Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Serious games as a tool to model attack and defense scenarios for cyber-security exercises

Muhammad Mudassar Yamin, Basel Katt, Mariusz Nowostawski*

Norwegian University of Science and Technology, Gjøvik 2815, Norway

ARTICLE INFO

Article history:
Received 30 March 2020
Revised 8 May 2021
Accepted 16 August 2021
Available online 20 August 2021

Keywords:
Cyber range
Cyber-security
Exercises
Scenarios
Attack
Defense

ABSTRACT

Technology is evolving rapidly; this poses a problem for security specialists and average citizens as their technological skill sets are quickly made obsolete. This makes the knowledge and understanding of cyber-security in a technologically evolving world difficult. Global IT infrastructure and individuals' privacy are constantly under threat. One way to tackle this problem is by providing continuous training and self-learning platforms. Cyber-security exercises can provide a necessary platform for training people's cyber-security skills. However, conducting cyber-security exercises with new and unique scenarios requires comprehensive planning and commitment to the preparation time and resources. In this work, we propose a serious game for the development of cyber-security exercise scenarios. The game provides a platform to model simulated cyber-security exercise scenarios, transforming them into an emulated cyber-security exercise environment using domain-specific language (DSL) and infrastructure orchestration. In this game, players can play as cyber attackers or defenders in a multiplayer environment to make operational cyber-security decisions in real-time. The decisions are evaluated for the development of operational cyber-attack and defense strategies.

1. Introduction

During the European cyber-security challenge (ecs, 2019), we found that the teams involved were facing problems in strategizing their approach for solving cyber-security exercise scenarios. The team members had a sufficient level of technical skills to tackle the technical problems present in the challenge, but their decision-making skills in prioritizing the best moves were lagging. One way to overcome this issue would be by conducting many operational cyber-security exercises with unique scenarios, such that the exercise participants could get the right level of experience in decision making. However, conducting such exercises is resource intensive and time-consuming (Yamin and Katt, 2018). Therefore, in the current research, we investigate an efficient way to conduct cyber-security exercises that can help exercise the participants' skill improvement.

Through a review of the literature (Amorim et al., 2013; Hendrix et al., 2016; Schreuders and Butterfield, 2016), we identified that serious games are actively used in cyber-security skill development. However, one problem with such games is related to the static level design, which makes the integration of new and unique scenarios difficult. Another problem is that many are turned-based games as opposed to real-time strategy games. Real-time strategy games enhance players' cognitive flexibility (Glass et al., 2013) and help in training for the dynamic nature of cyber-security scenarios. To address

* Corresponding author.
E-mail addresses: muhammad.m.yamin@ntnu.no (M.M. Yamin), basel.katt@ntnu.no (B. Katt), mariusz.nowostawski@ntnu.no (M. Nowostawski).
https://doi.org/10.1016/j.cose.2021.102450

these issues and challenges, we ask the following research questions:

1. How can serious games be used to model dynamic cyber-security exercise scenarios in a realistic manner?
2. How can modeled cyber-security exercises be used in devising cyber attack and defense strategies in a realistic manner?
3. Is it efficient to conduct cyber-security exercises in a simulated modeled environment for exercise participants' skill improvement?
4. Is it efficient to transform simulated cyber-security exercise scenarios in a game to an emulated infrastructure, and how usable, flexible, and complete is it?

To answer the research questions, we developed (1) a real-time cyber-security strategy game that enhances players' cognitive flexibility and (2) a cyber-security exercise scenario domain-specific language (DSL) to model the scenario and generate the emulated infrastructure. The game is thoroughly assessed using surveys given to a large group of participants during the Norwegian cyber-security challenge 2019 (sta, 2019); the findings of this survey are presented in this article.

The rest of the current article is organized as follows: First, we share the research background and key concepts of cyber-security exercises. Then, we proceed with sharing a brief related work on serious games in cyber-security. Continuing this, we will state our research methodology, present our cyber-security strategy game with the developed DSL, and their assessment and evaluation. Finally, we conclude the article with a discussion and conclusion.

2. Research background

The importance of cyber warfare training is critical when considered in light of contemporary examples, such as the cyber-attacks on Estonia and the crippling of Georgia's government websites using advanced hacking methods. Cyber-warfare was first deployed during *Operation Desert Storm* against Iraq. The communication networks of the Iraqi forces were crippled, so they were forced to use less-secure microwave communication, which was easily intercepted and led to their eventual defeat. The Bosnia–Herzegovina war saw the use of cyber warfare to cripple governmental infrastructure to such an extent that the paramilitary force was turned against the actual military. In the 1990s, the U.S. (United States) government realized that they were vulnerable to cyber-attacks; they had been using offensive cyber capabilities to achieve their tactical and strategic objectives, which resulted in a similar response from other state actors.

Norway is one of the leading digital nations in the world. The government encourages public and private sectors to take part in digital innovations for the country's progress. Digital infrastructure is challenged by many factors, including the existence of complex vulnerabilities that can be exploited by advanced cyber-security attacks, along with the need to secure the provision of successful digitalization solutions. Norway was the first country to make a cyber-security strategy in 2003 and then revised it in 2007 and 2012 (cyb, 2012). A follow-up report was also published, emphasizing responsibility for securing digital assets. The present strategy is Norway's fourth cyber-security strategy, and its purpose is to address the challenges of digitalization faced by Norwegian society (Nat, 2020). While designing strategies, stakeholders from the public and private sectors are taken into account.

More than 300 delegates took part in the formulation of Norway's present strategy. The introduction deals with the challenges, strategy, vision, and strategic goals. In the introduction, it is noted that Norwegian society has become immensely digitalized to benefit private individuals, companies, and authorities. Here, the challenge of digitalization is cyber threats and the thorough dependency of society on digitalization. Digital infrastructures and systems are growing more complex, global, and integrated. All kinds of devices are being connected to the Internet. The fast speed of digitalization makes it challenging to forecast the resulting threats. Some threats include ransomware, industrial espionage, sabotage, blackmail, cyber-bullying, and identity theft (Nat, 2020). The strategy aims to address cyber-security issues, but to do this, the appropriate authorities must give access to a broad range of tools, along with the development of regulations and knowledge of supervisory activities. A follow-up report (lis, 2021) was released, highlighting the 50 steps taken by the government to implement the national cyber-security strategy. Step 27, 41, and 42 are as follows:

- Development of the Norwegian Cyber Range (NCR), which will be the first national test arena for cyber-security.
- Conducting *National cyber-security exercise*.
- Participation in international exercises such as NATO coalition, Locked Shields, Cyber Europe, and NATOs CMX.

2.1. *Cyber-security training and exercises*

There is a constant need for training and self-learning platforms to achieve the stated objectives for cyber-security education. Cyber-security threats are on the rise, so the field of cyber-security education is emerging to train the next generation of cyber-security professionals (Ford et al., 2017). However, the cyber-security field faces a skills gap problem because of the nature of the rapidly changing cyber-security environment (Endicott-Popovsky and Popovsky, 2014). This situation makes it difficult to train and educate the next generation of cyber-security professionals. There are two main types of cyber-security exercises. The first is table-top discussions, and the second is practical hands-on operation-based exercises (Gurnani et al., 2014). Table-top exercises are discussion based and conducted in the form of seminars, workshops, and idea exchanges mostly related to policy-oriented issues. In comparison, most operation-based cyber-security education and training programs employ hands-on activities aiming to improve the exercise participants' technical skills and abilities. These cyber-security exercises are executed in simulated, emulated, physical, or hybrid practice environments. Recent studies have identified that the required practice environments are being developed via manual setup and configuration, a methodology that is ineffective, tedious, and error

prone (Beuran et al., 2018). These practice environments are known as cyber-ranges.

According to Pham et al. (2016), cyber-ranges are well-defined controlled (virtual) exercise environments that are used in cyber-security training to efficiently help trainees gain practical knowledge through hands-on activities. Creating these exercise environments that contain the necessary features such as network topology, virtual machines, and security-related content is not an easy task (Beuran et al., 2018; Yamin and Katt, 2018). Many cyber-ranges try to automate the creation of these exercise environments, such as with Cytrone (Beuran et al., 2018), CyberVAN (Chadha et al., 2016), Cyris (Pham et al., 2016), Telelab (Casini et al., 2003), and Secgen (Schreuders et al., 2017). Like the natural environment in which animals and plants interact with the environment to utilize its resources, these exercise environments need to be interacted with to utilize the resources in them. These interactions can be done in the form of cyber-security exercises, an assessment of new technologies, a vulnerability assessment, malicious activity profiling, security data generation, and so forth. Individuals and teams on a cyber-range perform these interactions. In terms of operation-based cyber-security exercises, these teams include the following:

1. White team: A team that creates or generates a cyber-security exercise environment.
2. Red team: A team that attacks the cyber-security exercise environment.
3. Blue team: A team that defends the cyber-security exercise environment.

Multiple additional teams are also part of cyber-security exercises and can include Green, Orange, Yellow, and Purple teams, which we have explained in our previous work (Yamin et al., 2019). Their involvement solely depends on the scale and objectives of an exercise. However, in the current work, we are only focusing on the White, Red, and Blue teams. These teams are primarily involved in three main types of cyber-security exercises:

1. Cyber-attack exercise: theses exercises are conducted to train, assess, and evaluate the performance of red teams. An environment is created by a white team in which red teams need to achieve specific objectives to compromise the exercise environment in a particular time period.
2. Cyber-defense exercise: these exercises are conducted to train, assess, and evaluate the blue team's performance. A white team creates an exercise environment. A blue team needs to investigate and prevent cyber-attacks by red teams and to prevent these attacks within a particular time period.
3. Cyber-attack/defense exercise: these exercises are conducted to assess and evaluate red and blue teams' performance at the same time. A white team creates an exercise environment in which active engagement between red and blue teams occurs to simultaneously attack and defend an exercise environment.

Based on our research findings (Yamin et al., 2018), we have identified that automation can reduce the time requirements for cyber-security exercises. For this, serious games could help (Hendrix et al., 2016)*to overcome the inefficiencies in cyber-security exercises*. The gamification of cyber-security exercises is a recent trend in which participants are divided into teams for achieving a specific objective like finding flags. The participants' strategies to solve the problems like *Capture The Flag (CTF)* in a cyber-security exercise scenario are very difficult to model because of the real-time decision-making of exercise participants. This makes the decision tree that is involved in such problem solving very complex. To address this, we propose a real-time cyber-security strategy game. Players will have the ability to play as an attacker or defender in a real-time multiplayer environment. Resources are assigned to attackers and defenders based on the scenario requirement, and their actions are recorded and observed by an observer. A detailed scenario creator is developed in which experts model the scenario in the game that can be transformed into an emulated environment. This results in a dynamic generation of attack and defense trees generated during the real-time cyber-security strategy game.

3. Related work

In the related work section, we provide a brief overview of serious games developed and used for cyber-security exercises and the methods for cyber-security scenario modeling.

3.1. *Games for cyber-security exercises*

In 2016, Hendrix et al. (2016) conducted a detailed survey of serious games for cyber-security education. They identified 15 games from industry and 14 games from academia that are actively being used for this purpose. Next, they categorized the games by their types like 2D point and clicked turn-based scenarios, 3D virtual world (sims style), and enterprise contingency planning. The games' target audiences comprises science curriculum students, children, and teenagers. The researchers stated that these games are used for training and education for short-term purposes only. For long-term training and education, scenario-based games are required. These scenario-based games represent unique case studies that can help in case-based learning.

In 2016, Alotaibi et al. (2016) conducted a review of serious games for cyber-security awareness. They identified 12 academic research articles and nine serious games being used for cyber-security education and awareness. These games had shown positive results in the evaluation of their effectiveness in cyber-security education and awareness. However, a large population set is needed in future research to better understand their impact. Moreover, the games currently being used deal with general cyber issues; there is a need to develop games that can be used in training specific scenarios.

In 2016, Schreuders and Butterfield (2016) conducted a two-year study on gamification for teaching and learning computer security in higher education. The study aimed to improve student engagement, increase student experience, and the content coverage of education material. They used freely available security educational games with in-house developed solutions for measuring students' progress with semi-

autonomous evaluations. The authors identified that games could be useful for initial motivation and student engagement; however, with time, students' motivation and engagement levels tend to decrease. In terms of increasing positive student experiences and content coverage, the study yielded positive results. The authors stated that gamification approaches work well when no extensive task-based assessment is involved in the education and training process.

In 2013, Amorim et al. (2013) proposed gamification as a new cyber-security education and training approach. They stated that the new approach should be a model-driven approach for the agile development of cyber-security exercises. The authors further stated that in terms of simulation and emulation, exercise execution depends on the training needs. The effectiveness of training exercises can be assessed with performance support systems. Their research was concluded by stating that cyber-security training requirements change with the technology. Therefore, new content and material for training exercises are continuously required, and model-driven agile development techniques can achieve this.

Adam Shostack (Ada, 2020) maintained an online list of table-top cyber-security games used for educational purposes. These games were mostly board and card games played between multiple players to learn different cyber-security concepts in a fun and engaging manner. As of June 2020, the list contained 28 games for security educational purposes, one game for teaching privacy principles, three non-game decks, and table-top games with some additional resources. These games did not require any software to play.

3.2. *Methods for cyber-security scenario modeling*

Cheung et al. (2003) presented CAML *(correlated attack modeling language)*, which uses a module of small attack steps to create a cyber-security attack scenario. The modules were designed to be very generic so that they could be used to model different cyber-attack scenarios. The researchers divided an attack model scenario into four parts: *vulnerability*, *exploit, attack step*, and *composite attack*. *Vulnerability* is the condition in the system or procedure that enables an adversary to perform actions that violate the security of the system and procedure, while *exploitation* is the process of exploiting a single vulnerability. *Attack steps* are the actions of the adversary for achieving specific goals, while *composite attack* combines multiple attack steps. The researchers' attack modeling methodology considered the attacker's goal and sub-goals, developing a relationship between attacker goals and the coresponding system events that can be observed to detect an attack.

Liu et al. (2005) presented AIOS *incentive-based modeling and inference of attacker intent, objectives, and strategies*. They integrated attacker intent regarding the cost of action to model the attackers' objectives. They also developed a game-theoretic AIOS formalization to capture the inter-dependencies between attacker intent, objective, strategies, and defender objective, along with the strategies to deduce AIOS automatically. They applied the developed AIOS on a real-world DDoS scenario to validate AIOS effectiveness in modeling attack and defense scenarios.

Marshall (2009) presented CyberSMART i.e.(*cyber scenario modeling and reporting tool*). They divided the cyber-security exercise into three tracks: *description and objectives*, *gamespace*, and *scenario*. *Description and objectives* define the scope of the exercise and what learning outcome is expected to be achieved. *Gamespace* defines the exercise environment and networking topology on which the exercises are planned to be executed. In comparison, the *scenario* defines the set of events expected to happen to achieve an objective. The researchers argued that there might be multiple objectives and sub-objectives in a cyber-security exercise, so there would be multiple scenarios to achieve those objectives. The authors proposed an event-based pyramid model for the representation of exercise objective and the corresponding scenarios.

Shiva et al. (2010) applied game theory concepts to dynamic cyber-security scenarios and considered the interaction between attackers and defenders in the cyber-security scenario as a game. The researchers suggested a model with rewards and punishments for the adversaries' actions. The model works by considering the *Nash equilibrium* as a key defining point for defender strategies. The defenders try to reach the *Nash equilibrium* to win against the attacker, while the attacker tries to avoid the *zero-sum* state in the game. The attacker receives a payoff if they can avoid a zero-sum state, and the whole game continues.

Russo et al. (2018) presented *scenario design and validation for next generation cyber ranges*, in which they proposed a model to design, validate, automatically generate, and test cyber-security scenarios. The researchers introduced scenario description language *SDL*, which is used to model the scenarios. The SDL has 10 elements: *system, firewall, policy, software, user, principal, vulnerability, file, invariant*, and *goal*. In the scenario, *principals* represent the subjects operating in the system while a *goal* is the objective of the *principals*. The researchers executed the SDL on a cloud orchestration platform for scenario deployment and validation.

4. Research methodology

Numerous different research methods were employed in the current work, including serious game development methodology, ontology development methods, and model-driven engineering methodology. The last one includes the development of the DSL and its compiler. Furthermore, various quantitative and qualitative assessment methods for evaluation were used. First, for the development of serious games, we used the framework in cyber-security proposed by Le Compte et al. (2015). The framework provides a precise methodology for conducting research related to serious games in cyber-security. The framework has six steps for the development life cycle and evaluation of serious cyber-security games: (1) Preliminary analysis in which the available resources for game development are evaluated, pedagogical objectives are defined, the target audience is identified, and the game mechanics are defined based on the pedagogical objectives. (2) The design phase is responsible for the game's conceptual modeling, ensuring that the game objectives are well conveyed to the players. (3) The development phase aims at developing the game based on the resources and objectives identified before. (4) Game assessment evaluates the game, in which a test group of a target audience can be used in the

game assessment process, and the feedback will then be used to improve the game mechanics. (5) The deployment phase is the one responsible for deploying the game for real-world assessment and training. The final phase is the (6) the player assessment phase, in which the game's effectiveness in achieving its pedagogic objectives is measured. This can be achieved through tests, surveys, and questionnaires given to game players.

Second, to model the various concepts present in the cyber-security exercise scenarios, we carefully analyzed the cyber-security exercise domain and developed an ontology (Maines et al., 2015). This ontology highlighted various abstract concepts related to cyber-security exercises that must be incorporated in the DSL; these are presented in Fig. 10. For the development of the DSL, we employed model-driven engineering (Schmidt, 2006) techniques. These techniques are used to develop the scenario language and its syntax and then to develop a compiler for the language that will process an instance of the scenario language and generate various usable artifacts, such as *HEAT* (hea, 2019) and *Puppet* (Pup, 2020) templates, which can be used to generate the exercise infrastructure.

For the verification and assessment of our developed artifacts, we employed both quantitative and qualitative evaluation methods. We created a cyber-security exercise scenario based on a real penetration testing activity, along with using pre- and post-exercise survey methods (Yamin et al., 2018) to quantify and measure the skill improvement of the exercise participants. For the qualitative evaluation, we used expert feedback against a set of four predefined evaluation matrices: *efficiency*, *usability*, *completeness*, and *flexibility* that we identified from the literature (Yamin et al., 2019).

5. Proposed system

As we have argued, the current way of conducting cyber-security exercises is not efficient; therefore, we are proposing a system that addresses one of the most time-consuming parts of the cyber-security exercise life cycle (Yamin and Katt, 2018): the preparation of an exercise scenario. Furthermore, a dry run is partially covered as well. In the preparation of a scenario, a ***White Team*** creates the environment in which the ***Red Team*** and the ***Blue Team*** practice their attack and defense skills. We identified the major cyber-security scenario definition techniques (Yamin et al., 2019) in which a scenario definition language is used for the orchestration of the cyber-security exercises infrastructure. Our proposed system utilizes the concept of *SDL*; however, we are proposing a fundamentally new way of creating and deploying a scenario. Our proposed system has three primary parts:

1. *Cyber security strategy game*
   The game is used to model the network topology for a cyber-security exercise scenario. The games provide an interface for presenting high-level scenario requirements and transforming them into low-level technical requirements. The game is basically designed to facilitate the process of cyber-security exercise scenario modeling and validation in a simulated environment by incorporating the roles of the Red and Blue Teams. The game provides an opportunity for the scenario designer to develop and test hundreds of cyber-security scenarios before deploying them in a realistic, emulated environment.
2. *Domain specific language*
   The DSL is used to represent the low-level technical details present in the cyber-security exercise scenario. The *cyber security strategy game* saves an exercise scenario as an instance of the DSL in the form of a YAML Ben-Kiki et al. (2005) file. The DSL is designed to accommodate 11 key concepts related to cyber-security exercises. It can be used to implement three types of cyber-security exercises, which are presented in Section 7.
3. *Infrastructure orchestration module*
   The *infrastructure orchestration module* is a compiler that takes the DSL and performs syntax validation. If the code has no errors, then it generates the infrastructure, as described in the DSL. The infrastructures generated in the form of **HEAT** (hea, 2019) and **Puppet** (Pup, 2020) stack and deploy them on the open stack cloud environment. The technical details of the *infrastructure orchestration module* are presented in the corresponding section.

A schematic representation of the proposed system and its layers of abstraction are presented in Fig. 1:

6. Cyber-security strategy game

As discussed in Section 2, we identified the inefficiencies in cyber-security exercise development (Yamin and Katt, 2018). We also identified that automation could help in reducing these inefficiencies (Yamin et al., 2018). As a first step toward this automation, we hypothesized that serious gamification would help (Yamin and Katt, 2019b) in removing the identified inefficiencies. To validate our hypothesis, we conducted a survey during NCSC (Norwegian Cyber-Security Challenge) 2019 (sta, 2019). The test subjects consisted of 25 participants who qualified for the initial CTF round at the NCSC, in which around 150 people participated. In the survey, we asked questions about serious games for cyber-security education, evaluated our developed game, and assessed players' skill sets, the details of which are given in subsequent sections.

6.1. *Preliminary analysis*

The game was developed as a proof of concept by three bachelor students during their final year project at NTNU (Norwegian University of Science and Technology) (git, 2020). The game pedagogic objectives are to achieve the following:

1. Increase player awareness of how cyber-attacks and defenses are conducted.
2. Provide an understating for strategizing cyber-attacks and defenses.
3. Provide an understating for decision making at the operational cyber-security level.

This was achieved by incorporating the concepts of penetration testing methodology (Allen et al., 2014) and the cyber

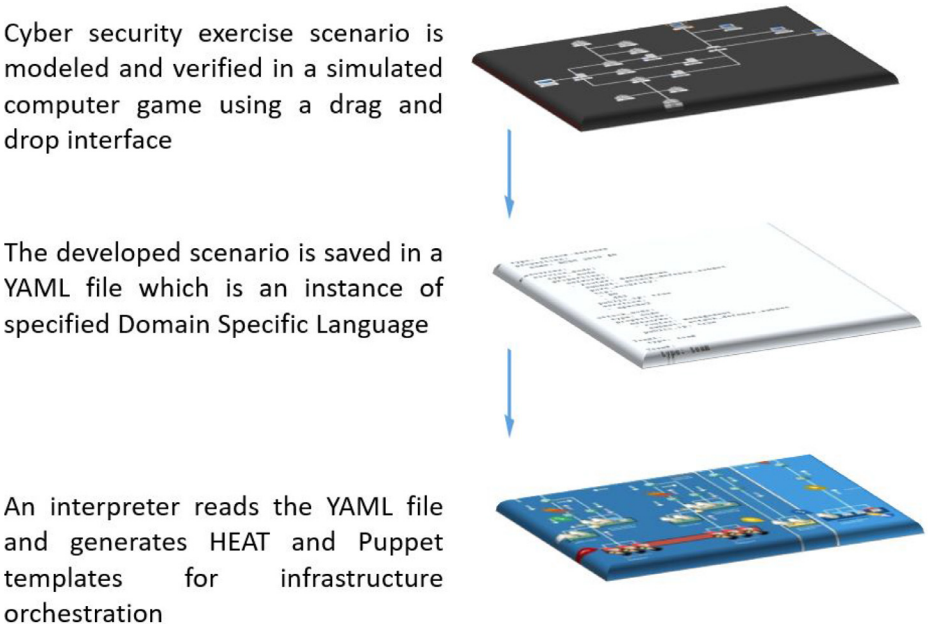


Fig. 1 – Proposed system parts and corresponding layers of abstraction.

kill chain (Hutchins et al., 2011; Yadav and Rao, 2015). These concepts included a total of 16 skills developed after an analysis of a common curriculum being taught at US-DODapproved certification programs (Yamin and Katt, 2019a). The skill set concepts included in the game are as follows:

- Network and system security
- Information security and management
- Cyber-security incidents and response
- Risk analysis and management
- Forensics and cryptography
- Windows and Cisco device security
- Application and web security
- Security concepts and controls

To apply the cyber-security skill set in a realistic environment, a methodological approach of the cyber kill chain was used, incorporating both the attackers' and defenders' actions; the details are specifically given in Section 6.2.1. Currently, the game integrates most of the stated skill set; however, a very specific skill set related to Windows and Cisco device security still requires additional work. We planned to use the game for cyber-security education; therefore, we set the target audience age group between 16 and 25 years old. A sample of 25 top-ranking individuals selected out of 150 participants of NCSC 2019 qualifiers participated in the survey; this target audience group was selected based on the target audience of the European cyber-security challenge (ecs, 2019). We considered the sample group as a reliable indicator for such research in the Norwegian context. The survey questions were straightforward, neutral, and easy to understand. Besides *Yes* and *No* answers, the participants were given the option *Maybe* if they were not sure. One of the questions related to computer games in general, and the other two tackled attack and defense scenarios separately. The wording was carefully chosen based on the background of the participant (highly technical). Finally, the survey was administered, and the participants responded to the questions in a relaxed environment to avoid any biases. Below are the findings of our survey about serious games in cyber-security exercises.

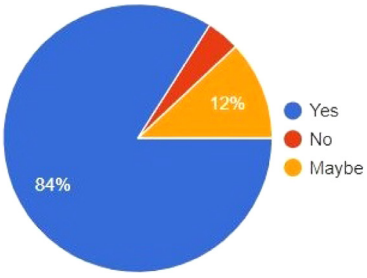


Fig. 2 – Percentage of the participants who thought computer games could help in cyber-security education.

1. Do you think computer games can help in cyber-security education?
   The first question was a general question about the role of computer games in cyber-security exercises. Here, 84% of participants considered that they could play an important role, 12% were not sure about the role of computer games in cyber-security exercises, and only 4% did not consider them useful. The survey findings are presented in Fig. 2.
2. Do you think practicing attack strategies in games is useful before launching a real attack?
   The second question was related to cyber-attack strategies. The purpose was to identify whether it is a good approach to practice a simulated attack strategy before launching a

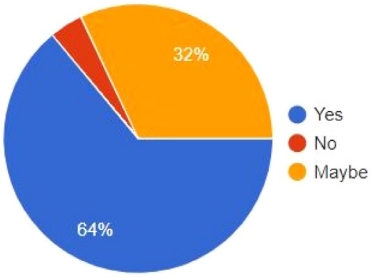


Fig. 3 – Percentage of the participants who thought practicing attack strategies in games is useful before launching a real attack.

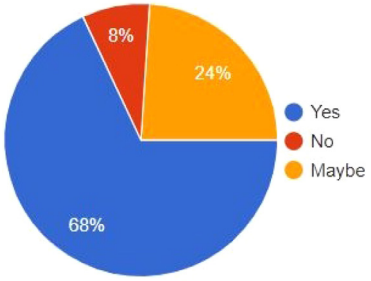


Fig. 4 – Percentage of the participants who thought practicing defense strategies in games is useful before defending against a real attack.

real attack on actual infrastructure. Here, 64% of the survey participants considered this a useful approach, 32% were not sure, and 4% did not find this approach useful. The survey findings are shown in Fig. 3.

3. Do you think practicing defense strategies in games is useful before defending against a real attack?
   The third question was related to cyber defense strategies. The purpose was to identify whether it is a good approach to practice a simulated defense strategy before defending against a real attack on actual infrastructure. Here, 68% of the survey participants considered this a useful approach, 24% were not sure, and 8% did not find this approach useful. The survey findings are shown in Fig. 4.

In *Question 1*, 84% of the survey participants stated that the game could help in cyber-security education, while in *Question 2* and *Question 3*, 64% and 68% stated that it could help in devising attack and defense strategies, respectively. Here, we observed a slight deviation of the survey participants' feedback. We believe that the majority of the survey participants considered that such games are good for cyber-security education in general. However, the survey participants had their own experiences and skill sets regarding operational strategies, which we believe caused the deviation. For instance, most of the survey participants were good at devising attack strategies; therefore, they believed the game would help improve their competence in devising defense strategies and rated it a bit higher.

Regarding the game mechanics, multiple cyber-security strategy games already exist (Hendrix et al., 2016); they are mostly turn-based strategy games. However, because of the dynamic and complex nature of cyber-security exercises, a turn-based strategy is not beneficial for developing cognitive flexibility. Therefore, we decided to develop a real-time strategy game (Glass et al., 2013) to accommodate cyber-security concepts such as penetration testing methodology and cyber kill chain in a multiplayer environment.

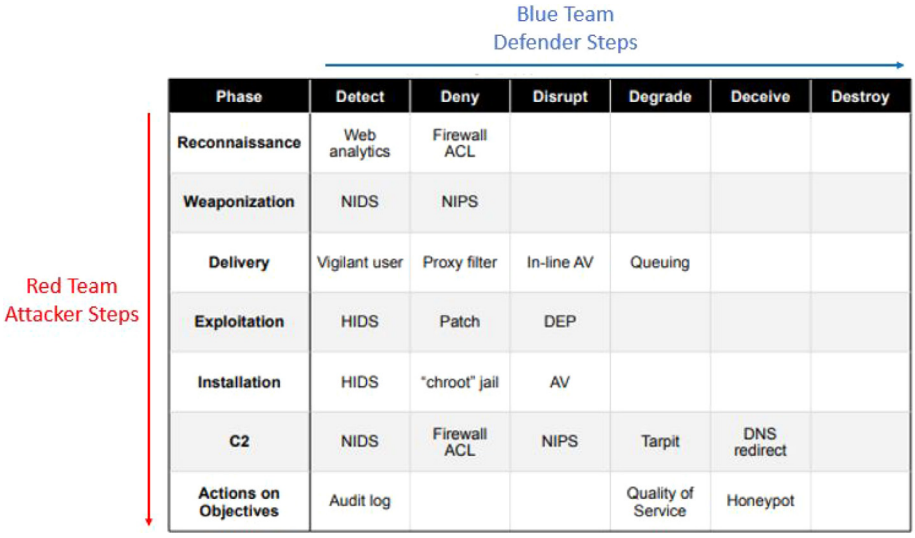
6.2. *Design*

6.2.1. *Integrating cyber-security in a serious game*

- Cyber-security exercise scenario modeling
  Cyber-security exercise scenarios are quite dynamic, and modeling the scenarios based on specific events (Marshall, 2009) is not useful in a multiplayer environment. Adversary player actions can change planned scenario events. Therefore, we opted for a no-win condition in the cyber-security strategy game model. The game players are given the objective to attack or defend a system within a specific time interval. The penetration level assesses players' performance during the attack or the number of attacks stopped during the defense. For the attack and defense steps, we used *Lockheed Martin's* course of action matrix (Hutchins et al., 2011), which is widely accepted in the academic and industrial communities; this matrix is presented in Fig. 5.
- Penetration testing methodology
  We incorporated concepts from penetration testing execution standards in the game design (PTE, 2020) for the attackers. These concepts deal with reconnaissance and information gathering about systems by using active and passive measures. Then, the gathered information is used for the identification and discovery of vulnerabilities. After this, the discovered vulnerabilities are used for the exploitation and post-exploitation of the systems. Additionally, performing an analysis of the exploited vulnerabilities and sharing the findings in a report is also incorporated.
- Cyber kill chain
  For the defenders, we incorporated the concepts from the cyber kill chain (Yadav and Rao, 2015). Cyber kill chain concepts are used to stop the attacker during different phases of attacks, such as during discovery, weaponization, exploitation, and so forth. The defenders have to prioritize the security of the assets at risk and assets protected by other security controls like firewalls, IPS (intrusion prevention systems), and so forth.

6.2.2. *Actors and functionalities*

- White Team
  For White Team members, a scenario modeling interface is proposed in which a White Team member can create a complete network topology. The network topology can contain interconnected components such as APIs, web servers, computers, firewalls, IPS, and so forth. These components have a security level that can be defined as affecting the attack and defense cost within the game. New se-



Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	"chroot" jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

Fig. 5 – Cyber kill chain course of action matrix for attackers and defenders (Hutchins et al., 2011).

curity vulnerabilities (owa, 2020) can be injected in these components according to the need of the scenario. White Team members can also introduce cyber asymmetry by setting system vulnerabilities and exploitation levels from low to high, depending on the exercise objectives. For observing the game, an interface for White Team members is also proposed to observe the Red and Blue Teams' gameplay and progress in real time.

- Red Team
  For Red Team members, an interface is proposed to access different penetration testing methods, such as discovery, probing, and exploitation. They also have a list of known exploits that can be used if they found a vulnerable system. However, all the systems are not vulnerable to known exploits, so they have a panel for researching new exploits related to those systems.
- Blue Team
  For Blue Team members, an interface is proposed to have full visibility of the network topology in the scenario. They have to identify whether the systems are up to date with no vulnerabilities; if they find a vulnerability, they can patch it. The defenders have the option to place security controls like firewalls and IPS within the topology to secure the systems further.
- Green Team
  For Green Team members, an interface is proposed to have full visibility of the network topology in the scenario where there are live-action representations of Red and Blue Teams at the same time. This interface provides the capability to monitor the team's performance and engage spectators in the game.
- Game economy
  Every action in the game has a cost; the cost is determined by the White Team members who planned the scenario. Red and Blue Team members have to make operational cyber-security strategy decisions to achieve their objectives while keeping the cost of their actions in mind. Moreover, time plays an important role during the gameplay because the game is intended to be completed in a specific amount of time, so the game players must make quick decisions.

6.3. *Development*

The game took nearly five months from its initial planning to complete development. The game was developed using Unity 3D (Unity Technologies, 2020), a standard game development engine. The game is called *Red vs Blue, Cyber-security Simulator*. We made the game open source so that anybody can make changes to the in-game functionality per their requirements; the game source code is available at GitLab (git, 2020). The game's most important component is a dynamic cyber-security exercise scenario creator, which provides drag-and-drop functionality of different IT infrastructure objects to create a network topology. The developed topology is saved in a YAML file in the form of a scenario model. We used the developed models to deploy an emulated cyber-security exercise infrastructure. The developed real-time cyber-security strategy game is presented in Fig. 6. The technical details of the game functionality are discussed next.

6.3.1. *Program flow*

The game has two main parts: first is a scenario creator, which was developed to help *White Team* members in designing cyber-security exercise scenarios. A new scenario can be created, or old scenarios can be edited from a YAML file in the scenario creator. The second part involves the gameplay in which *Red* and *White* Team members play the developed scenario. The Red Team members can attack, exploit, probe, and analyze the system's components present within the scenario environment, while Blue Team members can probe, analyze, and defend the system components. There is a third part of the game in which the whole gameplay of Red and Blue Team members can be monitored; this is for the *Green Team* mem-

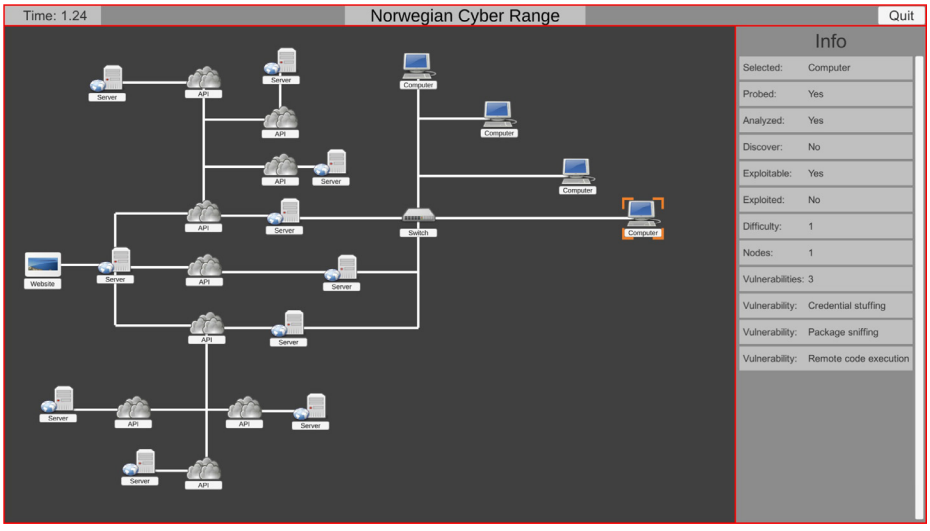


Fig. 6 – Developed real-time cyber-security strategy game.

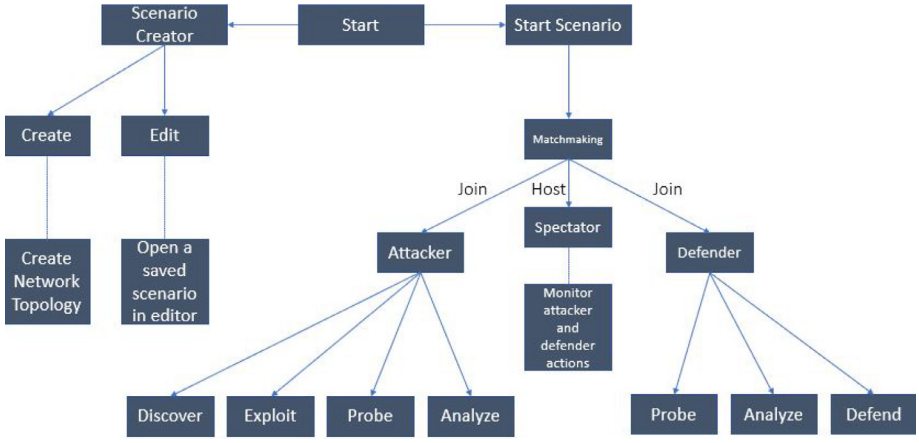


Fig. 7 – Game program flow.

bers. The game program flow with its major components, is presented in Fig. 7.

6.3.2. *Scenario creator*

The scenario creators provide the *White Team* members with three functionalities:

- *System components*
  System components comprise computers, router, switches, APIs, and other infrastructure-related components that can be dragged and dropped on a 2D plane. The components are configurable in such a way that the vulnerabilities or defenses associated with them can be defined.
- *Component connections*
  The component connection allows the *White Team* members to define the inter-connectivity between the different system components. This inter-connectivity helps design wide ranges of cyber-security scenarios because system components can be the same for multiple scenarios. However, the network topology can change the way the attackers and defenders play the scenario.
- *Component menus*
  Component menus allow the *White Team* members to configure the component with vulnerabilities and defenses. Then, they can configure the component level of exploitation by adding high-risk vulnerabilities in it, or they can set the component with no vulnerabilities at all. It all depends on the scenario requirement and complexity. Fig. 8 represents the component menu, as seen by White Team members.

6.3.3. *Attack and defense game play*

The game offers simulated attack and defense gameplay in which attackers can discover, exploit, probe, and analyze different system components, while defenders can probe, analyze, and defend different system components. Attackers and defenders have realistic options available at their disposal to make choices like scanning a network for an attacker or patching the system and placing a firewall in front of a vulnerable

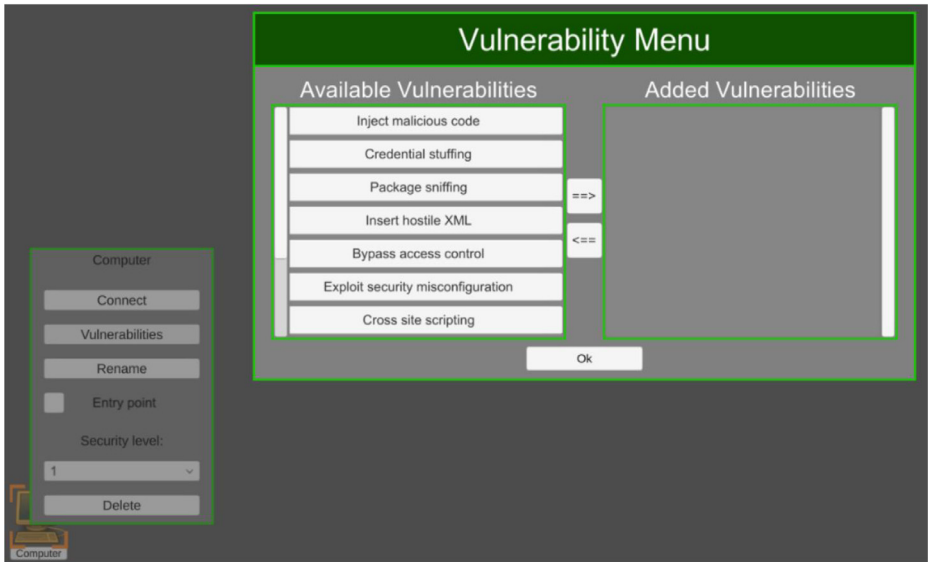


Fig. 8 – Component configuration menu.

component for a defender. These attack and defense choices have a cost that is pre-assigned by the *White Team* members. There is a reward system for successful exploitation and defenses, which opens other options to attackers and defenders, like scanning for 0day vulnerabilities for the attacker and setting up SIEM (security information and event management) solutions for defenders during the gameplay. For a specific example, consider a machine present in a network that has an RCE (remote code execution vulnerability). If the attacker can identify the vulnerability, then the attacker can exploit the vulnerability with a cost of 5. The defender has two options here: (1) patch the vulnerability, which may have a cost of 2 or (2) to place a firewall in front of the vulnerable machine, which may have a cost of 10. There may be multiple machines with the same vulnerabilities to make things complex, and patching all those machines might not be the ideal solution. So the defender has to identify the network paths from where an attacker can exploit such vulnerabilities and place the appropriate defenses.

6.3.4. *Networking and logging*

The game is implemented as a client-server architecture, in which one instance can host a game, while multiple instances of attackers and defenders can join the game. When a game is hosted at an instance, it acts as a server and starts to listen for a TCP connection. When a client wants to join the server, it needs to send a request to a server, and the server assigns the client a game lobby in which the game is hosted. When a client joins the lobby, it obtains access to a messaging server in which different team members can communicate in a textual format. The actions performed by the attackers and defenders and their communications are logged for a post-exercise evaluation about what can be done or what went wrong for a team. The logs are visible to cyber-security exercise observers and are presented in Fig. 9.

Fig. 9 – Event logs collected during a cyber-security exercise.

7. Domain-specific language

In connection with the strategic game, we developed a DSL for specifying and generating cyber-security scenarios as a part of *Norwegian Cyber Range* research activities; this was done with the collaboration of one of our master's students, Dunfjeld (2019). The scenario language, together with its interpreter, are publicly available on Github (Git, 2020). DSLs are programming languages used to solve problems in a very specific domain compared with *general purpose programming languages*, which are used to address problems in a wide area of domains. A DSL provides a layer of abstraction to the user that closely matches the domain-specific problem description and removes the unnecessary overheads of setting up frameworks and writing application-specific technical code. In our proposed DSL, we have identified 11 key concepts required to model a cyber-security exercise scenario, which are presented in the DSL ontology in Fig. 10. A *scenario* has *objectives*, such as capture or defend a flag. To achieve the *objectives*, a *scenario* includes *teams*, *challenges* and *phases*, all of

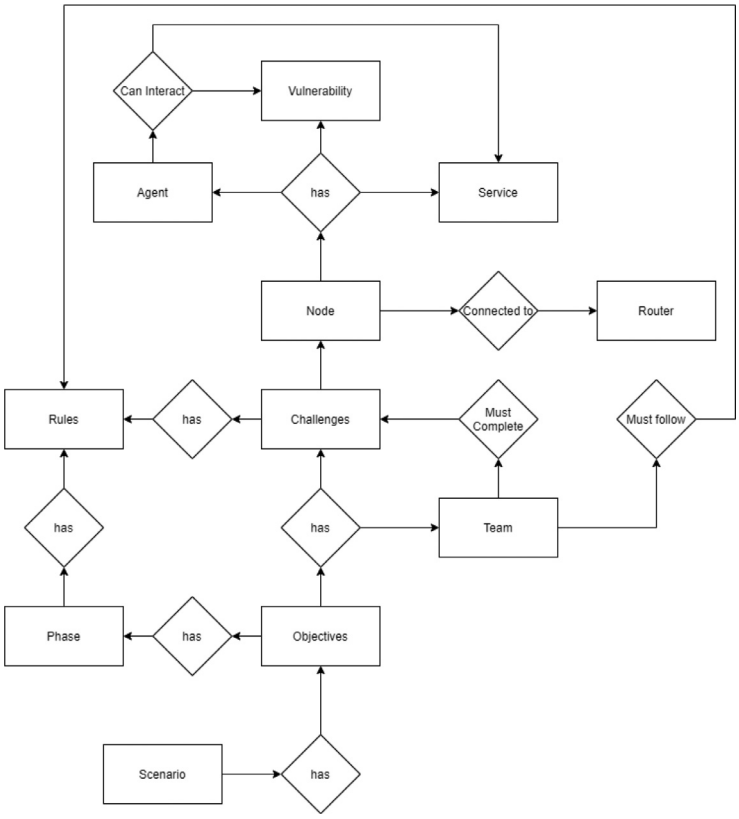


Fig. 10 – An ontology of the main concepts of a cyber-security exercise scenario.

which have *rules*. Teams can be attackers or defenders trying to pass the challenges presented in the scenario. Challenges includes attacking a vulnerable system or patching a vulnerability. These actions are performed during different phases of the exercise, such as the start or middle phase. The challenges are presented on the *node*, which has *vulnerabilities*, *services* and *agents*. The nodes are connected to the *router*, which is connected to the internet for providing access to the exercise platform. For the language, we defined both the abstract syntax and concrete syntax. Additionally, the language compiler/interpreter has been defined and developed (Voelter et al., 2013). The details are presented below.

7.1. *Abstract syntax*

The abstract syntax of the DSL is used to represent the different concepts present within the domain; the identified concepts are as follows:

1. *Scenario properties*
   There are multiple types of cyber-security exercises scenarios; the scenario concept in the DSL is used to define the main properties of a scenario, which are as follows:
   - *Name:* A string value that is used to define the name of the competition or the event within which the scenario will be executed, such as *Defcon CTF*.
   - *Type:* A string value used to define scenario types, such as *jeopardy* and *attack-defense*.
   - *Start date:* A date value that indicates the scenario start date in the format of dd.mm.yyyy.
   - *End date:* A date value that indicates the scenario end date in the format of dd.mm.yyyy.
   - *Start time:* A time value that indicates the start time of the scenario in the 24 h format of hh:mm.
   - *End time:* A time value that indicates the end time of the scenario in the 24 h format of hh:mm.
   - *Docker hosts:* An integer value that indicates the number of docker hosts that are going to run virtual machines in the scenario.
   - *Objectives:* A list of all objectives of the scenario, which will be explained later in more detail.
   - *Agents:* A list of all agents that are active during the scenario, which will be explained later in more detail.
   - *Rules:* A list of all rules that need to be followed in the scenario, which will be explained later in more detail.
   - *Teams:* A list of all teams participating in the scenario, which will be explained later in more detail.

   Scenario properties that have list values like *objectives*, *agents*, *rules*, and *teams* do not contain the definition of the concepts; they just refer to the objects that have the concept definition.
2. *Node*
   The concept of a *node* is used to define a virtual machine that is present in the scenario. It has the following properties and sub-properties:

- *Type:* A string value that is used to define the type of the node.
- *Flavor:* Flavor is used to allocate the amount of RAM, CPU, and storage in the cloud. It is a string value, and if it is not implemented, the default settings are used.
- *OS:* A string value that is used to specify the operating system for a virtual machine.
- *Public IP:* A Boolean value that is used to assign a public IP address to the VM. By default, a VM is not publicly accessible, so this property is used to assign a floating IP address to the VM.
- *Networks:* A list value that is used to represent the connection of nodes in a network topology. It should at least have two of the following sub-proprieties:
  - *Router:* It is the name of the router with which nodes are connected.
  - *Subnet:* A string value is used to indicate the subnets in which nodes are connected.
  - *Port security:* One or many TCP or UDP property values used to represent the open ports on the virtual machine and respective services. By default, only SSH and ICMP ports and services are open for management and diagnostic purposes.
- *Vulnerabilities:* A list value that indicates the vulnerable application and services that need to be installed on a node. The detailed properties and sub-concepts of the vulnerabilities will be explained later.
- *Services:* A list value that indicates the services that needs to be installed on a node.
- *User accounts* A list value that contains the user account details that are present on a node. It has the following sub-properties:
  - *Username:* A string value that indicates the username of the user.
  - *Name:* A string value that indicates the user's full name.
  - *Password:* A string value that indicates the user password in a hash form.
  - *Uid:* A string value that indicates the user's identifier.
  - *Gid:* A string value that indicates the user's primary group ID.
  - *Group:* A string value that can be used to override the user's primary group value.
  - *Groups:* A list value that contains the groups' names in which the user is present.
  - *Home:* A string value that indicates the user's home folder.
  - *ssh key:* A list value that contains the ssh key, which is used to access the user's account.
  - *Shell:* A string value that indicates the user shell's address.

3. *Router*
   A router is used to provide the necessary networking functionality to different nodes. It has the following properties:
   - *Type:* A string value used to define the type of the router.
   - *Network:* The network property contains the information of all the subnets connected to the router. The subnets have their own properties, which are as follows:
     - *CIDR:* A string that indicates the IP range of the subnet.
     - *Gateway IP:* A string that indicates the gateway IP address of the subnet.
     - *Routes:* A list of strings that contains the information of routes between different subnets.
4. *Service:* Services are used to define the applications that are running on the nodes and that are not vulnerable and are used to make the scenario more realistic.
   - *Type:* A string value used to define a service that contains the information about the service that is needed to be connected to a node.
5. *Vulnerability*
   A vulnerability is a component of a node, for example, an application or a service installed in a node, that is intentionally vulnerable or contains an implementation bug or design flaw.
   - *Type:* A string value used to define the vulnerability type such as *DoS*, *RCE*, *XSS*, and so forth.
6. *Challenge*
   A challenge concept is used to represent an exercise or a task that needs to be completed to earn points in a cybersecurity exercise. It has the following properties:
   - *Type:* A string value used to define the type of a challenge.
   - *Points:* An integer value that represents the maximum number of points awarded after completing a challenge.
   - *Port:* An integer value that indicates the port number through which the challenge is accessed.
   - *Prerequisites:* A list of strings indicates some other challenges that need to be completed before accessing the current challenge.
7. *Team*
   The team concept is used to identify the participants' role in a cyber-security exercise. It is also used for point allocation. There can be multiple Red or Blue Teams present in a cyber-security exercise.
   - *Type:* A string value used to define the team type, that is, Red Team or Blue Team.
   - *Members:* A list value that contains the contact information of each member of the team.
8. *Agent*
   Agents are used for performing specific tasks in a cybersecurity exercise in an automatic manner. They can be used to generate traffic or launch autonomous attacks.
   - *Type:* A string value that is used to define the type of the agent like *Traffic generator*, *Attacker*, and so forth.
   - *Sub type:* A property is used to define the sub-category of an agent. The agents can be used for traffic generation, user behavior simulation, and so forth.
9. *Phase*
   A scenario can be broken down into multiple phases, for example, vulnerability discovery, vulnerability exploitation, and so forth. Transitioning from one phase to another results in a possible change of objectives and rules defined in this concept.
   - *Type:* A string value used to define the type of a phase like *start*, *middle*, and *final*.
   - *Objectives:* A list value that contains scenario objectives in textual format with respect to a particular phase.
   - *Rules:* A list value that contains scenario rules in textual format with respect to a particular phase.

```
scenario:
type: jeopardy
properties:
    name: Fancy-name-of-CTF
    start_date: 01.01.1970
    end_date: 07.01.1970
    start_time: 12:00
    end_time: 23:59
    docker_hosts: 3

resources:
    shellshock:
      type: challenge
      properties:
        port: 1338
        points: 50

    heartbleed:
      type: challenge
      properties:
        port: 1337
        points: 30
```

Fig. 11 – Jeopardy-style CTF generation sample code.

- *Agent*:: A list of agents that are phase specific

10. *Objectives*
    A description of scenario objectives that must be completed to successfully complete a scenario. A scenario may have single or multiple objectives depending on the complexity of the scenario.
    - *Text:* A string value that indicates scenario objectives in textual format.
11. *Rules*
    Rules contain information for teams in the scenario, such as *"DOS on the nodes is not allowed"*.
    - *Type:* A string value used to define the type of a rule like *allowed* or *not allowed*.
    - *Text:* A list value that contains scenario rules in textual format.

7.2. Concrete syntax

The concrete syntax is used to create a scenario instance. It can be generated by the real-time cyber-security strategy game and presented in the previous sections, or a user can write it directly with the help of an interactive interface. YAML specification is used for the specification of the concrete syntax of our scenario language. It provides the necessary indentation, helping in creating hierarchical structures and representation of the data in lists, keys, or a combination thereof. All concepts of our language are defined in the form of objects, and the structure for representing any concept in object format is identical. Below is an example of how a concept is represented in an object form in YAML.

```
identifier:
type: concept-type
properties:
    some property: some value
```

The concepts are identified by the object identifier. The object identifier is used as a reference to that object. Each object needs to have a mandatory property *type*, which is used to specify the object type. Each object has its properties assigned by a property identifier. Fig. 11 presents a sample of the concrete syntax used to generate a simple jeopardy-style cyber-security exercise containing three docker hosts and two vulnerabilities. Here, not all elements need to be present in the scenario. For those elements that are not mentioned, default values will be assigned, for example, routers and networks.

7.3. Compiler/Interpreter

Five steps are involved in the compilation of a DSL scenario instance:

1. *Loading*
   The scenario file is loaded into the compiler using the python library oyaml. *oyaml* preserves the dictionary ordering of the file when loading the scenario YAML file.
2. *Syntax validation*
   Syntax validation is performed, and whether the loaded files contain the scenario information according to YAML specification or not is checked. If the file does not follow the YAML specification, the syntax validation process fails, and compilations stop.

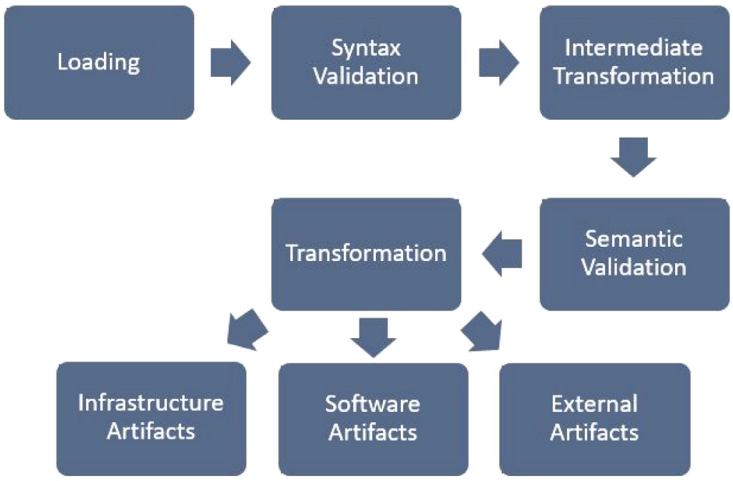


Fig. 12 – Compilation process of DSL.

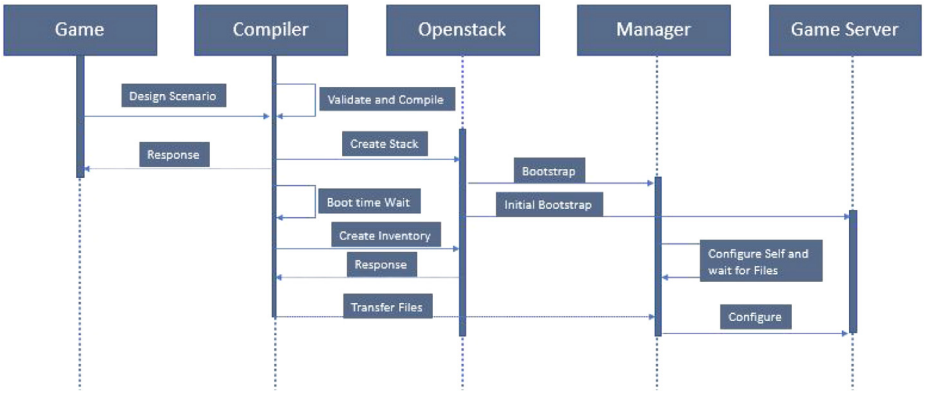


Fig. 13 – Infrastructure orchestration process from a simulated game to an emulated environment.

3. *Intermediate transformation*
   The YAML file data are transformed into a Python dictionarytype data structure. This helps access data and apply compilation logic in Python. One reason to choose YAML as a scenario specification language is its ability to be easily transformed into a Python data structure.
4. *Semantic validation*
   Although syntax validation can ensure that the scenario syntax is correct, semantic errors can still exist. To avoid semantic errors, multiple semantic validation types were applied, as follows:
   - Verification of the input flow structure.
   - Verification of objects that are associated with a particular scenario type.
   - Verification of mandatory properties in objects and their values according to the required data formats.
   - Verification of optional properties in objects and their values according to required data formats.
   - Verification of assigned OS/services/vulnerabilities in the objects present in the compiler database.
   - Verification of assigned IP addresses and that they are in the correct subnet.
5. *Transformation*
   After semantic validation, the data structures are transformed into three artifacts that are usable for low-level platform-specific technology. In our case, we use OpenStack as the cloud platform technology, which will host the final exercise infrastructure. Thus, the end result of the transformation is a set of HEAT templates that can be used to deploy the network and the virtual machines on our OpenStack-based private cloud. The details of the three artifacts are as follows:
   - *Infrastructure artifacts*
     These are artifacts that are used to build a networking component and the virtual machines present in a cyber-security exercise scenario; they are HEAT templates that are used to deploy the exercise infrastructure on OpenStack.
   - *Software artifacts*
     These are artifacts that are used for the installation and configuration of operating systems and services. They are transformed into Ansible (Ope, 2019) templates. Virtual machines have their separate configuration templates that define their settings according to the scenario definition.

- *External artifacts*
  The artifacts that are not related to software and infrastructure artifacts are presented as external artifacts. They contain rules and objectives of the scenario, which are merely textual information related to the cyber-security exercise participants' scenarios.

The compilation process of our DSL is presented in Fig. 12.

8. Infrastructure orchestration module

After a successful compilation of the scenario, the compiler also performs the provisioning process for deploying the cyber-security exercise infrastructure. Because of this, compilation and provisioning become a one-click process. Fig. 13 shows how different components of the proposed systems interact with each other for the full orchestration of the deployment of the cyber-security exercise infrastructure. This is a multi-step process in which, first, a *White Team* member creates the scenario topology within the cyber-security strategy game. According to the DSL specification, the scenario is saved in a YAML file, which is then validated and compiled by the compiler. If the compilation process is successful, the aforementioned artifacts, including the HEAT templates, are generated, and a request is generated to the OpenStack orchestration API to create a stack based on the generated HEAT templates. For the configuration of VMs, the Cloud-init option of OpenStack is used, which initiates basic bootstrapping of the VMs, such as installing Ansible and transferring SSH keys.

When nodes are created that do not have an IP address, DHCP is used to allocate the IP addresses to the nodes. The compiler makes a query to OpenStack and requests the IP addresses of the nodes that were created. A waiting period is added into the compiler to ensure all the nodes are set up and have acquired an IP address. The compiler then updates the list of IP addresses and uses SSH to transfer the required configuration of the nodes, that is, the compiler-generated Ansible templates.

Ansible is a push-based configuration setup utility, meanings that for configuring a VM of the cyber-security exercise scenario, an additional VM is needed to push the scenario configuration from within the scenario network. A manager node is created, which receives the configuration from the compiler. After the Ansible files are received from the compiler, a manager node starts pushing the configuration of all the network-related functions and VM-related services and vulnerabilities based on the scenario requirements specified by the scenario's DSL instance.

8.1. *Nature of emulation*

The toolset produces emulation for a cloud native environment, which is currently OpenStack based. The emulation supports *network*, *transport*, *session*, *presentation*, and *application* layer protocols. However, the *datalink* and *physical* layer protocols were not supported because of the inherent limitations of cloud-native software networking (Ope, 2021). The toolset can create and deploy small and large exercise environments based on the scenario requirements. These exercise environments can be configured to be vulnerable using Ansible. These environments can support a variety of exercises such as jeopardy-style CTF, attack/defense, Red/Blue teaming, and so forth. A generated exercise environment using the developed toolset is presented in Fig. 14.

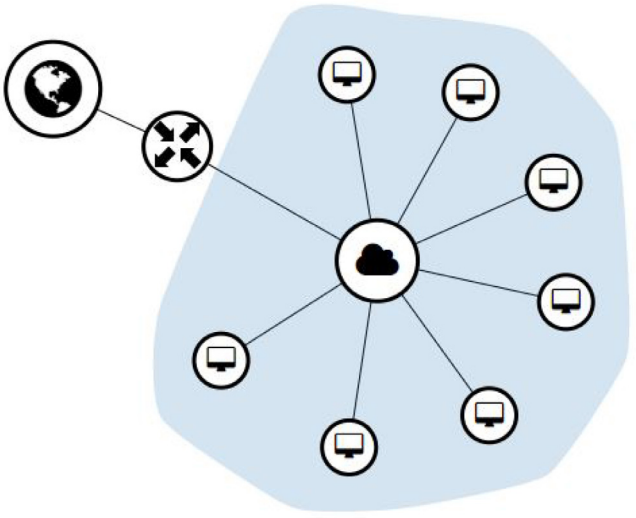


Fig. 14 – Emulated environment produced using the developed toolset.

9. Game and players assessment

To evaluate the whole toolset developed in this research work, we conducted a case study in the context of the NCSC in 2019; a summary of the results of the study are presented in Section 9.5. The NCSC is used for selecting, evaluating, and training the Norwegian team that will participate in the European Cyber-Security Challenge (ECSC). Our research team was part of this process, and the field study was conducted in this context. The goal was to evaluate both the developed game and the scenario language toolset during one of the two qualification rounds of NCSC 2019.

9.1. *Number of participants and demographic data*

The test subjects consisted of 25 participants, 20 male and five females, who qualified for the initial CTF round at NCSC 2019, in which more than 150 people participated from all over Norway. All the survey participants were ethnic Norwegians between the ages of 16 and 25.

9.2. *Task performed by the participants*

The participants were given a brief tutorial about the game and how it works. The participants were seasoned CTF players and had expertise in offense techniques. Therefore, a Red Team game was chosen. Each participant was tasked to play the game individually as an attacker for 20 min in an isolated environment without any external interference. They were

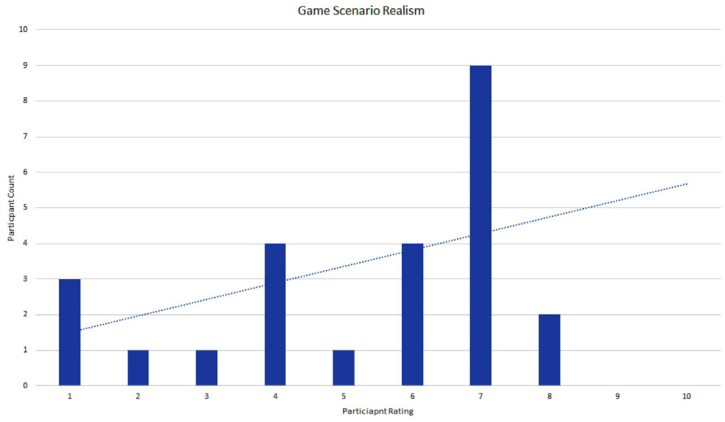


Fig. 15 – Game scenario realism rating.

asked to critically analyze the game because they were expected to provide feedback after the game session. The participants attacked a network with no active opposition but with limited resources. Their decision-making process for attacking the network was evaluated.

9.3. *Data collection*

The data for the study were collected in three ways:

- Post-game session survey
  A survey was conducted after the game session in which the participants were asked six questions, of which four questions were related to game realism and usability and two to the players' assessment of the improvement of their skills, which are reported in the summary of results.
- Game recordings
  The game can act as an observer, so the gameplay can be remotely observed. This functionality was used to record the participant gameplay for participant evaluation in making real-time strategy decisions.
- Post-game session interview
  An expert from CYFOR (Norwegian Cyber Force) (Cyb, 2020) conducted post-game session interviews with the participants. The interviews were used for the psychometric analysis of the participants to assess their cognitive abilities.

9.4. *Data analysis*

Data from the surveys were analyzed using a simple statistical method of *trend line* (Tre, 2020), which is a line that can be drawn on a scatter diagram to represent a trend in the data. In our study, the trend line is presented in histogram charts in the summary of the results. The game recordings and interviews were used for the evaluation of individual player performance during NCSC 2019. In the interviews, the participants were asked to self-reflect on their experience. The details of the individual participants' cognitive performance evaluation processes are out of the scope of this work and will be presented in a study dedicated to this topic. The cyber defense retrospective timeline analysis (Knox et al., 2019) was used for the qualitative evaluation of the individual participants.

9.5. *Summary of the results*

9.5.1. *Game assessment:*

In our field study, we developed a scenario related to an organization's internal network exploitation. The scenario was based on real cyber-security incidents that involved a private organization. The organization had an internet-facing website that was connected to multiple APIs. The website itself was not vulnerable, but one of the deployed APIs was vulnerable to RCE (remote code execution vulnerability). The attacker could exploit the vulnerability and ingress into an internal network with multiple subnets. After that, the attacker had to identify important subnet and resources based on the retrieved information from the network interfaces before penetrating into the important subnet to achieve full network exploitation. On the defender side, the defenders had to patch the vulnerable systems and make strategies to secure the important network subnets with limited resources. We developed questions related to scenario realism in the game and the overall game usability, asking the participants to rate the game from 1 to 10, where 1 was the lowest and 10 the highest value. To ensure correct and sound answers by all participants, they received a short training session on the questionnaires included in the study and the meaning of the scales used before the study. The findings of the survey are as follows:

1. How realistic is the current game in representing cyber-security exercise scenarios?
   Most of the survey participants considered that the representation of cyber-security exercise scenarios was realistic in the game. Here, 15 out of the 25 participants rated the game realism as more than 5, out of which two rated it 8, nine rated it 7, and four rated it 4, as shown in Fig. 15.
2. How realistic is the current game in devising cyber attack strategies?
   The majority of the survey participants considered that the game was realistically devising cyber-attack strategies. Here, 14 out of the 25 participants rated the game realism at more than 5, out of which one rated it 9, two rated it 8, six rated it 7, and five rated it 6, as shown in Fig. 16.
3. How realistic is the current game in devising cyber defense strategies?

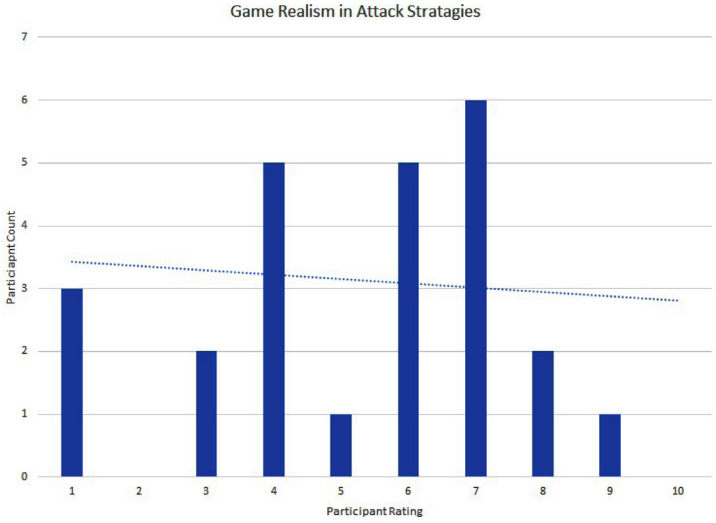


Fig. 16 – Cyber attack strategies realism.

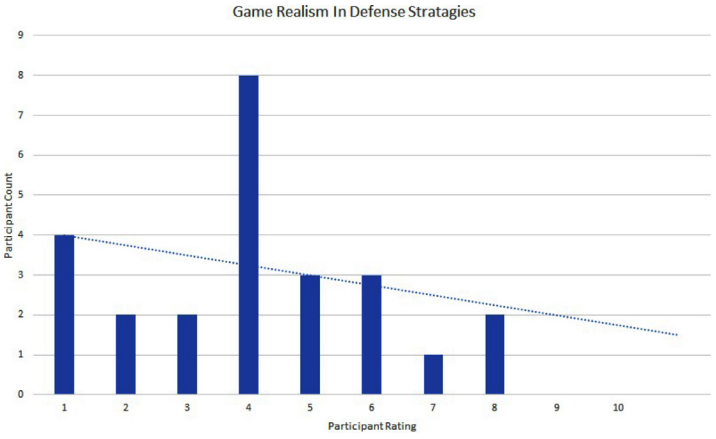


Fig. 17 – Cyber defense strategies realism.

Most of the survey participants considered that the current game was not suitable for realistically devising cyber defense strategies. Here, 19 out of the 25 participants rated the game realism as less than 6, out of which four rated it 1, two rated it 2, two rated it 3, eight rated it 4, and three rated it 5, as shown in Fig. 17.

4. Do you think that the current game can be useful for cyber-security education?
   Here, 44% of the survey participants considered that the game could be useful for cyber-security education. In addition, 36% of the participants were not sure about the game's usability, while 20% of the participants did not consider the game useful in cyber-security education, as shown in Fig. 18.

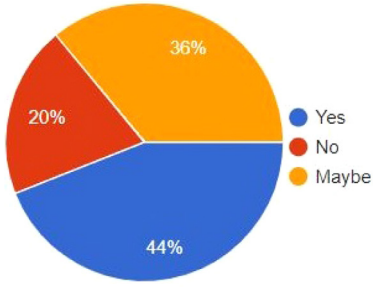


Fig. 18 – Percentage of the participants who thought the developed game is useful for cyber-security education.

9.5.2. *Player assessment:*
The game was successfully deployed during the NCSC 2019 (sta, 2019), in which it was used for players' assessment. We asked the participants to self-assess the way in which the cyber-security exercise was conducted and if their skills had improved. The findings of the survey are as follows:

1. Do you think playing/practicing the cyber-security exercise scenario in a simulated/modeled game is an efficient way to conduct cyber-security exercises?
   Here, 64% of the survey participant considered that playing and practicing cyber-security exercises in a simulated/modeled environment is an efficient way of conduct-

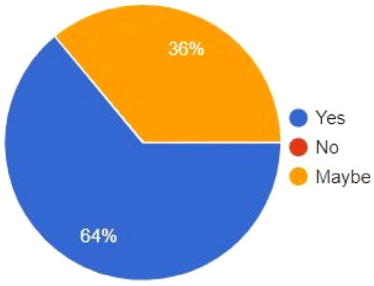


Fig. 19 – Percentage of the participants who thought it is efficient to conduct cyber-security exercises scenarios in a simulated modeled environment.

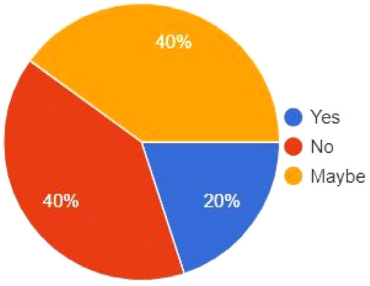


Fig. 20 – Percentage of operational strategy decision-making skill improvement in cyber-security exercises.

ing the cyber-security exercise. In comparison, 36% of the participants were not sure about it, as indicated in Fig. 19.

2. Do you think that your cyber-security exercise operational strategy decision-making skills have improved after playing this game?
   Here, 20% of the survey participants considered that the game helped them in developing their operational cyber-security strategy skills, while 40% stated they do not see any skill improvement and 40% were not sure, as shown in Fig. 20.

During the interview, one observation was that when the expert from CYFOR asked two participants *"Why did you choose the selected strategy?"*. The first replied that she randomly selected the strategy, while the second participant replied that she critically evaluated all possible strategies and then selected the optimum strategy. The decision making helped the second participant secure a place on the national team.

9.6. *Threats to validity*

We tried to quantify the findings of the field study by using statistical methods on a small data set of 25 participants. Although the data set is comparable to similar studies (Abbott et al., 2015), a larger data set would have provided us with more insights. Moreover, we did not take notes of the participants' prior experiences in cyber-security exercises and only tested one scenario during the evaluation process. This is because of the limited time available for the exercise participants. Conducting the experiment with different scenarios and adapting (Pusey et al., 2016), the scenario according to the participants' prior experiences could have yielded more accurate results, which will be taken into account in future experiments. Additionally, the post-survey questionnaire was only tested by the research team, which caused threats to its validity. This is because similar instruments were not identified in the literature. In this exploratory study, we wanted to test the survey, and we plan to validate our instrument in future work.

10. Scenario language evaluation

To evaluate the developed scenario language and its related toolset, including the compilation, deployment, and orchestration, two field studies were conducted in the context of the NCSC 2019. Below, we discuss the conducted studies and their results.

10.1. *Number of participants and demographic data*

Two technical experts were used for the assessment of the developed system. One expert from the Netherlands was actively involved in creating and deploying cyber-security exercise scenarios for NCSC 2019. The other expert from Norway has expertise in infrastructure orchestration on OpenStack using HEAT and Puppet templates.

10.2. *Task performed by the participants*

To evaluate the developed DSL and compiler's performance in terms of the cyber-security exercise scenario infrastructure and its provision, we conducted two case studies. These case studies involved replicating two infrastructures used in NCSC 2019. Two independent experts conducted the replication. For the first time, they did not use our language toolset, and the second time, they used our language toolset. The experts were tasked with generating the two scenario infrastructures: (1) a jeopardy-style CTF and (2) an attack/defense style cyber-security exercise. The exact examples of both the CTF and attack/defense scenarios can be found and accessed in the Github project of the language toolset (Dunfjeld, 2019) in the *examples* folder.

10.3. *Data collection*

After replicating the infrastructure, the experts were interviewed about their experience using the developed artifacts. Interviews were conducted in a semi-structured form to collect their qualitative feedback. The interviews contained questions about a set of four metrics used to evaluate the performance of the DSL toolset qualitatively. The four metrics are as follows:

- *Efficiency:* In this metric, we measured the time required by manual labor compared with the proposed system to deploy and generate the same infrastructure. Time is one observation data point; however, the experts' opinion was also used for making the assessment.
- *Usability:* In this metric, we tried to identify how useful the proposed system was in generating cyber-security exercise infrastructure. Expert observation and feedback were used to assess this metric.

Table 1 – Result of case studies.

Case study	Efficiency	Usability	Completeness	Flexibility
Replicating jeopardy NCSC	5 min with the developed tool compare to 20+ minutes by 2 experts without the developed tool	Bare-bones structure of scenario only	Limited to container-based challenges	Not flexible after deployment
Attack and Defense Exercises	5 min with the developed tool compare to 60+ minutes by 2 experts without the developed tool	Bare-bones structure of scenario only	Limited to container-based challenges	Not flexible after deployment

- *Completeness:* In this metric, we measured the capability of the proposed system of fulfilling the infrastructure requirements for a given cyber-security exercise scenario. The data source for this measurement was the observation made during the replication of the given infrastructures and experts' feedback.
- *Flexibility:* In this metric, we tried to identify the post-deployment modification capability of a cyber-security exercise scenario generated by the proposed system.

The list of questions asked during the interviews is presented in Appendix A.

10.4. *Data analysis*

The expert feedback was analyzed using a comparative analysis (Berg-Schlosser et al., 2009). Their feedback was compared to establish a common understanding of the performance of the system. The common understating was then used to evaluate the overall system using the pre-defined qualitative metrics.

10.5. *Summary of the results*

Both experts agreed that the developed tool was efficient in deploying cyber-security exercise infrastructure when it came to time. For example, it took only five minutes to deploy a replica of the NCSC jeopardy scenario using the developed tools; in contrast, the two experts took more than 20 min for the same task using general purpose infrastructure orchestration technologies like OpenStack HEAT. The attack/defense scenario took the experts more than an hour to deploy, while with the developed tools, they were able to replicate it in five minutes. However, in terms of usability, completeness, and flexibility, there is a room for improvement because our method only provides a bare-bones infrastructure that only supports container-based challenges. These challenges are suitable for application layer security exercises but do not provide much of an attack surface for network layer attacks. The summarized results are presented in Table 1.

10.6. *Threats to validity*

We used only two experts for the assessment of the proposed system. This is because of the lack of such experts in the field. In the future, we will try to get the feedback of as many experts as possible to obtain more feedback of the system. Moreover, the proposed system was only tested in NTNU's highly customized cloud infrastructure. There may be operational and technical difficulties in other deployment environments. We made the proposed DSL toolset open source and hopefully will receive feedback from other researchers about the operational and technical issues and testing to further enhance its performance and functionality.

11. Discussion and conclusion

In the present study, we developed a multi-layer system (toolset) to support the planning and execution of cybersecurity exercises. The developed system bridges the gap between two different perspectives: a strategic simulation-based serious game and a low-level technical cybersecurity exercise infrastructure. The glue that connects both of these perspectives is a DSL and its corresponding ontology. The language was used to (1) define the input needed to configure the simulation game, (2) transform the game specification into an intermediate scenario format, and (3) use the concrete intermediate scenario format to generate low-level infrastructure artifacts. Additionally, we conducted a case study to evaluate realism and efficiency.

We developed a serious game that provides a drag-and-dropbased graphical user interface to configure the exercise scenario based on the scenario language. This helped model and test cybersecurity exercises scenario in a simulated environment before actual deployment in an emulated environment. The game provides a layer of abstraction to model cyber-security exercise scenarios and test different attack and defense strategies. In terms of cyber-security exercise scenario modeling, we developed a DSL that enables modeling *White Team* of the members' role. The developed language allowed for efficiently translating the cyber-security exercise scenario developed in the game's simulated environment to an emulated environment of an actual infrastructure.

We conducted a case study in which we identified that the game achieved its desired objectives for strategizing cyber attack and defense. The results from the case study indicate that the game realistically represents the cyber-security exercises scenario. The toolset developed during the present research produced an emulation for a cloud-native environment, which is OpenStack based. The emulation supports most of the application and network layer protocols, making it useful in conducting cybersecurity exercises in a university setting. We suggest that such a toolset is also useful for cyber-security ed-

ucation, which is key because practicing cybersecurity strategies in a simulated environment can result in skill improvements.

Currently, the scenarios developed by our toolset offer low fidelity and are not suitable for use in a military cyber operations center and for exercises conducted to train cyber mission planners. For such scenarios, complex, multi-sector, and evolving organizational infrastructures are needed, for which the developed game is not yet flexible enough. Moreover, in terms of devising cyber defense strategies, the results are not positive. This could be because of the participants' profiles with experience in attacking techniques. The game's target audience played the game from the attacker's perspective, which, according to our assessment, did not give them full insights into a defender's actions and strategies.

Serious games can be a viable tool to model new and unique scenarios for cyber-security exercises. The modeled scenarios can be realistic and can be used to realistically devise cyber-attack strategies. In terms of cyber defense strategies, the research results are inconclusive and require further research. The developed game has been identified as a useful tool for conducting cyber-security exercises in an efficient manner, which helps in operational cyber-security skill set improvement. The target audience for the game was individuals between the ages of 16 and 25, and the game can be useful for their skill improvement. However, the developed toolset is not suitable for complex military-grade cybersecurity exercises yet.

In the future, we plan to use the data generated from the game from devising attack and defense strategies to develop autonomous attack and defense agents. These agents can emulate Red and Blue Teams' actions in an actual cyber-security exercise. It would be interesting to assign different levels of capabilities to these agents and test different cyber warfare concepts such as cyber asymmetry. Moreover, we plan to conduct a longitudinal study during the *Ethical Hacking* course taught at NTNU, which will help us identify the usability of the game in providing continuous training and self-learning, hence providing new and unique scenarios. Moreover, to improve other factors such as usability, completeness, and flexibility, we are conducting further research.

Declaration of Competing Interest

The authors declare no conflict of interest in publishing the article "Serious Games as a Tool to Model Attack and Defense Scenarios for Cyber-Security Exercises"

CRediT authorship contribution statement

Muhammad Mudassar Yamin: Conceptualization, Investigation, Methodology, Software, Validation, Writing – original draft. **Basel Katt:** Supervision, Writing – review & editing, Project administration. **Mariusz Nowostawski:** Supervision, Writing – review & editing, Project administration.

Acknowledgment

We would like to acknowledge the valuable contributions of the three undergrad students Christian Bråthen Tverberg, Maarten Dijkstra, and Nataniel Gåsøy, and one master's student, Mihkal Dunfjeld, all of whom took part in ongoing research activities at the Norwegian Cyber Range and assisted us in developing the necessary artifacts for this research.

Appendix A. Interview questioner

1. How much time did it take to deploy a specific cyber-security exercise scenario manually?
2. How much time did it take to deploy a specific cyber-security exercise scenario with the developed tool?
3. Is the deployed scenario usable for cyber-security exercise?
4. Does the deployed scenario provide the required functionality?
5. Is the deployed scenario flexible for changes?
6. What do you think can be improved in the developed tool?

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cose.2021.102450.

REFERENCES

Adam shostack: Tabletop infosec games. 2020. https://adam.shostack.org/games.html. (Accessed on 06/20/2020).

Abbott RG, McClain JT, Anderson BR, Nauer KS, Silva AR, Forsythe JC. In: Technical Report. Automated Performance Assessment in Cyber Training Exercises. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); 2015.

Allen L, Heriyanto T, Ali S. Kali Linux–Assuring Security by Penetration Testing. Packt Publishing Ltd; 2014.

Alotaibi F, Furnell S, Stengel I, Papadaki M. A review of using gaming technology for cyber-security awareness. Int. J. Inf. Secur. Res.(IJISR) 2016;6(2):660–6.

Amorim JA, Hendrix M, Andler SF, Gustavsson PM. Gamified training for cyber defence: methods and automated tools for situation and threat assessment. Proceedings of the NATO Modelling and Simulation Group (MSG) Annual Conference 2013 (MSG-111), 2013.

Ben-Kiki O, Evans C, Ingerson B, 2005. Yaml ain't markup language (yaml version 1.1. Technical Report, yaml.org, 23.

Berg-Schlosser D, De Meur G, Rihoux B, Ragin CC. Qualitative comparative analysis (QCA) as an approach, 1; 2009. p. 18.

Beuran R, Tang D, Pham C, Chinen K-i, Tan Y, Shinoda Y. Integrated framework for hands-on cybersecurity training: cytrone. Comput. Secur. 2018;78:43–59.

Casini M, Prattichizzo D, Vicino A. The automatic control telelab: a user-friendly interface for distance learning. IEEE Trans. Educ. 2003;46(2):252–7.

Chadha R, Bowen T, Chiang C-YJ, Gottlieb YM, Poylisher A, Sapello A, Serban C, Sugrim S, Walther G, Marvel LM, et al. Cybervan: a cyber security virtual assured network testbed. In: Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference. IEEE; 2016. p. 1125–30.

Cheung S, Lindqvist U, Fong MW. Modeling multistep cyber attacks for scenario recognition, Vol. 1. IEEE; 2003. p. 284–92.
Cyberforsvaret - forsvaret.no. 2020. https://forsvaret.no/cyberforsvaret. (Accessed on 06/20/2020).
Cyber security strategy Norway 2012, 2012. https://tinyurl.com/rxmw9t9m. (Accessed on 04/27/2021).
Dunfjeld M. Cyber security testbed provisioning using a domain specific language. NTNU; 2019. Master's thesis.
Endicott-Popovsky BE, Popovsky VM. Application of pedagogical fundamentals for the holistic development of cybersecurity professionals. ACM Inroads 2014;5(1):57–68.
Ecsc2019. 2019. www.europeancybersecuritychallenge.eu.
Github - mdunfjeld/ctfgen. 2020. https://github.com/mdunfjeld/ctfgen (Accessed on 03/30/2020).
Ford V, Siraj A, Haynes A, Brown E. Capture the flag unplugged: an offline cyber competition. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM; 2017. p. 225–30.
Glass BD, Maddox WT, Love BC. Real-time strategy game training: emergence of a cognitive flexibility trait. PLoS One 2013;8(8):e70350.
Gurnani R, Pandey K, Rai SK. A scalable model for implementing cyber security exercises. In: Proceedings of the International Conference on Computing for Sustainable Global Development (INDIACom). IEEE; 2014. p. 680–4.
Maarten dijkstra / cyber security simulator. 2020, https://ntnu.box.com/s/1ysjltlu025h0w0383gmgziqqsu0vp85.
Heat openstack - orchestration. 2019. https://wiki.openstack.org/wiki/Heat (Accessed on 12/01/2019).
Hendrix M, Al-Sherbaz A, Victoria B. Game based cyber security training: are serious games suitable for cyber security training? Int. J. Serious Games 2016;3(1):53–61.
Hutchins EM, Cloppert MJ, Amin RM, et al. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead. Issues Inf. Warf. Secur. Res. 2011;1(1):80.
Knox BJ, Lugo RG, Sütterlin S. Cognisance as a human factor in military cyber defence education. IFAC-PapersOnLine 2019;52(19):163–8.
Le Compte A, Elizondo D, Watson T. A renewed approach to serious games for cyber security. In: Proceedings of the 7th International Conference on Cyber Conflict: Architectures in Cyberspace. IEEE; 2015. p. 203–16.
Liu P, Zang W, Yu M. Incentive-based modeling and inference of attacker intent, objectives, and strategies. ACM Trans. Inf. Syst. Secur. (TISSEC) 2005;8(1):78–118.
list-of-measures–national-cyber-security-strategy-for-norway.pdf. 2021. https://www.regjeringen.no/contentassets/c57a0733652f47688294934ffd93fc53/list-of-measures--national-cyber-security-strategy-for-norway.pdf. (Accessed on 04/26/2021).
Maines CL, Llewellyn-Jones D, Tang S, Zhou B. A cyber security ontology for BPMN-security extensions. In: Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE; 2015. p. 1756–63.
Marshall J. The cyber scenario modeling and reporting tool (cybersmart). In: Proceedings of the Cybersecurity Applications & Technology Conference for Homeland Security. IEEE; 2009. p. 305–9.
National cyber security strategy for norway - regjeringen.no. 2020. https://www.regjeringen.no/en/dokumenter/national-cyber-security-strategy-for-norway/id2627177/. (Accessed on 06/19/2020).
Norwegian cyber security challenge 2019 (ncsc19) finale. 2019. https://www.ntnu.no/ncsc/ncsc19-finale.
Openstack docs: Openstack-ansible documentation. 2019. https://docs.openstack.org/openstack-ansible/latest/. (Accessed on 12/01/2019).
Openstack docs: Openstack networking. 2021. https://docs.openstack.org/neutron/train/admin/intro-os-networking.html. (Accessed on 04/27/2021).
Owasp top ten vulnerabilities. 2020https://www.owasp.org/index.php/.
Pham C, Tang D, Chinen K-i, Beuran R. Cyris: A cyber range instantiation system for facilitating security training. In: Proceedings of the Seventh Symposium on Information and Communication Technology. ACM; 2016. p. 251–8.
Puppet - openstack. 2020. https://wiki.openstack.org/wiki/Puppet.
Pusey P, Gondree M, Peterson Z. The outcomes of cybersecurity competitions and implications for underrepresented populations. IEEE Secur. Priv. 2016;14(6):90–5.
Russo E, Costa G, Armando A. Scenario design and validation for next generation cyber ranges. In: Proceedings of the IEEE 17th International Symposium on Network Computing and Applications (NCA). IEEE; 2018. p. 1–4.
Schmidt DC. Model-driven engineering. Comput. IEEE Comput. Soc. 2006;39(2):25.
Schreuders ZC, Butterfield E. Gamification for teaching and learning computer security in higher education. Proceedings of the {USENIX} Workshop on Advances in Security Education ({ASE} 16), 2016.
Schreuders ZC, Shaw T, Shan-A-Khuda M, Ravichandran G, Keighley J, Ordean M. Security scenario generator (secgen): a framework for generating randomly vulnerable rich-scenario VMS for learning computer security and hosting {CTF} events. Proceedings of the {USENIX} Workshop on Advances in Security Education ({ASE} 17), 2017.
Shiva S, Roy S, Dasgupta D. Game theory for cyber security. In: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research. ACM; 2010. p. 34.
The penetration testing execution standard. 2020. http://www.pentest-standard.org.
Trendline analysis in excel. 2020. https://www.uwyo.edu/ceas/resources/current-students/classes/esighelp/windows help files/microsoft office/excel-trendline analysis.pdf.
Unity Technologies. https://unity.com/.
Voelter M, Benz S, Dietrich C, Engelmann B, Helander M, Kats LC, Visser E, Wachsmuth G. DSL engineering: designing, implementing and using domain-specific languages. dslbook.org; 2013.
Yadav T, Rao AM. Technical aspects of cyber kill chain. In: Proceedings of the International Symposium on Security in Computing and Communication. Springer; 2015. p. 438–52.
Yamin MM, Katt B. Inefficiencies in cyber-security exercises life-cycle: a position paper. Proceedings of the AAAI Symposium on Adversary-Aware Learning Techniques and Trends in Cybersecurity (ALEC 2018), 2018.
Yamin MM, Katt B. Cyber security skill set analysis for common curricula development. In: Proceedings of the 14th International Conference on Availability, Reliability and Security; 2019. p. 1–8.
Yamin MM, Katt B. Modeling attack and defense scenarios for cyber security exercises. In: Proceedings of the 5th Interdisciplinary Cyber Research Conference 2019; 2019. p. 7.
Yamin MM, Katt B, Gkioulos V. Cyber ranges and security testbeds: scenarios, functions, tools and architecture. Comput. Secur. 2019:101636.
Yamin MM, Katt B, Torseth E, Gkioulos V, Kowalski SJ. Make it and break it: an IoT smart home testbed case study. In: Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control. ACM; 2018. p. 26.

Muhammad Mudassar Yamin is currently doing his Ph.D. at the Department of Information and Communication Technology at the Norwegian University of Science and Technology. He is the member of the system security research group and the focus of his research is system security, penetration testing, security assessment, intrusion detection. Before joining NTNU, Mudassar was an Information Security consultant and served multiple government and private clients. He holds multiple cyber security certifications like OSCE, OSCP, LPT-MASTER, CEH, CHFI, CPTE, CISSO, CBP.

Basel Katt is currently working as an Associate Professor at the Department of Information and Communication Technology at the Norwegian University of Science and Technology. He is the technical project leader of Norwegian cyber range. Focus of his research areas are: Software security and security testing Software vulnerability analysis Model driven software development and model driven security Access control, usage control and privacy protection Security monitoring, policies, languages, models and enforcement

Mariusz Nowostawski is an Associate Professor at Norwegian University of Science and Technology. Previously, an academic lecturer at University of Otago, New Zealand. His MSc studies were focused on AI and machine learning, and his Ph.D. on autonomous systems and computational modelling of the biological process of life. Mariusz has worked on high-end networking applications on GPUs and multicore systems with Sun Microsystems and Oracle. He is currently involved in forensics research with Europol. Bitcoin anonymity. Cryptocurrencies.