Frankenstack: Toward Real-time Red Team Feedback

Markus Kont
NATO Cooperative Cyber Defence Centre of Excellence
markus.kont[a]ccdcoe.org

Mauno Pihelgas
NATO Cooperative Cyber Defence Centre of Excellence
mauno.pihelgas[a]ccdcoe.org

Kaie Maennel
Tallinn University of Technology
kamaen[a]ttu.ee

Bernhards Blumbergs
NATO Cooperative Cyber Defence Centre of Excellence;
IMCS UL, CERT.LV Laboratory
bernhards.blumbergs[a]cert.lv

Toomas Lepik
Tallinn University of Technology
toomas.lepik[a]ttu.ee

***Abstract*—Cyber Defense Exercises have received much attention in recent years, and are increasingly becoming the cornerstone for ensuring readiness in this new domain. Crossed Swords is an exercise directed at training Red Team members for responsive cyber defense. However, prior iterations have revealed the need for automated and transparent real-time feedback systems to help participants improve their techniques and understand technical challenges. Feedback was too slow and players did not understand the visibility of their actions. We developed a novel and modular open-source framework to address this problem, dubbed *Frankenstack*. We used this framework during Crossed Swords 2017 execution and evaluated its effectiveness by interviewing participants and conducting an online survey. Due to the novelty of Red Team-centric exercises, very little academic research exists on providing real-time feedback during such exercises. Thus, this paper serves as a first foray into a novel research field.**

***Keywords*—automation, cyber defense exercises, education, infrastructure monitoring, real-time feedback, red teaming**

I. Introduction

Cyber defense exercises (CDX) are crucial for training readiness and awareness within the *cyber domain*. This new domain is acknowledged by NATO alongside with land, sea, air, and space [1]. Alliance nations are endorsing the development of both defensive and responsive cyber capabilities. In this context, the paper focuses on further evolving the quality and learning experience of CDX, aimed at developing cyber red teaming [2] and responsive skillset. Crossed Swords (XS) [3], a techical exercise developed by NATO Cooperative Cyber Defense Centre of Excellence (NATO CCD COE) since 2014, is used as a platform to create the proposed framework. The solution is applicable to any other CDX where standard network and system monitoring capability is available.

A. Background

XS is an intense hands-on technical CDX oriented at penetration testers working as a single united team, accomplishing mission objectives and technical challenges in a virtualized environment. While common technical CDX is aimed at exercising defensive capabilities (i.e., Blue Team – BT), XS changes this notion, identifies unique cyber defense aspects and focuses on training the Red Team (RT).

To develop and execute the exercise, multiple teams are involved: rapid response team (i.e., RT); game network and infrastructure development (Green Team – GT); game scenario development and execution control (White Team – WT); defending team user simulation (i.e., BT); and monitoring (Yellow Team – YT).

The RT consists of multiple sub-teams based on the engagement specifics, those being: network attack team, targeting network services, protocols and routing; client side attack team, aiming at exploiting human operator and maintaining access to the hosts; web application attack team, targeting web services, web applications and relational databases; and digital forensics team, performing data extraction and digital artefact collection. These sub-teams must coordinate their actions, share information and cooperate when executing attacks to reach the exercise objectives.

The main goal is to exercise RT in a stealthy fast-paced computer network infiltration operation in a responsive cyber defense scenario [4]. To achieve this, the RT must uncover the unknown game network, complete a set of technical challenges and collect attribution evidence, while staying as stealthy as possible. Note that XS is not a *capture-the-flag* competition, as the RT has to pivot from one sub-objective to another in order to achieve the final mission according to the scenario. Furthermore, Red sub-teams are not competing with each other, and rather serve as specialized branches of a single unit.

B. Problem Statement

Prior XS iterations revealed several problems with RT learning experience. Primarily, the YT feedback regarding detected attacks from the event logs and network traffic was presented at the end of every day, which was not well suited to the short, fast and technical nature of the exercise. The feedback session addressed only some noteworthy observations from the day, but RT participants need direct and immediate feedback about their activity to identify mistakes as they happen. This feedback needs to be adequately detailed, so that the RT can understand why a specific attack was detected and then improve their approach. Finally, to make the feedback faster, the slowest element in the loop—the human operator—needs to be eliminated.

Therefore, manual data analysis by the YT needs to be automated as much as possible. To achieve this, we used the same open-source tools as in the previous XS iterations, but added in event correlation, a novel query automation tool, and a newly developed visualization solution. We decided to call the framework *Frankenstack*. Fig. 1 illustrates the role of Frankenstack in the XS exercise.

978-1-5386-0595-0/17/$31.00 ©2017 IEEE

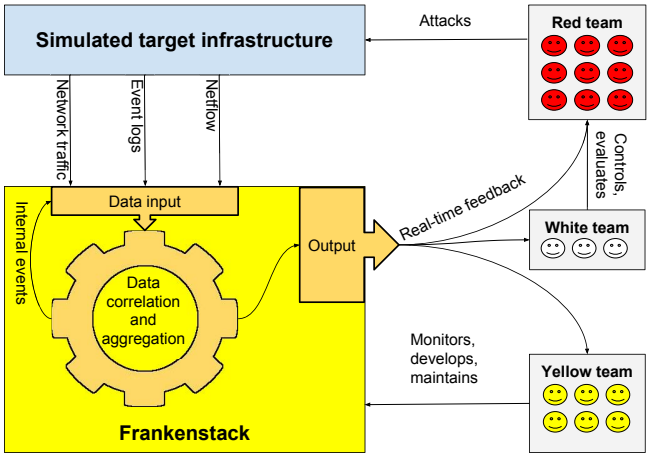

Fig. 1. High-level overview of Frankenstack

The RT has to receive timely and efficient feedback from the YT regarding detected attacks on the target systems. This feedback is critical to raise the level of stealthiness, identify the gaps of RT coordination, and analyze the tools and tactics used for computer network operations. The effectiveness of our framework was assessed during the main execution of XS 2017 (XS17), where the stack provided real-time monitoring feedback to the RT.

The remainder of the paper is organized as follows: section II provides an overview of related work, section III describes our monitoring stack, section IV presents RT feedback results, while section V discusses future work, and section VI concludes the paper.

II. Related work

For teaching purposes, the benefit of exercises and competitions is generally well accepted and documented [5], [6], [7], [8]. Unfortunately, not much research has focused on the perception of feedback which is provided to the training audience, especially in the context of monitoring technical indicators of compromise in realistic environments. Thus, this section presents research related to both measuring and improving the learning experience as well as situation awareness (SA) during cyber exercises.

Dodge et al. discussed CDX network traffic analysis in [9], a practice that is common in modern exercises not only for situational awareness (SA) but also as educational tool, for elaborating attacker campaigns, for training network analysts, etc. However, this early paper focuses on traffic capture and initial profiling, and does not consider distractions such as traffic generation, increasing infrastructure complexity, host instrumentation, data source correlation, or the need for immediate feedback. In [10], Holm et al. correlated network traffic and RT attack logs from Baltic Cyber Shield, a precursor for Locked Shields and Crossed Swords exercises. However, their goal was to improve existing metrics for vulnerability scoring, as opposed to participant education. Likewise, in [11], Brynielsson et al. conducted a similar empirical analysis on CDXs to profile attacks and create attacker personas.

In [12], Arendt et al. presented CyberPetri, a circle-packing visualization component of Ocelot, which was previously presented in [13] as a user-centered decision support visualization. They presented several use cases of the tool, but their main goal was high-level feedback to network analysts based on target system service availability reports. Although the tool was useful for high-level decision making, technical RT members are more interested in immediate effects of their attacks on target systems. Note that any single system is often a supporting pillar for more complex services, and is not noticeable to end-users. Nevertheless, modern security monitoring is built upon instrumentation of these systems, to find RT *footprints* and to trigger notification upon breaching these digital tripwires.

A paper [14] by Henshel et al. describes the assessment model for CDXs based on the Cyber Shield 2015 example, as well as integrated evaluation of metrics for assessing team proficiency. In addition to data collected during the exercise, they also conducted a pre-event expertise survey to determine possible relationships between prior expertise and exercise performance. For future assessments they suggest that near real-time analysis of the collected data is required—they stress that raw data collection is not a problem, but the capability to meaningfully analyze is the limiting factor. Manual methods do not scale with the huge amounts of incoming data. This closely coincides with our observations in section I-B and this is what we aim to improve.

Furthermore, existing academic research commonly relies on monolithic tools, which are often not accessible to the general public, thus, making experiments difficult, if not impossible, to reproduce. We seek to provide an inexpensive open-source alternative to these products. The next section describes our modular monitoring architecture.

III. Frankenstack

Commercial tools are too expensive for smaller cyber exercises, in terms of licensing fees, hardware cost and specialized manpower requirements. Detection logic in commercial tools is also not available to the general public, which hinders YT's ability to provide detailed explanations of detected attacks. Frankenstack is easy to customize as individual elements of the stack are industry standard tools which can be interchanged. Note that we opted to use a commercial tool *SpectX* as an element within Frankenstack for log filtering, due to on-site competency and developer support. However, this function could have been achieved with the open-source Elastic stack [15]. Our stack provides a clear point of reference to other researchers and system defenders who wish to compile the monitoring framework in their particular environments, as the overall architecture is novel.

The data available to us during XS included full ERSPAN (Encapsulated Remote Switched Port ANalyzer) traffic mirror from gamenet switches and NetFlow from gamenet routers. This was provided by the GT. Furthermore, we instrumented

gamenet systems to collect numerical metrics (e.g., CPU and memory usage, and network interface statistics) and logs (e.g., syslog from Linux, Event Logs from Microsoft Windows, Apache web server access logs, and audit logs from Linux command line executions). Such host instrumentations are very difficult to implement in a standard CDX with BT training focus: if the intent is to give BTs full control of a simulated infrastructure, then they also have full volition to disable these tools. However, as the XS training audience is the RT, then we could maintain control of all target systems and ensure a constant stream of monitoring data. Moreover, we complemented the list of BT data sources with various YT honeypots and decoy servers.

Detailed overview of the resulting stack, in relation to data processing pipelines, is presented in Fig. 2. The blue area represents available data sources, the gray area stands for data storage, and the yellow area denotes the YT presentation layer (i.e., visualization tools on five monitors). Blue and green elements represent target systems and all other elements outside colored boundaries are processing tools. Custom tools that we developed are highlighted with a dark yellow circle. Note that some tools, such as Moloch, are designed for both data storage and visualization, but are not presented in these respective areas because only their API components were used for processing automated queries.

We opted against using NetFlow data, as modern packet capture analyzers (e.g., Suricata, Bro, and Moloch) can fill this role, albeit by needing more processing power and memory. Additionally, these tools commonly present their output in textual log format, which we fed back into the central logging and correlation engine. Thus, the problem of identifying and displaying high-priority IDS alerts can be simplified into a log analysis problem.

Frankenstack uses event correlation for integrating various information sources as this field has been well researched in the context of log management [16], [17], [18]. We open-sourced the correlation ruleset in [19]. See Listing 1 for an example raw log entry from Snoopy Logger [20] that was converted into a more universal human-readable security event that could be presented to the general audience on various dashboards while preserving the raw message for anyone wishing to drill down. Note that specific IP addresses have been removed from this example. This generalization is necessary for handling and grouping subsequent log entries that continue describing the same event, e.g., additional commands executed on the same host via SSH.

Listing 1. Event generalization by frankenSEC

```
#INPUT
login:administrator ssh:(SRC_IP 58261 DST_IP 22)
username:administrator uid:1001 group:administrator
gid:1001 sid:6119 tty:(none) cwd:/home/administrator
filename:/usr/bin/passwd: passwd administrator

#OUTPUT
SRC_IP->[DST_IP]: Command execution by administrator
over SSH
```

Post-mortem analysis of available data sources has proven effective during prior CDXs for packet capture (PCAP) analysis, but requires a significant amount of time and manual work. Again, this clashes with the short time-frame of a CDX. Furthermore, search queries are often written ad hoc during investigations and subsequently forgotten, making analysis results difficult to reproduce. Thus, we created *Otta* [21], a novel query documentation and automation tool for periodically executing user-defined queries on large datasets and converting aggregated results into time-series metrics. *Otta* enables trend graphing, alerting, and anomaly detection for stored user-defined queries. This reduces time spent on analysis and ensures reproducibility by documenting the queries that produced the results.

We used various open-source tools for timelining metrics and log data, for displaying alerts, and presenting correlated information. There are slight differences in handling various incoming alerts. While many types of alerts (e.g., CPU and disk usage) trigger and recover automatically based on a set of thresholds, there are some types (e.g., IDS alerts) that lack the concept of a recovery threshold. Thus, the alert will never recover once raised, leading to an overabundance of information on the central dashboard. Furthermore, batch bucketing methods and timelines are lossy, as only the most frequent items are considered. The volatile nature of CDXs and an abundance of generated network traffic can therefore cause these dashboards to be too verbose to follow efficiently.

Attack maps are not usable because they rely on geographical data which is completely fictional in many CDX environments. Therefore, we developed Event Visualization Environment, or EVE, a novel web-based tool for visualizing correlated attacks in relation to gamenet infrastructure. The Alpha version of this tool has been made publicly available in [22]. EVE is a web application that shows attacks carried out by the RT in real time with a custom gamenet network map as background. Probes can send information to EVE listener in JSON format. Real-time visualization is using WebSocket technology—eliminating the need to reload the page for updated results.

EVE supports combining multiple events in a short time window, and that share the same source and destination addresses, into a unified attack. Resulting attacks are subsequently displayed as arrows connecting the circles around source and target hosts on the network map, while detailed attack information is displayed next to the network map. Using the gamenet map makes EVE a very intuitive tool for enabling participants and observers alike to comprehend CDX events on a high-level.

During the exercise EVE was available only to YT and WT members, as it revealed the entire exercise network map that RT had to discover on their own. However, EVE has a dedicated replay mode to display all the attacks condensed into a given time period, allowing participants to obtain an overview of the attacks as well as understand the pace and focus of the previous attack campaign. For instance, attacks from the previous day can be replayed in 15 minutes. EVE

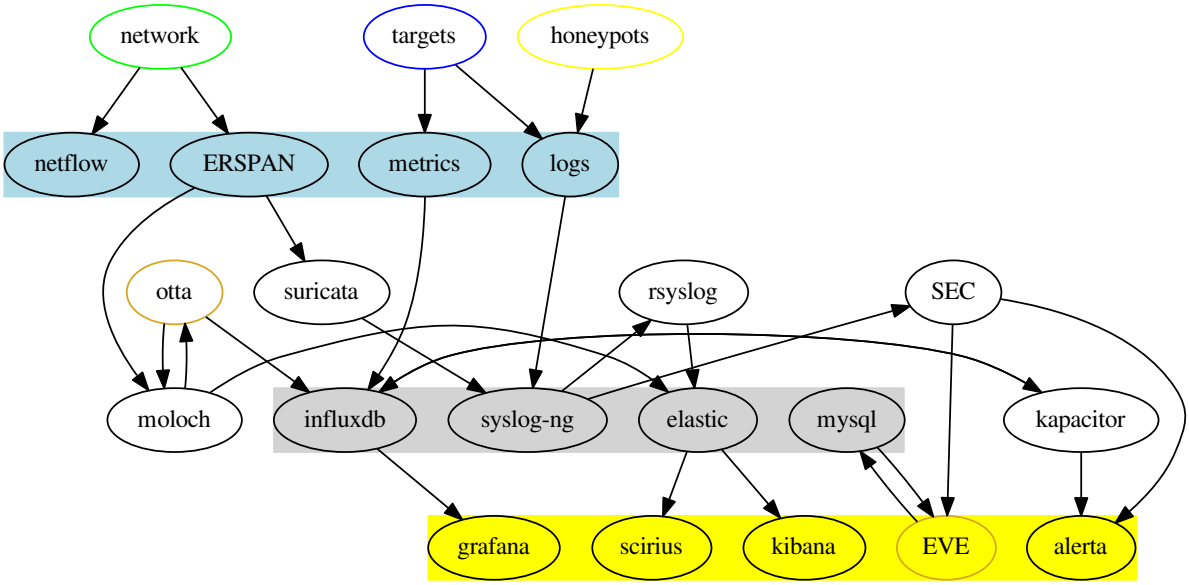

Fig. 2. Data flow between Frankenstack elements during XS17

TABLE I
DEDUPLICATION BY EVENT SOURCE

Event source	Total events	Unique events displayed	Percentage displayed
Apache	1908	35	1.83%
IDS	23790	616	2.59%
Snoopy logger	2962	40	1.35%
Total	*28660*	*691*	*2.41%*

was shown in replay mode to RT participants after the exercise concluded. This compressed replay was very effective in presenting the most prevalent problems, such as periodic beaconing during otherwise silent night periods and verbosity of particular network sub-team attacks.

Alerta [23] served as the primary dashboard to display alerts to the RT. We used the HTTP API for submitting Frankenstack events to Alerta. The RT had direct access to the Alerta web interface and could write their own filtering rules to view information relevant to their current campaign. Finally, we present Tab. I to illustrate how Frankenstack performed in deduplicating the events that were displayed to the RT on the Alerta dashboard. Note that deduplication was primarily based on the generalized event descriptions (see Listing 1).

IV. ASSESSMENT

The tools and infrastructure are essential for learning, but they do not make the exercise successful by default. Often human factors, such as how YT and RT members perceive and use the tools, have significant impact.

One essential part of the assessment was to observe the behavior of the RT members and their interaction with Frankenstack during the exercise in order to gain further insights into their progress and learning experience. We carried out qualitative interviews with RT participants, to estimate their reaction to Frankenstack and their overall learning progress. The interviews took place in casual settings during breaks in execution. Furthermore, we conducted a quantitative survey in the form of an online questionnaire. The survey consists of multiple choice or ranking style questions with the ability to provide additional comments for each question. The survey concluded by asking some general questions about meeting the training objectives and overall satisfaction with the exercise.

A. Feedback combined from interviews, survey and observations

This subsection includes the analysis of participants feedback. Improvement suggestions to learning design are presented in the following subsection IV-B.

We received 14 survey responses out of 27 participants (52%). 46% of participants had attended other exercises, but none of those exercises had attempted to provide SA via a similar toolset. The remaining 54% had not previously attended any exercise.

There were four large screens in the training room directed to the RT, displaying Alerta, Grafana, Scirius, and Suricata. A fifth screen displaying EVE was only visible to YT and WT members. Most RT members preferred to view the main screens displayed in training room, and 38% responded that they checked the screens every 60 minutes or less. Another 38% checked the screens every 30 to 50 minutes. RT members were not restricted from accessing any of the Frankenstack web interfaces. The survey revealed that learners did access the monitoring framework on their local computers when attempting new attack vectors. Thus, tools served their intended usage.

Alerta was considered most useful (46%), followed by Moloch (31%). There was no clear result for the least useful tool. The respondents expressed mixed feelings on the ease of

use of the SA tools: 38% equally agreeing and disagreeing, and the remainder (24%) being neutral.

Regarding learning impact, 79% agreed (of those 57% strongly agreed) that the SA given during exercise is useful for their learning process, while 21% were neutral. In terms of the feedback rate, 77% of the respondents considered the speed of feedback to be at the correct level, 15% considered it too slow and 8% considered it too fast. Furthermore, 57% agreed that alerts were accurate and sufficient for their learning process, while 43% were neutral about this question. However, several respondents revealed being too focused on achieving their primary objectives, and thus unable to properly switch between their tools and feedback screens.

In relation to visibility, 45% of the participants agreed that they had learned a lot about how their actions can be detected (i.e., it is useful to see simultaneously what attack method could be detected, and how), and 30% were more careful with their attacks and thus tried to be stealthier than they normally would have been. However, there were some unintended side-effects. The feedback sometimes provided insight into the network map that the RT was tasked to discover independently. For example, if the RT probes a yet unknown node on a network, the logs generated on the host might reveal the target hostname (e.g., *sharepoint* or *ftp*), which consequently implies the purpose of the system—something that would not be apparent from an IP address. Thus, there is a fine line between revealing too little or too much to the training audience.

Furthermore, some comments revealed a loss of emphasis on stealth due to exercise time constraints, i.e., RT members knowingly used more verbose techniques closer to the objective deadline. To clarify, 64% of respondents confirmed that the SA tools were not distracting them nor had negative impact, while 30% agreed that they were distracted. The remaining 6% were neutral. This confirms the challenges of providing instant feedback, as the learning potential is not fully used. The question is how this learning experience is impacting long-term behavior of the participant.

One of the key training aspects is working as a team in achieving goals. Thus, team communication and cooperation are vital. Overall, 83% of respondents indicated some improvement of the skills for these specific training objectives. However, feedback concerning the impact of SA tools on team communication and cooperation is mixed—50% perceived positive impact, whereas 21% were negative and remainder were neutral. Several respondents acknowledged less need for verbal communication, as they could see relevant information on the screens. Unfortunately, not all RT members were able to interpret and perceive this information correctly. This combined with the reduced need for communication meant that not all participants progressed as a team.

Compared to other CDXs, 50% responded that they needed less information from YT members, as they obtained relevant SA on their own. Guidance, however, is a critical success factor for learning, especially in a team setting. 64% of participants said they had sufficient help for their learning process, i.e., when they did not know how to proceed, their team members or sub-team leaders provided guidance. However, 64% is a rather disappointing result and could clearly be increased with improved learning design. Some respondents admitted that they did not know how other teams were progressing and wasted time on targets that were not vulnerable. This caused significant frustration and stress, especially when combined with the compressed timeframe of a CDX.

B. Learning improvement suggestions

Given the amount of work that goes into preparing such exercises, the level of learning potential needs to be maximized. Our analysis suggests that small learning design changes may have significant impact. This section presents the main recommendations derived from these results.

From the learning perspective, we cannot assume that participants know how to use or interpret the results. Lack of in-depth knowledge of monitoring tools (e.g., where is raw data collected, what is combined and how, what needs to be interpreted in which way, etc.) has a negative impact on learning. A dedicated training session or workshop needs to take place prior to execution. Furthermore, in the light of the survey results, inclusion of various tools into Frankenstack needs to be carefully evaluated to avoid visual distractions for RT participants. There is also a need to reduce prior system and network monitoring knowledge by making the output more self-explanatory.

Given the difficulties in switching between multiple screens whilst also trying to achieve an objective in unfamiliar network, one can easily suggest compressing the amount of presentable information to reduce the number of monitoring screens. However, this cannot be attained without reducing the amount of technical information. The purpose of Frankenstack is not to provide SA to high-level decision makers, but to present feedback to technical specialists. Thus, a better approach would be restructuring each sub-team with a dedicated monitoring station with a person manning it, allowing team members to focus on their objectives and get feedback relevant only for their actions. As such, RT members must be given a *hands-on* opportunity to use monitoring systems.

In RT exercises such as XS, there are several main objectives to be achieved by the whole RT. It is challenging to evaluate reaching objectives, since there are many steps involved in reaching a specific objective. Often the tasks or sub-objectives are divided between sub-teams (network, web and client-side) and between individuals in those sub-teams. The difficulty of a specific exploitation depends on the individual's skillset, which varies widely. Hence, there is a trade-off between assigning a task to an experienced member to increase the chance of success, versus teaching a new member. For example, an experienced network administrator is more effective in exploiting network protocols and is likely less visible while doing so, but may not learn anything new.

Discussions and feedback revealed that several respondents felt they were stuck and working alone. Division of the tasks between sub-teams and individuals also diminishes the learning potential. One training design option to alleviate this issue

would be regular *team timeouts* for reflection. Reflective team sharing is crucial for the learning success of each individual, and would overcome the project management approach where each team member focuses only on personal objectives. Higher emphasis should be on offering tips and helping those stuck on an objective to move forward whilst also keeping track of the feedback provided by Frankenstack. The coaching could also be handled in the form of a *buddy system* where RT members are not assigned a sub-task individually, but in groups of two or three. They would then have to share their knowledge and can benefit from different individual backgrounds.

Finally, it is important to have better time-planning during the execution. While it is certainly appropriate to allow for flexibility in the paths that the RT can take to solve the objectives, participants should avoid spending too much time on wrong targets. Nevertheless, the learning impact of the exercise in this format (i.e., with real-time feedback) is very positive. Only 13% of all participants' responses reported no significant change in their skills, while an overwhelming 87% perceived an improvement in their skill level, and 93% agreed that they were satisfied with exercise.

V. Future work

We encountered several unforeseen problems, as methods for assessing technical RT campaigns have to be incorporated into the game scenario itself. However, most XS17 targets had already been developed before the initial stages of this research. We plan to increase information sharing between Red and Yellow teams to improve RT progress measurement. Thus, we can develop better assessment methodologies for RT skill levels and YT feedback framework.

Development of a new dynamic version of EVE is already underway for the next XS iteration. In addition to the network map view, it can draw the network map dynamically as RT compromises new targets. Currently, EVE can only be used after the end of the exercise. However, in addition to providing more actionable alerting, the new version can also reduce RT work for mapping new systems and allow them to focus on the technical exercise.

VI. Conclusion

In this paper, we have presented the core challenges in organizing a CDX with Red Team emphasis, such as timeliness and accuracy of feedback, and ensuring participant education without compromising the game scenario. We compiled a novel stack of open-source tools to provide real-time feedback and situational awareness, and conducted surveys among the RT members to assess the effectiveness of this method.

Frankenstack feedback regarding learning impact was mainly positive. However, there are critical questions to answer when designing the RT exercises, such as what is the right balance of information to provide to the RT, does the behavior change due to monitoring or information visible (i.e., learners unconsciously limit themselves by not trying out more risky strategies, etc.). Also, some further learning design changes, and not necessarily only limited to SA, can maximize the return on the significant investment into preparing such RT exercises. We hope to spark a discussion on improving these problems.

VII. Acknowledgments

The authors would like to thank Mr. Risto Vaarandi, Mr. Hillar Aarelaid and Prof. Olaf M. Maennel for their valuable contributions. This work has been supported by the Estonian IT Academy (StudyITin.ee).

References

[1] T. Minárik, "NATO Recognises Cyberspace as a Domain of Operations at Warsaw Summit," Available: https://ccdcoe.org/nato-recognises-cyberspace-domain-operations-warsaw-summit.html.

[2] P. Brangetto *et al.*, "Cyber Red Teaming - Organisational, technical and legal implications in a military context," NATO CCD CoE, Tech. Rep., 2015.

[3] "Crossed swords exercise," Available: https://ccdcoe.org/crossed-swords-exercise.html.

[4] P. Brangetto *et al.*, "From Active Cyber Defence to Responsive Cyber Defence: A Way for States to Defend Themselves Legal Implications," Available: https://ccdcoe.org/multimedia/active-cyber-defence-responsive-cyber-defence-way-states-defend-themselves-legal.html.

[5] B. E. Mullins *et al.*, "The impact of the nsa cyber defense exercise on the curriculum at the air force institute of technology," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, Jan 2007, pp. 271b–271b.

[6] A. T. Sherman *et al.*, "Developing and delivering hands-on information assurance exercises: experiences with the cyber defense lab at umbc," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, June 2004, pp. 242–249.

[7] R. C. Dodge *et al.*, "Organization and training of a cyber security team," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 5, Oct 2003, pp. 4311–4316.

[8] G. H. Gunsch *et al.*, "Integrating cdx into the graduate program," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 5, Oct 2003, pp. 4306–4310.

[9] R. C. Dodge and T. Wilson, "Network traffic analysis from the cyber defense exercise," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 5, Oct 2003, pp. 4317–4321.

[10] H. Holm *et al.*, "Empirical analysis of system-level vulnerability metrics through actual attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825–837, Nov 2012.

[11] J. Brynielsson *et al.*, "Using cyber defense exercises to obtain additional data for attacker profiling," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Sept 2016, pp. 37–42.

[12] D. Arendt *et al.*, "Cyberpetri at cdx 2016: Real-time network situation awareness," in *2016 IEEE Symposium on Visualization for Cyber Security (VizSec)*, Oct 2016, pp. 1–4.

[13] D. L. Arendt *et al.*, "Ocelot: user-centered design of a decision support visualization for network quarantine," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, Oct 2015, pp. 1–8.

[14] D. S. Henshel *et al.*, "Predicting proficiency in cyber defense team exercises," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Nov 2016, pp. 776–781.

[15] "Elastic stack," Available: https://www.elastic.co/.

[16] R. Vaarandi *et al.*, "Simple event correlator - best practices for creating scalable configurations," in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2015 IEEE International Inter-Disciplinary Conference on*, March 2015, pp. 96–100.

[17] R. Vaarandi, "Platform independent event correlation tool for network management," in *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*, 2002, pp. 907–909.

[18] ——, "Sec - a lightweight event correlation tool," in *IP Operations and Management, 2002 IEEE Workshop on*, 2002, pp. 111–115.

[19] "Frankensec," Available: https://github.com/ccdcoe/frankenSEC.

[20] "Snoopy Logger," Available: https://github.com/a2o/snoopy.

[21] "Otta," Available: https://github.com/ccdcoe/otta.

[22] "Eve - event visualization environment," Available: https://github.com/ccdcoe/EVE.

[23] N. Satterly, "alerta," Available: http://alerta.io/.