Visualizing the New Zealand Cyber Security Challenge for Attack Behaviors

Jeffery Garae*, Ryan K. L. Ko‡, Janice Kho†, Saidah Suwadi†, Mark A. Will‡ and Mark Apperley‡
*Cyber Security Lab - Department of Compter Science
Unversity of Waikato, Hamilton, New Zealand 3240
Email: jg147@students.waikato.ac.nz
Email: {mark.will, mapperle, ryan.ko}@waikato.ac.nz
†Nanyang Polytechnic, Singapore 569830
Email: {140603H, 144668C}@mymail.nyp.edu.sg

***Abstract*—Datasets are important for security analytics and mitigation processes in cyber security research and investigations. "Cyber security challenge (CSC)" events provide the means to collect datasets. The New Zealand National cyber security challenge event is designed to promote cyber security education, awareness and equally as important, collect datasets for research purposes. In this paper, we present the: (1) Importance of cyber security challenge events, (2) Highlight the importance of collecting datasets, and (3) present a user-centric security visualization model of attack behaviors. User-centric features with the theoretical concept of Data Provenance as a Security Visualization Service (DPaaSVS) are used to display attacks commencing at the reconnaissance stage through to compromising a defending team machine and exploiting the systems. DPaaSVS creates the ability for users to interact and observe correlations between cyber-attacks. Finally we provide future work on Security Visualization with Augmented Reality capabilities to enhance and improve user interactions with the security visualization platform.**

Index Terms - Security Visualization; Cyber-attacks; User-centricity; Data Provenance; Datasets.

I. Introduction

Visualisation is a key method for researchers to effectively analyse cyber attack patterns. However, there is often a lack of datasets available, as victim organisations tend not to share data showing how they were hacked for fear of reputation damage. To overcome this lack of shared, or public dataset(s), we established the New Zealand Cyber Security Challenge (NZCSC) as a way to collect and analyse realistic cyber attack patterns on top of providing contestants a hand-ons, educational environment through capture-the-flag, and red-blue team rounds [18], [26]. Generally, CSC platforms are in two forms: (1) Capture the Flag (CTF) - a reverse engineering challenge and (2) Attack and Defend challenge [5], [19], [11]. In addition, collecting cyber security datasets for academic research purposes is another core reason of implementing cyber security challenges. Allowing participants from high schools, universities and industry experts gives a wider range of datasets during the competition.

A. Paper Structure and Outline

In this paper we present an (1) Overview of the "national CSC" competition platform and the purpose as mentioned in Section III, and (2) "User-centric security visualization platform" implemented to help academic cyber security research.

Section II shares past and existing reseach work around visualizing cyber security challenge events. Section IV provides our first contribution, the "New Zealand CyberSecurity Challenge (NZCSC)" Backend platform and the importance of designing a backend visualization platform that can efficiently communicate to the frontend visualization platform. Section V provides our main contribution, that is the User-centric security visualization Frontend platform. It serves with core purpose of interacting with users. Section VI evaluates the CyberSecurity Challenge platform, namely identifying challenges and how to improve the competition. Section VII evaluates the security visualization platform and added user-centric features and finally, Section VIII concludes this paper and states future work.

II. Security Visualization Background

With benefits of experiencing adversarial cyber incidents and their natures, both aim to contribute to developing skilled cyber security professionals [18], [19]. Security challenge competitions are a powerful educational resource platform which drives by motivating students to excel in security research with future innovation in security techniques and tools [12], [8], [31], [14]. CSC competitions provides near real-time experiences and opportunities to educate students, provide situation awareness and execute holistic cyber-attack scenarios in a controlled environment [12], [29]. Understanding how hacking is carried out elevates the participants (students & industry security professionals) knowledge on how to handle cyber-attacks during an incident response scenario [6], [11], [28].

Humans learn faster with the use of visual representation of concepts, ideas, thoughts and knowledge [25]. DARPA's visual software analysis platform that aims to observe attacks executed during a capture the flag (DEFCON CTF) challenge by plotting attack execution and comparing them to normal traffic [1]. Visual interactions and sensory representations based on security abstract data to reinforce cognition [7]. The use of AI bots to identify, diagnose and fix software flaws at real-time during the challenge [3].

2324-9013/17 $31.00 © 2017 IEEE
DOI 10.1109/Trustcom/BigDataSE/ICESS.2017.362

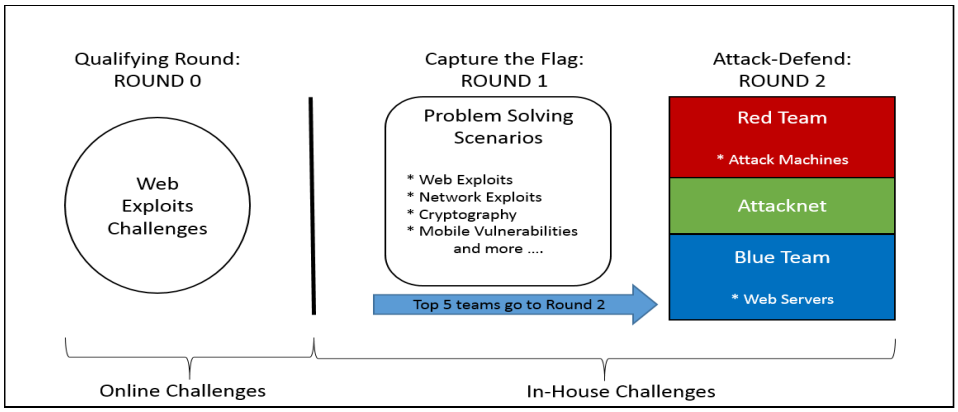

Fig. 1. The CyberSecurity Challenge Platform Design.

III. Cybersecurity Challenge Platform

The New Zealand "National CyberSecurity Challenge (NZCSC)" (https://cybersecuritychallenge.org.nz/) competition was established in 2014 by the University of Waikato along with its industry partners. For the past three years (2016 challenge - 267 qualifying participants), the challenge has been created into three rounds: (0) Online qualifying challenge, (1) Capture the Flag (CTF) challenge, and (2) Attack and Defend challenge. The competition aims to provide cyber security education across academia and the industry environment by up-skilling interested students and providing security professionals with the latest possible attack and defend scenarios.

Overall, the academic purpose of establishing and executing cyber security challenges are in relation to the following reasons: (1) Cyber security education and situation awareness, (2) Eliminates and minimizes data collection & sharing ethical issues, (3) Creates an avenue for Dataset collection, and (4) Ability to run low cost cyber security events in a controlled environment.

The open online qualifying challenge and CTF challenge are tailored around web exploits, encryption, network routing and mobile vulnerabilities. All challenges are scored to a scoring system which allocates different points for various challenges depending on their complexity to solve. The "Attack and Defend" challenge infrastructure is based on a local network environment with virtual machines for the teams. Figure 1 shows the infrastructure design. The top 5 teams from the Capture the Flag (CTF) challenge, qualify to compete in the Attack and Defend challenge.

IV. NZCSC Security Visualization Backend Platform

While we have briefly introduced the cyber security challenge competition infrastructure and environment, our main focus and contribution for this paper is on two research areas:

1) *Dataset*: The data collected from the past three years of the New Zealand cyber security challenge events.
2) *Security Visualization*: Understanding security attack events using a 'user-centric' Security Visualization framework with Provenance features.

A. Data Collection and Logging Types

Data logging and collection are important for monitoring systems and networks. It allows network and security experts to monitor and maintain systems in a most known secure environment with the help of regularly implementing security protocols, rules and policies based on identified cyber-attacks and threats. As briefly mentioned in Section I, Datasets are crucial for cyber security and data science researchers. This means understanding cyber-attacks heavily relies on collected datasets from the captured attacks. The Cyber Security challenge logging mechanisms are in the following categories: (1) Network (pcap) logs using Wireshark, (2) Linux kernel audit logs, (3) System logs using sysdig, (4) Apache top logs and (5) VLC screen-captured videos capturing user actions and inputs during different security challenge scenarios. The selection of these logging mechanisms aims to monitor and log all attack actions executed by the participating teams from all levels, starting from network traffics to kernel level actions, user logs, system level actions, application access & error logs and user inputs. These logging mechanisms are configured for selected teams in Capture the Flag challenge and all Attack and Defend challenge teams. Logs are configured to write and are saved into a separate external virtual machine - backup storage.

B. NZCSC Competition Raw Dataset

Datasets are very important in fostering research, in particular understanding how attacks occur. Therefore, obtaining datasets is vital given the ability to use them for security analysis. The NZCSC competition datasets are in the following types and formats:

1) Wireshark - pcap logs.
2) Linux kernal audit logs.
3) System logs - sysdig.
4) Apache top logs.
5) VLC screen-captured videos.

At this stage, the datasets collected are privately stored and used only within the university's research purpose. However, the ultimate goal is to provide a public dataset which can be used by other interested researchers in the near future.

C. Anonymization and Standardization

In order for such sensitive datasets to be used for cyber security research purposes, with the ultimate goal of publishing the available datasets publicly, "Anonymizing and Standardizing" the dataset is crucial. *Why data anonymization process?* Due to security, privacy and sensitive reasons, this eliminates the chances of attributing back to distinctive network sources. The anonymization method focus on the following:

- Locate names and IP addresses attributing to any known sources
- Substitute the names and IP addresses to new generic names and IP addresses based on created standard.

Why data standardization process? Standardization procedures are taken to allow datasets of various formats be used across numerous analytic tools. This allows interested public researchers to easily integrate the dataset with their data analytics or threat Intelligence tools.

The process of analyzing the collected data is done in three methods: (1) manually analyzing logs and identifying their existing format (knowing how many attributes and types of delimiters used), (2) identifying and categorizing different attacks by analyzing all different types of logs, and (3) creating scripts to dynamically and automatically anonymize the dataset based on analysis and insert them into database tables. Table I shows anonymized data being categorized into different types of attacks and stored into MySQL. Such scripts includes regular expressions that are being used to search and match rows, or excluded rows in various logs. Examples of excluded rows are 'commented information' and duplicated information which do not contribute to how security attacks are executed. This script acts as a "*Collector*" mechanism that checks for new data inputs, anonymize the inputs and inserts them into respective database tables.

D. Backend Server Implementation

With the ultimate goal of providing a user-centric security visualization infrastructure for our existing cyber security challenge competition, anonymizing and standardizing the datasets are made easy with a choice of known database. Based on the cyber security challenge event time-frame (duration) against the estimated data collected within that time-frame, 'MySQL' managed through phpMyAdmin. This is due to practical reasons such as user friendly web interface with less implementation complexities and existing web server (XAMPP) integration capabilities [13].

E. Backend Design Overview

Figure 2 details the NZCSC Security Visualization backend infrastructure overview. The components include the CSC platform (Figure 1), a collector and the MySQL database. The 'Collector' is a php script-base component which checks the NZCSC data storage platform for new data inputs, collects them and writes them into appropriate tables in MySQL. Once attack datasets are analyzed, anonymized and stored in the database, selected data can be exported into comma-separated values (.csv) or JavaScript Object Notation (.json) formats for frontend use, such as visualization.

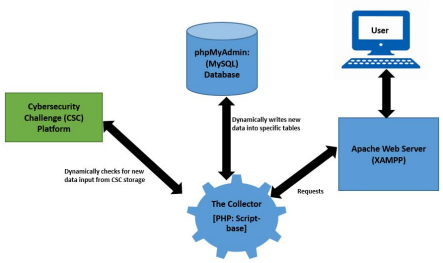

Fig. 2. Backend Implementation Overview.

While most Security Visualization platforms concentrate on the frontend, our backend development objectives are:

- Develop an easy-to-use backend platform with interface capabilities for any users to use and not just developers and IT experts
- A less expensive backend-frontend integration platform with reasonable efficient storage and processing power.
- A easy to manage security visualization backend infrastructure for educational use.

The core component of the NZCSC Security Visualization platforms are: (1) Apache XAMPP (Web Server) with phpMyAdmin, (2) a "Collector (PHP Scripting - base)" and (3) NZCSC competition data source. All backend processes are scripted, automated and connected to the security visualization frontend platform.

F. Attack Analysis and Anonymization

In order for our NZCSC security visualization framework to be effective and efficient with useful visual insights, a crucial contributing process to our visualization infrastructure is '*Attack Analysis*' process. This process is executed in two steps: (1) Identification of attacks and (2) Attack verification against recorded screen captured video.

1) Identification of Attacks: Identifying different types of attacks based on the collected dataset requires both manual user checks and scripting mechanisms to obtain the right information linked to the attacks. This means, the steps used to identify the types of attacks executed during the CSC competition require extra effort and precise inputs. These steps include: (1) manually identifying the attack signatures, e.g. SQL injection; (2) Creating scripts to scan and read through all logs, collect, categorize and format attack footprints into attack types; (3) Create tables in database; and (4) Insert and store attack records into related tables in the database.

2) Attack Verification against Screen Captured Videos: As part of the logging requirements, we needed to evaluate and verify that the attacks logged are synchronized with actual user-inputs captured from participating teams. This eliminates any error on the information collected using the logging mechanisms. The most attractive contents of the dataset are the red (Attacking) and blue (Defending) team logs showing the

TABLE I
DYNAMICALLY STORING ATTACKS INTO THE DATABASE.

ID	Time	Source	Destination	Protocol	Command	Attack Type
26	18:29:28	10.0.53.4	10.42.122.123	TCP	nmap 10.42.122.0/24	Reconnaissance
27	18:29:28	10.0.53.4	10.42.122.151	TCP	nmap 10.42.122.0/24	Reconnaissance
28	18:29:28	10.0.53.4	10.42.122.200	TCP	nmap 10.42.122.0/24	Reconnaissance
29	18:29:28	10.0.53.4	10.42.122.60	TCP	nmap 10.42.122.0/24	Reconnaissance
30	18:29:43	10.0.53.4	10.42.122.11	TCP	nmap -sT –top-ports=100 10.42.122.0/24	Reconnaissance
31	18:29:43	10.0.53.4	10.42.122.123	TCP	nmap -sT –top-ports=100 10.42.122.0/24	Reconnaissance
32	18:29:43	10.0.53.4	10.42.122.151	TCP	nmap -sT –top-ports=100 10.42.122.0/24	Reconnaissance
33	18:29:43	10.0.53.4	10.42.122.200	TCP	nmap -sT –top-ports=100 10.42.122.0/24	Reconnaissance
34	18:29:43	10.0.53.4	10.42.122.60	TCP	nmap -sT –top-ports=100 10.42.122.0/24	Reconnaissance
35	18:29:57	10.0.53.2	10.42.122.200	TCP	/usr/bin/python /usr/bin/sqlmap -u http://10.42.12...	SQL Injection
36	18:30:24	10.0.53.3	10.42.122.200	HTTP	GET /adminlogin action?username=&password=...	URL Manipulation
37	18:31:18	10.0.53.1	10.42.122.200	HTTP	GET /adminlogin action?username=%27&password=...	URL Manipulation
38	18:31:29	10.0.53.1	10.42.122.200	HTTP	GET /adminlogin action?username=Admin&password=...	URL Manipulation
39	18:31:59	10.0.53.1	10.42.122.200	HTTP	GET /adminlogin action?username=Admin&password=...	URL Manipulation
40	18:32:49	10.0.53.2	10.42.122.200	TCP	/usr/bin/python /usr/bin/sqlmap -u http://10.42.12...	SQL Injection
41	18:33:26	10.0.53.2	10.42.122.200	TCP	/usr/bin/python /usr/bin/sqlmap -u http://10.42.12...	SQL Injection
42	18:35:01	10.0.53.3	10.42.122.200	HTTP	GET /adminlogin action?username=&password=...	URL Manipulation
43	18:35:48	10.0.53.1	10.42.122.200	HTTP	GET /post/create action?name=Admin&date=12%2F%...	Remote Code Execution
44	18:35:51	10.0.53.3	10.42.122.200	HTTP	GET /adminlogin action?username=&password=...	URL Manipulation
45	18:38:37	10.0.53.1	10.42.122.200	HTTP	GET /adminlogin action?username=%3C%3ECoolGuy...	URL Manipulation
46	18:39:59	10.0.53.3	10.42.122.200	HTTP	GET /adminlogin action?username=	URL Manipulation
47	18:41:02	10.0.53.1	10.42.122.200	HTTP	GET /post/create action?name=NewAdmin&date=12%2F%...	Remote Code Execution
48	18:41:20	10.0.53.1	10.42.122.200	HTTP	GET /post/create action?name=%3C%3ECoolGuy...	Remote Code Execution
49	18:42:03	10.0.53.1	10.42.122.200	HTTP	GET /post/create action?name=%3C%3ECoolGuy...	Remote Code Execution
50	18:42:12	10.0.53.1	10.42.122.200	HTTP	GET /adminlogin action?username=Mark&password=...	URL Manipulation

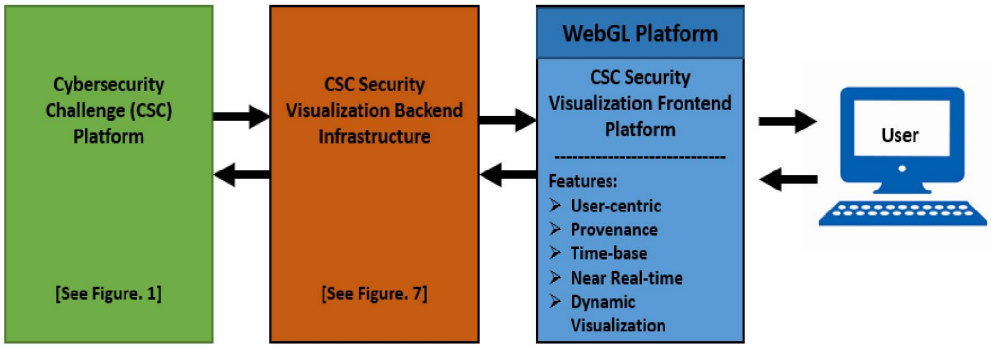

Fig. 3. NZCSC Security Visualization Implementation Overview.

most attack correlation events between the teams. Therefore closely observing the screen captured videos of red and blue team was one of our main tasks for the backend infrastructure. The verification tasks emphasized on the log *time-stamps* with screen captured video time-stamps. This sync process helps verify the actual method, source and destination of attacks. Once all processes are identified, automated and dynamic scripts are implemented as part of the verification process to filter and store important details such as source and destination IPs. '*Tshark*' commands and '*regular expressions*' are used in scripts to store results in multi-dimensional arrays of multiple attack protocols. These scripts allow efficient data transition from the backend to the frontend - the NZCSC visualization frontend which will be discussing more in Section V.

V. NZCSC SECURITY VISUALIZATION FRONTEND PLATFORM

The NZCSC Security Visualization frontend performance heavily relies on how efficient data is being processed from the backend then pushed to the frontend for visualization. And there are important specifications and features that needs to be addressed during the design phase of our visualization mockup. These includes:

- Frontend and backend compatibility.
- Data processing power and performance between backend and frontend.
- User-centric features for frontend visualization platform.

The entire NZCSC Security Visualization platform design (Figure 3) shows our WebGL [24] user-centric security visualization platform which displays the various attacks during the cyber security challenge competition. With the amount of data

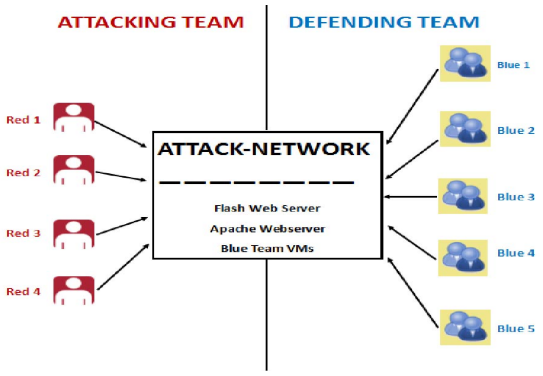

Fig. 4. Red - Blue Challenge Design Overview.

analyzed, our core focus was on Red-Blue Team competition. Therefore, data requested from the backend and visualized are the 'attack and defend' competition as shown in Figure 5. In brief, our security visualization frontend showcase cyber-attack activities between four red attacking teams against five blue defending teams as illustrated in Figure 4.

Key components for the security visualization frontend are: (1) WebGL visualization platform and (2) PHP scripting platform. Similar to the backend 'Collector', the PHP scripting platform checks the database tables for new inputs and pushes relevant to the frontend for visualization. For example, requesting to visualize an attack at the certain time (interested attack search). Different components of the platform are discussed in the remaining sections of this paper.

A. Implementation

1) Why the choice of WebGL?: The advantages of using WebGL for security visualization is due to its following features: (1) A suitable cross-platform for visualization, (2) it is fast and has capacity to fully utilize hardware acceleration, making it suitable for complex interactive visualizations, (3) It has efficient 3D visualization capabilities to visualize data, and (4) provides users with user-centric control over visualizations [24].

2) Frontend Development Methodologies: The frontend security visualization implementation uses dependencies such as libraries to create and display animated 3D visual graphics in web browsers. These includes three.js (a cross-browser JavaScript library/API, particularly trackballcontrols.js), jquery, and Bootstrap [10], [9], [27]. The frontend development steps are outlined below:

- *Setting up of the environment*: Components includes XAMPP, Three.js, jquery, Bootstrap CDN and Ajax.
- *Creating a WebGL visualization infrastructure (WebGL VI).*
- *Teams Representation.*
- *Stimulating an Attack.*
- *Data Provenance Timeline.*
- *Adding Information to the WebGL VI.*

B. Attack Analysis and Statistics

The security visualization platform was able to reveal interesting visual outputs as seen in Figure 5. It has the additional visual feature whereby attacks are tallied as they are fetched from the database for visualization. The statistics visual view in Figure 6 has indicated that majority of the time, 'Reconnaissance' was done during the cyber security challenge competition. 'Semantic URL attack' and 'Remote Code Execution' were highly used to exploit the blue teams systems and network. Other regular attacks used include 'URL Manipulation' and 'Directory Traversal attack'. These were the primary vulnerabilities added to the challenge. In addition, attack statistics are retrieved from collected datasets, with the use of functions and visually displaying them in the main security visualization window as well as in the statistical view. Different colors represent different attacks and the increase of colored points on the curves in Figure 5 indicates an increase in attacks visualized. Frequencies of attacks vs time are visualized for the Round-2 duration of the cyber security challenge competition.

C. Data Provenance as a Security Visualization Service (DPaaSVS)

As mentioned in Subsection V-A2, data provenance is an important added feature for this security visualization platform [16], [20], [30]. We introduce the term "Data Provenance as a Security Visualization Service (DPaaSVS)" namely to provide tracking, monitoring and attribution of attacks using security visualization. IP addresses, time-stamps and user-centric visual features associated with known attacks identifying where various attacks originate (IP address sources) from and to which destination IP addresses are being the targeted victims of the attacks. Login / logout details, Password changes, and even failed resource access are used when trying to reconstruct security events. A provenance of the attack executions can be visualized as part of security visualization displaying the process of attacks beginning with the process of reconnaissance, then executing a default password (DPAtk) attack to compromise the defending teams machine and later executing other attacks such as: (1) remote code execution (RCE) attacks, or (2) URL manipulation (URL-M-Atk) attack. These related commands which allows attackers to bypass a system also provides pieces of intelligence required to visually map out how an attack is executed from start to finish, and from source to destination. Figure 6 shows the frequency vs time graph illustrating an overview of the attacks execution and their corresponding times. Understanding the attack processes shown in Figure 9 provides users with the knowledge to map out how attacks are linked and are escalated from reconnaissance to compromising default passwords and further executing harmful attacks.

Therefore, equipping and enabling users with the opportunity to interact effectively with the visualization platform using such provenance features to search for any IP address of interest, creates the concept of DPaaSVS.

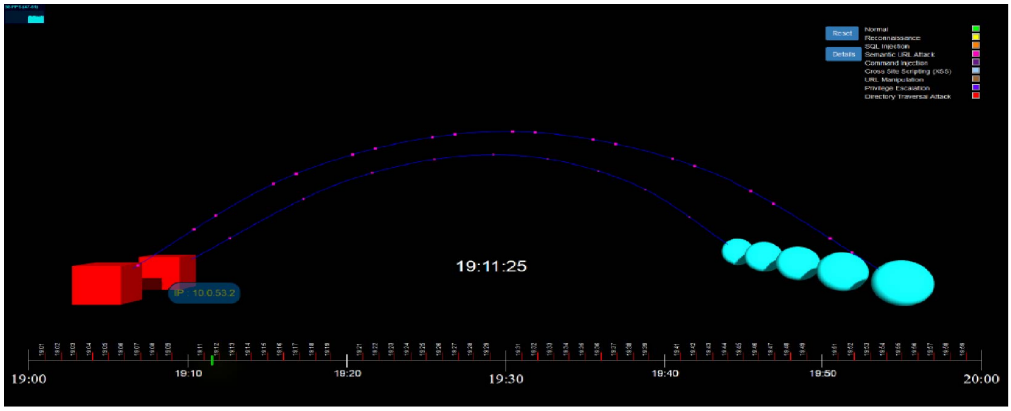

Fig. 5. Attack - Defend Team Visualization with Provenance Features.

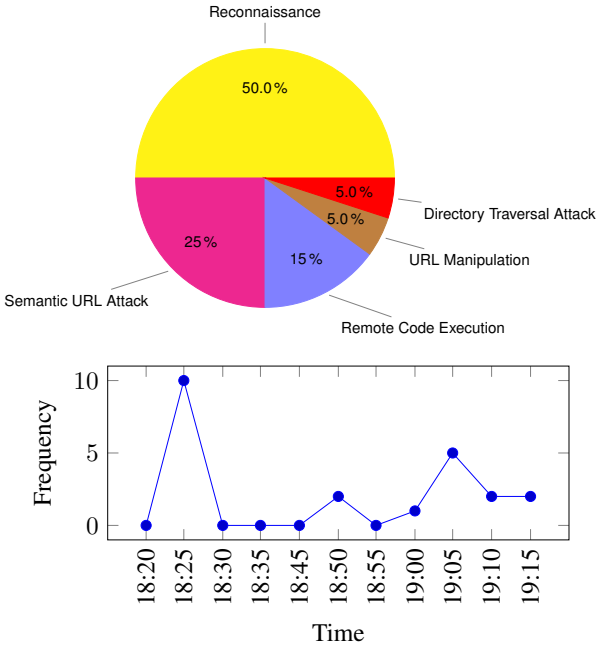

Fig. 6. Total Number of Attacks & Frequency of Attacks vs Time.

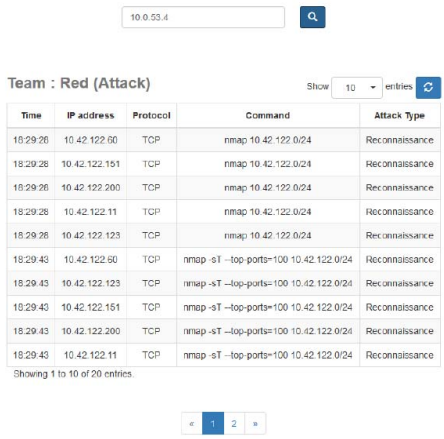

Fig. 7. Search Results Showing Type of Attacks Performed.

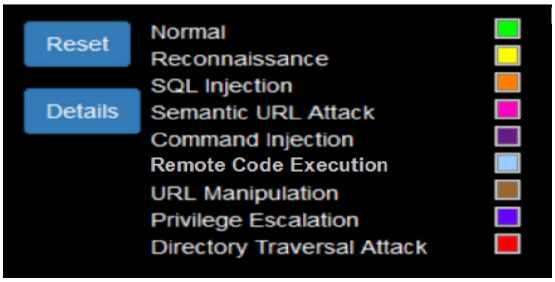

Fig. 8. Attack Color Categories.

D. User-Centricity with Augmented Reality

From re-imagining the environment through a mobile screen, to the state-of-the-art Microsoft HoloLens [2], recent advances in augmented reality [4] are offering new approaches for cyber security visualization. Multidimensional objects can be released from their traditional 2D prison and positioned in our world. The ability to see in real-time where attacks originated from (red team) or which machine is being targeted (blue team) can help to better understand attacks, and provide a sense of realism to these virtual threats. With a cyber security challenge, augmented reality provides spectators with a new medium to learn [32] and experience something that is typically hidden. This could also be deployed in industry as an awareness technique for the dangers of cyber-attacks, and used by cyber security personal to visualize their infrastructure. The WebGL visualization in Figure 5 can be moved into the actual lab environment with augmented reality as shown in Figure 12. Instead of computer symbols, these can be the real machines in the room. The paths between machines can then be shown, allowing users to follow attacks in real-time.

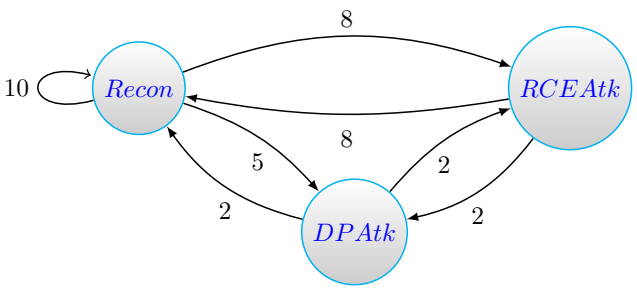

Fig. 9. A Attack Sample Process: Recon − > DPAtk − > RCEAtk.

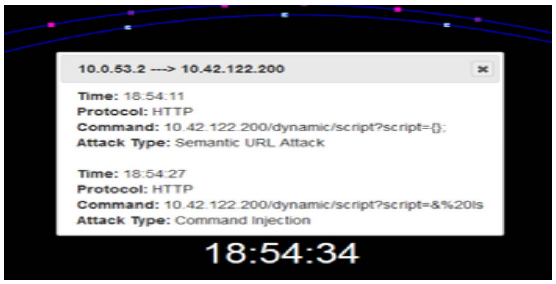

Fig. 10. Mouse-over Click to Display Attack Information.

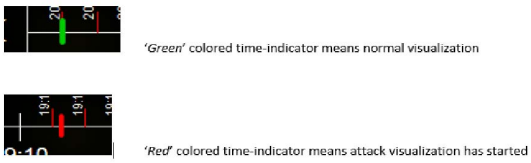

Fig. 11. Time-colored indicator of Attack.

The positioning of machines and identifying which physical machine associates with the log entries, along with different rooms for the two teams are current challenges. Related works on indoor positioning using known positions [23][22] or wireless signal strength [15][21] could be implemented. However the physical locations of the machines may need to be hard-coded, unless they are able to also learn their location automatically.

VI. Lesson Learnt: NZCSC Security Visualization

Overall, the data collection process is a challenging task. However, repeating the cyber security challenge competition yearly for the past 3 years, we were able to improve and tailor logging mechanisms according to what types of datasets required for academic research purposes and most importantly what we want to visualize. Other challenges include the backend and frontend implementation.

1) Backend Implementation Challenges: The concept of creating effective simple to use user-centric visualization is a challenging task. Creating effective security visualizations for targeted audiences, situation awareness requires thorough insights on designing the most interactive security visualization platforms. Factors contributing to high probability of a visualization platform being highly interactive depends on how well visualization designers understand the nature of the cyber-attacks, dataset type and structure, and who are the targeted audience.

Different log formats often create difficulties for certain databases, especially when dynamically reading in the data into allocated database tables. 'Transcribing' video logs for implementation verification and correlations between logs and user-input events is a tedious task.

2) Frontend Implementation Challenges: Understanding how WebGL works was the factor affecting how data has been rendered forward to the web browser. Integrating multiple programming languages and allowing them to communicate between each other were the major challenges for the security visualization platform. However, getting WebGL to link up with the backend based on the queries requested and picking which type of visualization should be used to visually display an attack was the challenge. Designing and implementing the security visualization with incorporating the concept of provenance into the real-time visualization became a time consuming factor of the entire visualization.

VII. Security Visualization Evaluation

A. Platform Evaluation

Security Visualization for cyber security challenge competitions have advantages and disadvantages. We are able to develop user-centric features allowing users to utilize the security visualization platform and gain most security insights from cyber security challenges. Such interactive user-centric features are: 'mouse-over clicks' with information details (see Figure 10), color-change indicators (see Figure 8 & Figure 11) to highlight different security events, and statistical visualization features (see Figure 6) to show number of attacks executed during the competition (see Figure 5). Based on these prototype, continuous implementations will be done for future cyber security challenges.

B. Logging and Attack Evaluation

The performance of the security visualization platform depends on many factors. These includes rendering methods, functions, proper use of visualization libraries and most importantly how and what data format is produced for the frontend to use for visualization. Comma-separated values (.csv) and JavaScript Object Notation (.json) data formats have enhanced the performance and how data is represented visually. Near real-time visualization effectiveness were depended on how well data are retrieved using searching algorithms prior to pushing them to the frontend for visualization. Dynamically, a 'constantGet' function constantly checks the database using Ajax [17] every second for new data inputs to visualize. Data provenance highlighted in the visualization platform with the use of timeline indicating the cyber security challenge duration and specifically highlight the exact time an attack is executed

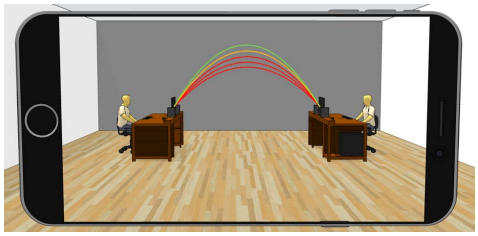

Fig. 12. Example of augmented reality, where a user is looking through a mobile device.

from the red team to the blue team (see Figure 11). Additional visualization features in identifying the source and destination of different attacks are made available with the ability to search for IP addresses using the search option on the visualization platform. Mouse-over clicks and pop-up information boxes helps users to interact effectively with the security platform. Users are able to click, snap and drag the visualization view around to clearly see interested attacks.

VIII. Conclusion

In this paper we proposed a user-centered security visualization infrastructure for the NZCSC, and outlined effective visualization techniques that attracts and captures users to effectively use security visualizations for insight retrieval in an event of cyber-attacks.

Our research goal is to 'visually connect the dots' between attack sources and destinations plus attack correlations between red with blue teams. Equally important is connecting the dots between the users visual perception and our security visualization platform allowing users to actively interact and understand cyber-attacks in a more realistic way. For future work we aim to add more user interactive features (mobile platform capabilities), forensic visualization features to analyze exploits, infected files and protocols.

Acknowledgment

The authors wish to thank the members of the Cyber Security Researchers of Waikato (CROW), Joshua Scarsbrook, Sam Shute, Cameron Brown and Meena Mungro. This research is supported by STRATUS (Security Technologies Returning Accountability, Trust and User-Centric Services in the Cloud - (https://stratus.org.nz), a science investment project funded by the New Zealand Ministry of Business, Innovation and Employment (MBIE)), and the University of Waikato.

References

[1] Darpa Goes Full Tron With Its Grand Battle of the Hack Bots.

[2] Microsoft HoloLens. Online https://www.microsoft.com/microsoft-hololens/en-us (Accessed 08/03/17).

[3] R. Baldwin. AI hackers will make the world a safer place – hopefully.

[4] M. Billinghurst, A. Clark, G. Lee, et al. A survey of augmented reality. *Foundations and Trends® Human–Computer Interaction*, 8(2-3):73–272, 2015.

[6] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel, and G. Vigna. Nexat: A history-based approach to predict attacker actions. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 383–392. ACM, 2011.

[5] R. S. Cheung, J. P. Cohen, H. Z. Lo, and F. Elia. Challenge based learning in cybersecurity education. In *Proceedings of the 2011 International Conference on Security & Management*, volume 1, 2011.

[7] G. Conti. Microsoft PowerPoint - dc12-conti-information-visualization.ppt - dc-12-conti.pdf.

[8] C. Cowan, S. Arnold, S. Beattie, C. Wright, and J. Viega. Defcon capture the flag: Defending vulnerable code from intense attack. In *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, volume 1, pages 120–129. IEEE, 2003.

[9] K. De Volder. Jquery: A generic code browser with a declarative configuration language. In *International Symposium on Practical Aspects of Declarative Languages*, pages 88–102. Springer, 2006.

[10] J. Dirksen. *Learning Three. js: the JavaScript 3D library for WebGL*. Packt Publishing Ltd, 2013.

[11] A. Doupé, B. Boe, C. Kruegel, and G. Vigna. Fear the ear: discovering and mitigating execution after redirect vulnerabilities. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 251–262. ACM, 2011.

[12] A. Doupé, M. Egele, B. Caillat, G. Stringhini, G. Yakin, A. Zand, L. Cavedon, and G. Vigna. Hit'em where it hurts: a live security exercise on cyber situational awareness. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 51–61. ACM, 2011.

[13] D. D. Dvorski. Installing, configuring, and developing with xampp. *Skills Canada*, 2007.

[14] C. Eagle and J. L. Clark. Capture-the-flag: Learning computer security under fire. Technical report, DTIC Document, 2004.

[15] F. Evennou and F. Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, 2006:164–164, 2006.

[16] J. Garae, R. K. Ko, and S. Chaisiri. Uvisp: User-centric visualization of data provenance with gestalt principles.

[17] J. J. Garrett et al. Ajax: A new approach to web applications. 2005.

[18] E. Gavas, N. Memon, and D. Britton. Winning cybersecurity one challenge at a time. *IEEE Security & Privacy*, 10(4):75–79, 2012.

[19] L. J. Hoffman, T. Rosenberg, R. Dodge, and D. Ragsdale. Exploring a national cybersecurity exercise for universities. *IEEE Security & Privacy*, 3(5):27–33, 2005.

[20] R. K. Ko and M. A. Will. Progger: an efficient, tamper-evident kernel-space logger for cloud data provenance tracking. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 881–889. IEEE, 2014.

[21] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[22] A. Mulloni, H. Seichter, and D. Schmalstieg. Handheld augmented reality indoor navigation with activity-based instructions. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pages 211–220. ACM, 2011.

[23] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg. Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing*, 8(2), 2009.

[24] T. Parisi. *WebGL: up and running*. " O'Reilly Media, Inc.", 2012.

[25] T. Reuille and A. Hay. us-14-Hay-Unveiling-The-Open-Source-Visualization-Engine-For-Busy-Hackers.pdf, 2014.

[26] Sakai. Open Cyber Challenge Platform - Research - Digital Forensics and Cyber Security Center at the University of Rhode Island, 2017.

[27] A. Shenoy and U. Sossou. *Learning Bootstrap*. Packt Publishing Ltd, 2014.

[28] G. Vigna. The 2010 international capture the flag competition. *IEEE Security & Privacy*, 9(1):12–14, 2011.

[29] G. Vigna, K. Borgolte, J. Corbetta, A. Doupe, Y. Fratantonio, L. Invernizzi, D. Kirat, and Y. Shoshitaishvili. Ten years of ictf: The good, the bad, and the ugly. In *3GSE*, 2014.

[30] R. Wang, D. Sun, G. Li, M. Atif, and S. Nepal. Logprov: Logging events as provenance of big data analytics pipelines with trustworthiness. In *IEEE Conference on Big Data*, 2016.

[31] J. Werther, M. Zhivich, T. Leek, and N. Zeldovich. Experiences in cyber security education: The mit lincoln laboratory capture-the-flag exercise. In *CSET*, 2011.

[32] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang. Current status, opportunities and challenges of augmented reality in education. *Computers & Education*, 62:41–49, 2013.