching the visualization app.

8.2 Animated Commentator User Study

To investigate spectators' perception of the competition environment, we sought to (1) expose students to agents in controlled learning experiences and (2) obtain students' assessment of the agents on several affective dimensions including helpfulness, clarity, and desirability, qualitatively.

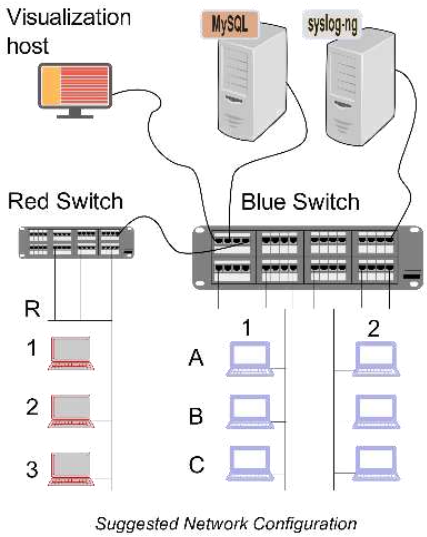

Figure 10: Network Topology

Participants in the evaluation are students enrolled at Bowie State University at the graduate and undergraduate levels. Students were recruited by their teachers who asked interested students to participate in the study and that they will receive extra credit points towards their final grades. Students were assigned to act as potential spectators and/or competitors. Undergraduates from non-computer science majors were assigned to be spectators, while computer science majors – depending on their knowledge – were assigned randomly to be spectators or participants of the blue or red teams. Graduate students were assigned the role of participant on the blue or red team. To ensure that no one team has the advantage of being overwhelming made up of graduate students, each blue and/or red team contained at least one undergraduate participant.

Participants were asked to complete a system/assessment evaluation form. They were strongly encouraged to give their opinions of the experience. The preliminary assessment consisted of four qualitative questions in which participants describe their familiarity in cybersecurity (in general), and their experience during the mock competition. Initial analysis of the pre and post-interview data shows a change in overall interest in the entire experience.

While further analysis of the impact on knowledge is required, a previous study conducted by Agada and Yan [21] in which participants were presented with three versions of the same virtual agent with different affective states. In presenting randomly selected participants with one of the three clones, we can make certain correlations between the learners' comprehension and agent's affect level. In those experiments, we see that the affect capable virtual agent outperformed all other clones in comprehension tests and agent perception. In the above mentioned study, the system was designed as a teacher, the challenges here is whether the same effects are observed when the agent role has changed from strict instructional mode to one of commentary. As Baylor [23]noted, the impact a virtual human has on the learner's motivation learning gains yields the same results [24,25].

9 Conclusion

This work presented a comprehensive software application for visualization, animated commentary and broadcast of cyber security competitions. The visualization application seeks to engage the audience at competitions by presenting information pertinent to understanding the real-time events occurring on networks of competing team as they unfold. The system's original design encompasses a variety of components. This article focuses on its network monitoring and visualization component—the Network Monitoring and Visualization Application (NMVA)—and its animated commentator component. It also briefly discusses how the two components interoperate. The role of NMVA is to monitor critical services and attacks on blue team networks. NMVA consists of five component types: components for monitoring network traffic, components for reporting various network messages, data storage components and a visualization component. The animated commentator subsystem in the framework aims to engage the spectators using a video game like approach. The commentator focuses on the visual and auditory component in terms of a virtual environment. On its own, the virtual commentator system is a sufficient education tool but in concert with other aspects of the entire competition environment, it has a potential to be a very powerful tool. The commentator would receive data about the network and competitors from network, visualizations, and scoring subsystems and would relay that information to the novice spectator.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DUE-1303424.

REFERENCES

[1] C. Turner, J. Yan, D. Richards, P. O'Brien, J. Odubiyi, and Q. Brown, "Lucid: A visualization and broadcast system for cyber defense competitions," *ACM Inroads*, vol. 6, no. 2, 2015.

[2] L. Hoffman and D. Ragsdale, "Exploring a National Cyber Security Exercise for Colleges and Universities," Washington, D.C.,, 2004.

[3] D. Jacobson and N. Evans, "Cyber Defense Competition," in 2006 ASEE Annual

Conference & Exposition: Excellence in Education.

[4] C. Papadopoulos, C. Kyriakakis, A. Sawchuk, and X. He, “CyberSeer: 3D audio-visual immersion for network security and management,” in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security - VizSEC/DMSEC '04*, 2004, p. 90.

[5] T. Baxley, J. Xu, H. Yu, J. Zhang, X. Yuan, and J. Brickhouse, “LAN attacker,” in *Proceedings of the 3rd annual conference on Information security curriculum development - InfoSecCD '06*, 2006, p. 118.

[6] R. Ball, G. A. Fink, and C. North, “Home-centric visualization of network traffic for security administration,” in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security - VizSEC/DMSEC '04*, 2004, p. 55.

[7] N. Kazemi and S. Azadegan, “IPsecLite,” in *Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE '10*, 2010, p. 138.

[8] M. Wang, J. Mayo, C.-K. Shene, T. Lake, S. Carr, and C. Wang, “RBACvisual,” in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '15*, 2015, pp. 141–146.

[9] H. Yu, X. Dai, T. Baxliey, X. Yuan, and T. Bassett, “A visualization analysis tool for DNS amplification attack,” in *2010 3rd International Conference on Biomedical Engineering and Informatics*, 2010, vol. 7, pp. 2834–2838.

[10] “Node.js.” [Online]. Available: https://nodejs.org/en/. [Accessed: 18-Jun-2018].

[11] “D3: Data-Driven Documents,” *IT Security Community Blog*, 2018. [Online]. Available: https://d3js.org/. [Accessed: 04-Jan-2018].

[12] Scotpack, “A Brief Introduction to Auditd.” [Online]. Available: https://security.blogoverflow.com/2013/01/a-brief-introduction-to-auditd/. [Accessed: 04-Jan-2018].

[13] Redis, [Online]. Available: https://redis.io/. [Accessed: 18-Jun-2018] DOI:https://doi.org/10.1007/978-3-540-74997-4_65

[14] Gifford Cheung and Jeff Huang. 2011. Starcraft from the stands: understanding the game spectator. *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.* (2011), 763–772. DOI:https://doi.org/10.1145/1978942.1979053

[15] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. 2006. ELAN: A professional framework for multimodality research. In *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*. DOI:https://doi.org/10.3758/BRM.41.3.591

[16] M. Schröder and M. Schröder. 2007. Interpolating expressions in unit selection. In *Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, 718–720. DOI:https://doi.org/10.1007/978-3-540-74889-2_66

[17] M. Schröder and J. Trouvain. 2003. The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. *Int. J. Speech Technol.* 6, (2003), 365–377. DOI:https://doi.org/10.1023/A:1025708916924

[18] Matthew P. Aylett and ChristopherJ. Pidcock. 2007. The CereVoice Characterful Speech Synthesiser SDK. *Aisb 2007* (2007), 174--178.

[19] CereVoice Engine Text-to-Speech SDK. *CereProc*. Retrieved January 1, 2017

[20] 2016. LipSync Documentation. *Rogo Digital*. Retrieved May 5, 2017 from https://lipsync.rogodigital.com/documentation/

[21] Ruth Agada and Jie Yan. 2012. Research to improve communication by animated pedagogical agents. J. Next Gener. Inf. Technol. 3, 1 (2012), 58. Retrieved from http://connection.ebscohost.com/c/articles/76342306/research-improve-communication-by-animated-pedagogical-agents

[22] Reza Feyzi-Behnagh, Roger Azevedo, Elizabeth Legowski, Kayse Reitmeyer, Eugene Tseytlin, and Rebecca S. Crowley. 2014. Metacognitive scaffolds improve self-judgments of accuracy in a medical intelligent tutoring system. Instr. Sci. 42, 2 (March 2014), 159–181. DOI:https://doi.org/10.1007/s11251-013-9275-4

[23] Amy L. Baylor. 2011. The design of motivational agents and avatars. Educ. Technol. Res. Dev. 59, 2 (February 2011), 291–300. DOI:https://doi.org/10.1007/s11423-011-9196-3

[24] Samuel Alexander, Abdolhossein Sarrafzadeh, and Stephen Hill. 2006. Easy with Eve : A functional affective tutoring system. In Workshop on motivational and affective issues in ITS-8th international conference on intelligent tutoring systems, 38–45.

[25] Samuel Thomas Vaughan Alexander. 2007. An affect-sensitive intelligent tutoring system with an animated pedagogical agent that adapts to student emotion like a human tutor. Massey University.

[26] Mehdi Malekzadeh, Mumtaz Begum Mustafa, and Adel Lahsasna. 2015. A review of emotion regulation in intelligent tutoring systems. Educ. Technol. Soc. 18, 4 (2015), 435–445.