Evaluation of HackLearn COFELET Game User Experience for Cybersecurity Education

Menelaos N. Katsantonis[1], Ioannis Mavridis[1]
[1]*Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece*
{mkatsantonis,mavridis}@uom.edu.gr

Abstract

HackLearn is a scenario-based hacking simulation game for teaching cybersecurity concepts while providing hands-on hacking experiences to the learners. HackLearn design is based on the COFELET framework, which assimilates modern learning theories, well-known cybersecurity standards, and built-in scaffolding and assessment features. Aiming at evaluating the user experience perceived by HackLearn's users, we describe the process of adopting it in a real educational environment based on the didactic framework for simulation games. Additionally, we present the evaluation methodology elaborated, based on the serious games' quality characteristics framework. We discuss the evaluation results which indicate that HackLearn is engaging, motivating, usable and effective in teaching cybersecurity concepts and hacking strategies and techniques. The evaluation results revealed the HackLearn's aspects that can be improved such as the scaffolding feature and the communication mechanism with the game's back-end facility. The presented work validates and finalizes prior work elaborated on the COFELET framework (e.g., COFELET ontology and the COFELET games life-cycle), whereas it provides directions for future work in the development and evaluation of cybersecurity serious games.

Keywords: *Cybersecurity education, Serious games, Evaluation, User experience, COFELET*

1. Introduction

Over the past years, the increasing lack of capable cybersecurity personnel [1] has drawn the attention to the enhancement of the cybersecurity education. In this light, several educational programs have been developed in organizations, universities and schools to attract people to follow a career in cybersecurity and to effectively train cybersecurity professionals. However, cybersecurity education programs face many issues and challenges reviewed and analyzed in [2] and [3]. For this reason, new strategies need to be developed with the potential to confront the challenges of cybersecurity education and to improve its effectiveness such as the adoption of game-based learning. To this end, the Conceptual Framework for eLearning and Training (COFELET) framework has been proposed in [2] as a reference for developing effective cybersecurity learning and training approaches. Based on the COFELET framework, the HackLearn prototype game and a HackLearn's scenario has been designed and preliminary evaluated [1]. For the evaluation of the HackLearn's design, an evaluation scheme based on key characteristics of cybersecurity game-based learning approaches was employed. This preliminary evaluation showed that HackLearn has the potential to deliver effective cybersecurity education services with advanced scaffolding and assessment capabilities. Besides, a preliminary estimation of the cost shows that HackLearn has lower preparation and running costs than live competitions,

as it is considered cheaper to create game scenarios based on reusable elements than organizing and running live competitions (e.g., Capture the Flag - CtF). Though, the employed evaluation in [1] aimed at assessing the game's feasibility in the design phase, thus an evaluation of HackLearn user experience in real settings is necessary to measure its educational effectiveness and to come to safe deductions on the game's impact.

In this paper, the evaluation process in real settings and the produced results of the HackLearn COFELET game are presented. The main research question regards the evaluation of HackLearn's user experience (UX) when utilized in a real educational environment as a pedagogical tool. The presented evaluation also focuses on the assessment of HackLearn's pedagogical effectiveness in a didactical approach. To exemplify the experiences of learners, the paper provides a brief overview and implementation considerations of HackLearn along with the game's environment and the prototype game scenario used. Although the implementation of the prototype scenario is based on the design presented in [1], this paper includes aspects of the prototype scenario explaining the cyber security hands-on experiences learners had (e.g., learner's actions, simulated tools learners used) while playing the game. Moreover, the necessary elements of HackLearn are described to illustrate the manner that HackLearn materializes the COFELET framework. Finally, this paper presents the methodology applied for the evaluation of HackLearn including the presentation of the didactic framework [4] and the manner it was adopted in the presented study along with the serious games' quality characteristics framework [5].

The remainder of this paper is organized as follows: section 2 briefly provides the theoretical background of this work; section 3 illustrates the HackLearn's overview, implementation, environment and scenario employed in the evaluation process; section 4 presents the evaluation methodology; section 5 presents the flow process of the conducted evaluation experiment; section 6 presents the HackLearn's evaluation results discussed in section 7; and section 8 concludes the paper.

2. Background

2.1. The COFELET framework

The COFELET framework (Fig 1) specifies the main elements that have to be considered for the development of educationally effective cybersecurity serious games, known as COFELET games.

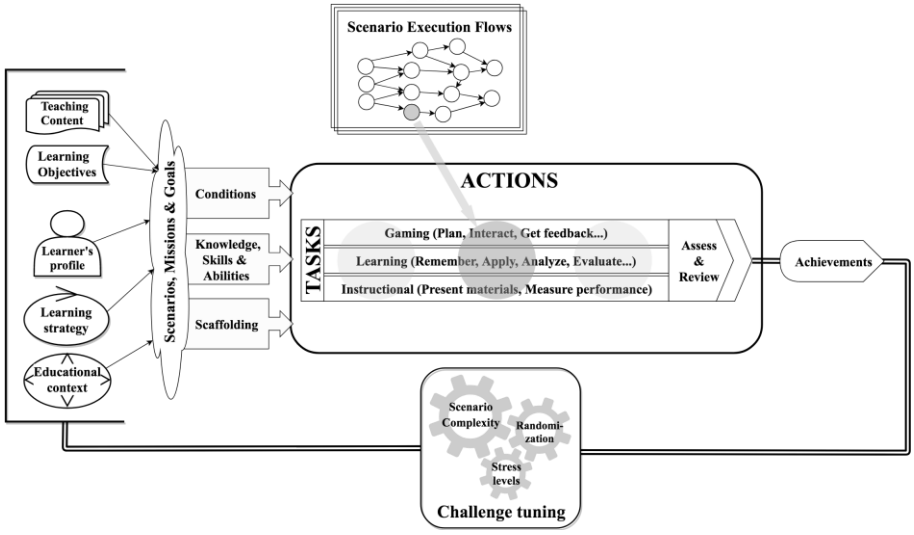


Figure 1. *The COFELET framework [2]*

COFELET foresees serious games that combine pedagogical elements with the elements that represent users' actions towards the fulfilment of the games' goals. According to the COFELET framework, user actions are represented by the *Task* elements (represented in Fig. 1 by unnamed circles) organized in the *Scenario Execution Flows* (SEFs) elements. A SEF defines the possible sequence in which tasks can be executed, the conditions that have to occur to make the tasks performable (i.e., the *Condition* elements) and the goals the task sequence aim to achieve (i.e., the *Goal* elements).

The COFELET games' pedagogical elements include the *learning objectives* (LOs), the *knowledge*, *skills* and *abilities* (KSAs), the *hints* and the *teaching contents*. The *LOs* describe the KSAs aimed to be fostered in learners, the *hints* are the advices provided to the learners to help them achieve the game goals and the *teaching contents* are illustrations of the KSAs. During a session, a COFELET game assesses, reviews and scaffolds the learners' efforts. At the end of a game session, the learner's profile is updated. In the subsequent game session, a new scenario can be selected for the learner with respect to the learner's profile and history, the LOs that she has to achieve, the educational environment of the game session (i.e., *educational context*) and the learning strategy.

The COFELET ontology (analytically presented in [6] and [1]) describes the *tasks*, the *conditions* and the *goals* (*primary elements*) denoting that a learner or an entity acts on an entity or an entity has a property. The *primary elements* are interpreted as quintuple statements of the form `<subject entity, property, object entity or property value, source, destination>` or triple statements of the form `<subject entity, property, object entity or property value>`.

The COFELET ontology analytically describes the *scenario* element which involves the appropriate information for a game session including the scenario's steps and the scenario's cyberspace. The COFELET framework envisages the development of scenarios of varying complexity according to the learner's profile, the LOs to be achieved, the learning strategy, and the educational context of the game session. The *steps* of the scenario are related with the set of learners' actions and they also describe information such as the step's goal, the LOs, the hints that will be provided to the learner. Moreover, the LO element is associated with the teaching content element, with the role(s) learner assumes in the game and with a grade scheme describing the manner the LO is assessed. On the other hand, the *cyberspace* of the scenario is a set of conditions and entities describing the game's environment. The entities of the cyberspace represent distinct concepts that lie in the context of each one game (e.g., hosts, tools, commands). For example, the tool entities in the COFELET ontology represent in-game tools that simulate the functionality of real tools used in cybersecurity practice.

The COFELET framework conforms with the ATMSG model [7], an extension of the Learning Mechanic - Game Mechanic (LM-GM) model [8], to facilitate the adoption of the activity theory and the fusion of the learning aspect in serious games. The conformity with the ATMSG model facilitates the systematic analysis, and organization of the games' components and the identification of the game's actions and activities (i.e., a series of actions). The identified components, actions and activities are classified under the gaming, the learning and the instructional perspectives. The COFELET framework also assumes the layer learning approach [2] [9] to apply cognitive principles and to enhance learning process. It also uses the continuous learning paradigm [10] to engage learners in a sustained cycle of learning, updating and reinforcing knowledge.

Nevertheless, COFELET envisages the adoption of well-known models and strategies generally used in threat analysis and modelling approaches that verify the validity, the applicability and the sustainability of the COFELET approaches. Specifically, the *SEFs* are proposed to be defined in analogy to attack patterns (APs), e.g., the APs defined in the Common Attack Pattern Enumeration and Classification (CAPEC) of MITRE [11]. *Scenarios* involving multistep missions and application of complex strategies are proposed to be defined by utilizing standard cyber security models as a guide such as the Lockheed Martin's Cyber Kill Chain (CKC) [12], a popular model describing 7 stages attackers follow

to unleash sophisticated cyber-attacks called advanced persistent threat (APT) attacks. Additionally, COFELET scenarios can be built in analogy to other models of cyber security domain such as the Diamond model [13] which describes the key components of an intrusion event or the MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) [14], a model of attackers' tactics and techniques.

2.2. The Didactic Framework

The didactic framework proposes four stages for the process flow of simulation games in the business field [4], whereas in [15] three stages of the framework have been associated with corresponding phases of assessment (Fig. 2).

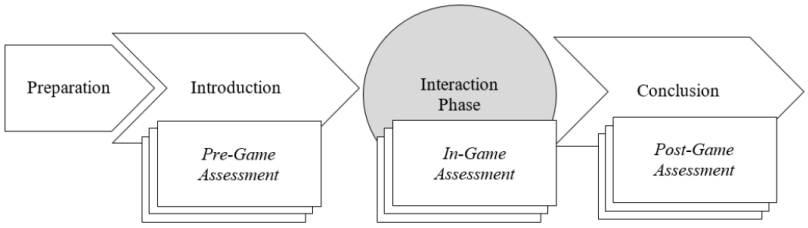


Figure 2. *The Didactic Framework associated with three assessment phases [15]*

In the *Preparation* stage, the appropriate organizational conditions are managed, and the participants are informed about the aims and objectives of the course. In the *Introduction* stage, the participants are familiarized with their roles and the problems they will have to solve in the game. In the *Interaction phase* stage (herein will be stated as *Interactions* stage), the participants interact with the simulation game (i.e., perform game sessions) and face the problems they have to solve. The *Interactions phase* stage consists of five sub-stages in which participants analyze the problem, develop a business strategy, implement a business strategy, run the simulation and present the results. In the *Conclusion* stage, the participants reflect on their decisions and applied strategies and their work is summarized.

On the other hand, in the *pre-game* assessment usually knowledge and capabilities of the learners are measured. Moreover, depending on the context of the educational approach, several data can also be collected such as demographic information (e.g., gender, age), participants' learning styles and attitudes [16]. The *in-game* assessment involves the collection of qualitative data which denotes the participant's performance (e.g., sequences of actions, percentage of goals accomplished, goal completion times). The data of the in-game assessment phase is collected though the games' logging mechanisms or through questionnaires and interviews. Finally, in the *post-game assessment* of the *Conclusion* stage the knowledge and capabilities of the participants are measured through questionnaires, discussions, interviews or performance evaluation by observers [15], [16].

2.3. The Quality Characteristics Evaluation Framework for Serious Games

The researchers of [5] performed a literature review on the quality characteristics used in the evaluation of serious games and they discussed their dependencies and associations. As a result, they also proposed the quality characteristics framework for evaluating the use of serious games (QC framework). The framework divides the quality characteristics found in the literature into primary and secondary characteristics. The primary characteristics include the characteristics of *learning outcomes*, *use experience*, *user satisfaction*, *engagement*, *motivation*, *understandability* and *usability*. They affect mainly the quality of serious games, whereas their absence downgrades the effectiveness of serious games. On the contrary, the secondary characteristics provide minor impacts to quality of serious games and they are not crucial for their success in delivering educational content. The

secondary characteristics include the *game design*, the *effectiveness*, the *user interface*, the *acceptance* and the *usefulness*.

The primary quality characteristics are associated with each other and with the secondary ones. For example, according to the literature review of the [5], the user experience is associated with the engagement, whereas the engagement is based on the motivation characteristic [17] and it is associated with the acceptance characteristic, as the learners will not engage with a game they do not accept.

3. The HackLearn COFELET Game

3.1. Overview

HackLearn is a research prototype based on the COFELET framework and the COFELET ontology and its architecture is illustrated in the COFELET games life-cycle, as described in detail in [1]. HackLearn is an innovative cybersecurity serious game of the hacking simulation game genre (i.e., hacking simulator), as it is the first cybersecurity serious game based on modern learning theories and well-known cybersecurity standards; it provides hands-on experiences on performing cyber-attacks; it incorporates advanced assessment and scaffolding features; and it is a scenario-based game consisting of various reusable elements. The HackLearn COFELET game aims at providing hands-on experiences to computer scientists in utilizing cybersecurity tools, applying attack patterns and unleashing cyber-attacks. Moreover, HackLearn draws several elements from live competitions (e.g., capture the flag (CtF)), as learners use in-game tools to unleash cyber-attacks in the game's cyberspace, they collect points and flags and they exercise their capabilities.

3.2. Implementation

HackLearn has been implemented in the Unity 3D game development engine with C# as the programming language. The game's implementation also included the creation of 300 key elements (e.g., tasks, conditions, goals, hints, LOs) implemented in XML. HackLearn interacts with a back-end storage facility (i.e., MySQL database) in which it stores learner's details, the game's learning analytics, the learners' answers to the in-game questions. The game addresses SQL queries to the back end by utilizing php scripts that communicate with the MySQL database to retrieve and store the game's data.

HackLearn has been developed by a game developer who has worked on the game's implementation for more than a year. HackLearn's implementation process also included the development of a prototype scenario, a MySQL database and a set of php scripts for the communication of the game with the database. The design of HackLearn's interface has been elaborated by a game designer who have worked the interface for three months. The key elements of HackLearn's attack patterns have been designed by a cybersecurity specialist and the game's scenario has been created with the collaboration of experienced educators. Although HackLearn was initially implemented as a PC standalone application, due to the COVID-19 virus pandemic it was exported from Unity as a WebGL web application to run in any web browser, anytime and anywhere.

3.3. Environment

To play HackLearn, the learner has to create an account (Fig.3 (b)) by registering her details and choose the role that she will have in the game (Fig.3 (b)). Once the learner has an account, she visits the HackLearn's login screen (Fig.3 (a)) and enters the username and password.

(a) (b)

Figure 3. *HackLearn's log in (a) and register screens (b)*

Then, the game's front-end communicates with the game's back-end to check the learner's credentials and the learner's profile. If the learner has previous experience with HackLearn, the main scene is loaded with the mission panel enabled (Fig 4) to read the mission. The main scene consists of the *terminal* in which the learner executes Linux-like commands, the *right panel* in which the game's windows appear, a *toolbar* and a *progress bar*. The *toolbar* contains the icons *profile*, *teaching contents*, *inquiry*, *leaderboard*, *mission*, *hint popups*, *messenger* that allow learners to pop up the game's windows in the *right panel*. The *toolbar* also displays a *time counter* and the *learner's name* and it includes the pause button from which the learner enables the pause menu and quits the game. The progress bar displays the learner's *progress* in the mission and her *score*.

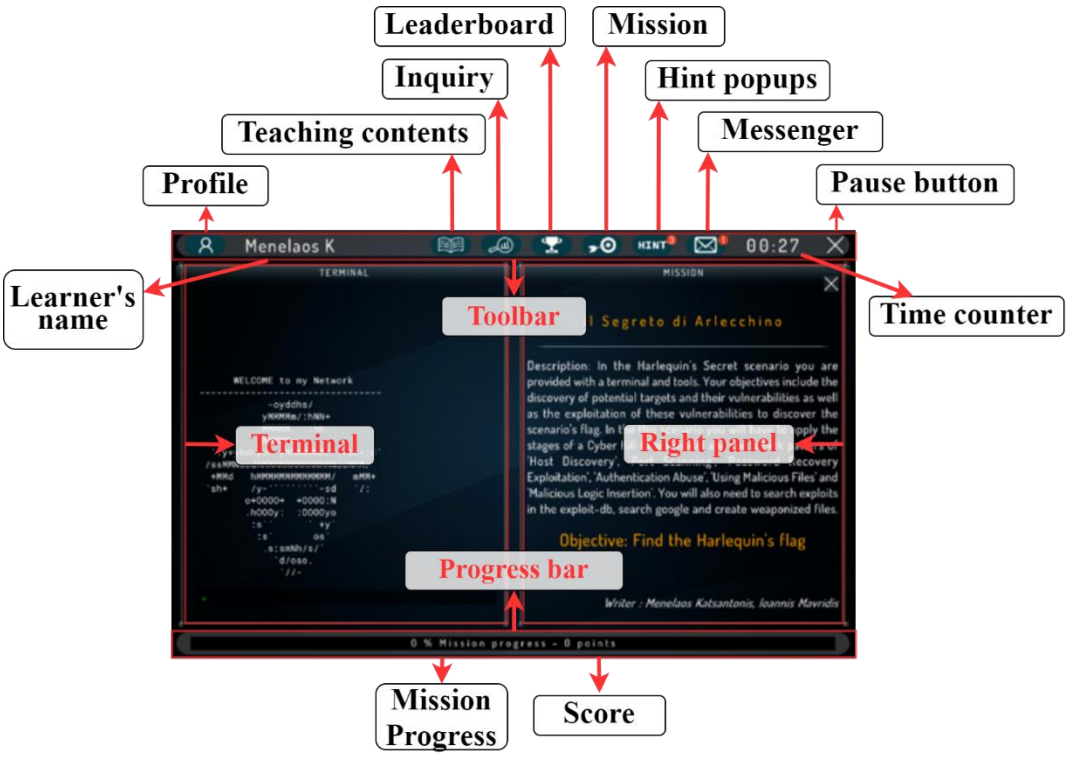
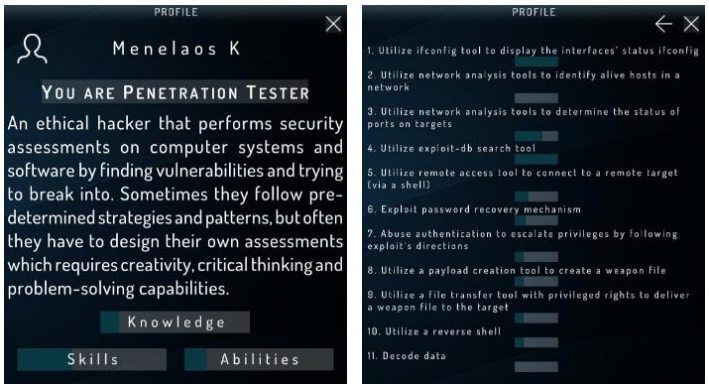


Figure 4. *HackLearn's main scene*

The learner can open her profile window to review the competencies she has to acquire progress, the progress she made and the progress she has to make (Fig 5 (b)).



(a) (b)

Figure 5. *Profile window*

If it is the first time that learner logs in the game, an interactive tutorial is presented to help the learner familiarize with the game's interface (Fig 6).

Figure 6. *HackLearn's tutorial*

HackLearn's missions can be associated with in-game questions that pop up during the gameplay in the inquiry window (Fig 7). An in-game question can be *compulsory* or *optional*. A *compulsory* in-game question requires the user to answer it in order to proceed, whereas an *optional* in-game question does not oblige the user to answer.

Figure 7. *Pop-up windows with an in-game question example*

3.4. Scenario

The COFELET framework envisages scenario-based learning and training approaches tailored to the LOs to be achieved, the learners' characteristics, and the properties of the educational context. In this study, the *il Segreto di Arlecchino* prototype scenario (i.e., the secret of Harlequin in Italian) was developed and used along with HackLearn in the presented study. The aim of the *il Segreto di Arlecchino* scenario is to make the learner comprehend and apply the most of the seven (7) stages of the CKC model to unleash an advanced persistent threat (APT) attack. The target audience of the *il Segreto di Arlecchino* scenario is computer scientists with prior knowledge in networks, operating systems and cybersecurity tools, whereas it was planned to be delivered in a formal educational context. The scenario's goal is to attack the *Harlequin* target host and find and capture the file *flag.txt* stored in this host.

3.4.1. Description

The *il Segreto di Arlecchino* scenario is a composite scenario consisting of nine (9) steps in which learners have to apply the stages of the CKC model [12] and perform 8 attack patterns. The *il Segreto di Arlecchino* scenario contains several game entities (i.e., the network, two hosts, the tools, the commands) and its implementation is based on the design presented in [1] together with its associated COFELET elements: LOs, roles, grade scheme, hints and teaching content.

The scenario draws many elements from the cultural sector (e.g., theatre, music, cinema) that enhance the fun factor and motivation of the learner. Specifically, it draws many cultural elements from the comedic theatre commedia dell'arte[1] (i.e., the Italian comedy) as it adopts the character names of commedia dell'arte to label the hosts, the network and the directories. In fact, the realization of the commedia dell'arte metaphor in the game can help the learner to better comprehend the functions that take place in the target host. For example, the *Harlequin* host (i.e., the scenario's target host) has a directory called 'kitchen' used by the *zanni* user group. Zanni in commedia dell'arte are the servants that carry out the characters' orders. The kitchen directory has low privilege rights as the zanni come and go and they are expected to have low security awareness. Thus, the learner has to figure that the kitchen directory is a good place to put a malware because the zanni users have a good chance to consume a weaponized file. Moreover, in the scenario the learners are required to search details about Andrea Calmo, the author of the commedia dell'arte and when they inspect the target, they will find clues related to the Joker movie character (i.e., an associate of Harlequin) and the Queen band, a band that used elements from the Italian literature in their lyrics.

3.4.2. Cyberspace

Fig. 8 illustrates the main entities of the scenario's cyberspace with which learners interact. The learner has to search the VictoryBall network, discover the *Harlequin* host, and scan the services of Harlequin to find the vulnerabilities that will allow her to gain access and capture the flag. The flag is stored in the '/home' directory of the system administrator's account (root), and thus the learner has to get the administrator's rights to have access to the target directory and capture the flag.

[1] *Commedia dell'arte*: https://en.wikipedia.org/wiki/Commedia_dell%27arte

The learner host is the primary entity with which learners interact and it simulates the functions of a host running the Linux operating system containing the appropriate tools (e.g., nmap, metasploit, msfvenom), which simulate the functionality of real Linux tools used in cybersecurity.

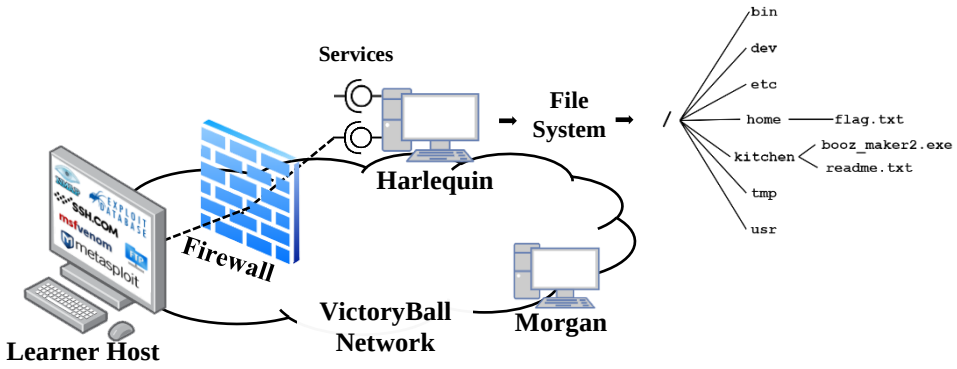


Figure 8. *Cyberspace of the il Segreto di Arlecchino scenario*

The *learner host* is associated with the appropriate *condition* elements, which according to the COFELET framework indicate whether the learners' tasks are performable. For example, the cyberspace contains the condition `<Learner - has - privileged rights - to learner host>` which provides administrator rights to learners to have access to several tools (e.g., the *condition* `<Learner - hasAccess - nmap port scanner tool>`) and functions. The *firewall* entity involves several *condition* elements controlling the flow of the packets such as the condition `<Firewall - drops - All ICMP Packets>` which indicates that firewall drops all the ICMP packets and the condition `<Firewall - accepts - TCP Packet - from learner host- to all target hosts>` which indicates that the firewall accepts the TCP SYN packets destined from learner host to the target hosts. The target hosts contain several entities (e.g., users, services, files) and conditions. For example, the cyberspace specifies that the learner needs administrator rights on the *Harlequin* target host to access the file *flag.txt*, whereas she does not need privileged rights to access the contents of the folder *kitchen*.

3.4.3. Steps

In *il Segreto di Arlecchino* scenario, the learner has to achieve the gaming goal of acquiring the file *flag.txt* by achieving the scenario's goals. According to the COFELET framework, the learner achieves the steps' goals by performing tasks with respect to the occurring conditions. As the primary LO of the *il Segreto di Arlecchino* scenario is to teach the CKC model [1], the rational of the scenario is based on this model. Thus, the learner has to plan a strategy which follows the stages of the CKC model and unleash an APT attack.

Initially, the learner performs SEFs of discovering the target hosts and the vulnerable services on the target hosts (i.e., the Reconnaissance stage of the CKC model). Then the learner creates a weaponized file (i.e., second stage - Weaponization stage), which she delivers to the target (third stage - Delivery stage). The file is consumed by the target (fourth and fifth stages - Exploitation and Installation), a backdoor is created to the target and a connection is delivered to the learner's host (sixth stage - Command and control).

Most of the scenario's steps are associated with corresponding in-game questions aiming to make the learner reflect on the activities she performed in a step and express the knowledge and competencies she exercised in a different form of representation (e.g., textual form). For example, in the question depicted in Fig 7, the learner reflects on an activity that did not bring the result she expected when she used the ICMP ping technique.

Table 1 provides information on the learner's experiences in the *il Segreto di Arlecchino* scenario by presenting a brief description of the tasks the learner has to perform along with the related entities (e.g., tools, hosts, files), the occurring conditions, and the

associated in-game questions. Table 1 shows that the complexity of the scenario's steps evolves as the learner proceeds with her mission. Specifically, in the first steps (e.g., S1 to S3) the learner performs obvious attack patterns (e.g., host discovery, port scanning) but as she proceeds to the scenario's succeeding steps, she has to be more creative and think outside of the box to fulfil the steps' goals (e.g., seek who Andrea Calmo was, guess Calmo's username, pick a target folder to deliver the weapon).

Table 1. *The learner's tasks in the il Segreto di Arlecchino scenario*

Step	***Description***	
S1	Learner uses the ifconfig tool to find out the details of the learner host network interfaces and finds the address of the VictoryBall network.	
	Task	Enters the command `Ifconfig` in the Terminal.
	Question	What does the command 'ifconfig' display?
S2	Learner uses the nmap network analysis tool with a host discovery option (e.g., the TCP SYN ping option or 'PS') to find alive hosts in the network (i.e., the host discovery AP). In case that the learner utilizes the ICMP ping type option, she is informed that the network's firewall drops ICMP packets and thus she has to use a different option.	
	Task	Enters the command `nmap -PS 192.168.*.0/24` (or `nmap -PA 192.168.*.0/24`) in the Terminal.
	Question	Did you try to ICMP ping your target (e.g., with ICMP Type 8 Echo Request datagrams) with an appropriate tool and option (e.g., 'ping -b |target|' or 'nmap -PE |target|')? Did it work or do you think it would have worked? Explain why.
S3	Learner uses the nmap network analysis tool with the TCP SYN stealth scan option (i.e., sS) to apply the port scanning AP and scan the ports of the hosts discovered in S2. In such way, the learner finds information on the services running on these hosts.	
	Task	Enters the command `nmap -sS 192.168.*.27` in the Terminal.
	Question	Is a filtered target port considered opened or closed?
S4	Learner uses the searchsploit tool to identify potential vulnerable services and searches the exploit-db for exploits.	
	Task	Enters the command `searchsploit ftp` in the Terminal.
	Question	Comment on the statement: 'Vulnerabilities and Exploits are more or less the same thing'?
S5	The learner uses the ssh remote connection tool to connect to a service of the *Harlequin* target host. The learner performs a task which involves the activation of the password recovery mechanism and the discovery of the Andrea Calmo's credentials for the target service. The step requires the learner to perform several actions such as the guessing of the Calmo's username, searching the internet to find out who Andrea Calmo is and find out the place of birth of Andrea Calmo (i.e., Venice).	
	Tasks	Enters the following commands in the Terminal and makes 2 random password guesses to activate the password recovery mechanism: 1) `ssh -p 1571 192.168.22.27` 2) `username Calmo` 3) `Venice`
	Question	Explain how you would have implemented a password recovery mechanism.
S6	The learner connects to the *Harlequin* target host, traverses in the file system and inspects the target's files. The step requires the learner to realize that the file booz_maker2.exe is suitable template file for the creation of the weaponized file. The learner achieves the step's goal by performing the task which involves the downloading of the template file to the learner host.	

	Tasks	Enters the command `ftp -p 21 192.168.22.27` in the Terminal followed by commands of traversing in the Harlequin's file system (e.g., `cd` to change directory, `ls` to list contents).
S7	The learner uses the msfvenom tool for creating the weaponized files with the template file acquired in step S6 to create the booz-maker3.exe payload file.	
	Task	Enters the command `msfvenom -p reverse_tcp -x booz_maker2.exe` in the Terminal.
	Question	What is the objective of the weaponization phase of the Cyber Kill Chain model?
S8	Learner utilizes the ftp file transfer tool with privileged rights to connect to the target service and deliver the weapon file (i.e., booz-maker3.exe) to the kitchen directory.	
	Tasks	Enters the following commands in the Terminal: 1) `ftp -p 21 192.168.22.27` to connect to the ftp service 2) `put booz_maker3.exe` to transfer the file to the kitchen directory
	Question	What is the objective of the deliver phase of the Cyber Kill Chain model?
S9	The learner starts the Metasploit Framework penetration testing platform and uses its console to utilize the backdoor to connect to the host with administrator rights. Then, she inspects the files of Harlequin host and in the '/home' directory discovers the 'flag.txt' file. The mission is fulfilled.	
	Tasks	Enters the command the following commands in the Terminal: 1) `msfconsole` 2) `run` 3) `get flag.txt` from the *home* directory

4. Methodology

HackLearn's evaluation methodology (presented in the Fig 9) adopts several aspects of the models and frameworks presented in section 2 (represented in Fig 9 by circles):

- *Didactic framework*: HackLearn's evaluation methodology adopts the flow process of the didactic framework by embracing its stages (i.e., *Preparation*, *Introduction*, *Interactions* and *Conclusion*) and the phases of assessment (i.e., *pre-game assessment*, *in-game assessment*, *post-game assessment*).
- *COFELET framework*: In the *Interactions* stage learners use HackLearn, which is a simulation game based on the COFELET framework and the COFELET ontology presented in section 2.1. The utilization of the COFELET framework helps in developing and running a hacking simulation game. Additionally, it aids in the design and performance of the *in-game assessment*, as it analytically describes the components that have to be included in such games and the elements that have to be assessed.
- *CKC model & APs*: they are demonstrated as teaching materials in the *Introduction* stage and then they are utilized by learners in the *Interactions* stage to plan and perform their mission. The CKC and the APs utilized in this study are associated with the *il Segreto di Arlecchino* scenario and the SEFs. Learners use the CKC model in the *il Segreto di Arlecchino* scenario as a blueprint for planning their strategy and the APs as patterns for applying hacking techniques. However, in other COFELET scenarios different models can be considered such as the Diamond model [13] and the ATT&CK [14] model for strategy planning and attacking.
- *QC framework*: aided in selecting the characteristics on which the questionnaire of the post-game assessment focuses.

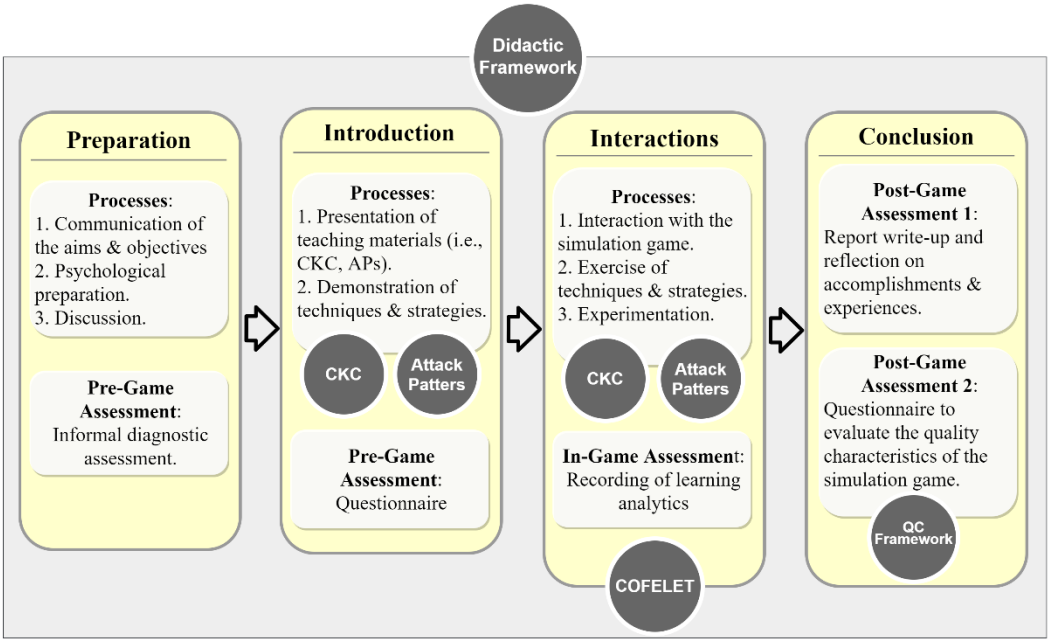


Figure 9. *Evaluation methodology*

In the remainder of this section, the aspects of the HackLearn's evaluation methodology are analytically presented such as the analysis of the questionnaire used in the *post-game assessment 2* of the *Conclusion* stage and the manner that the quality characteristics of HackLearn were assessed.

4.1. Assessments

HackLearn's evaluation mainly performed in the *in-game assessment* of the *Interactions* stage and the *post-game assessments 1* and *2* of the *Conclusion* stage depicted in Fig 9. Previously, the *pre-game assessment* was performed in the *Introduction* stage using a questionnaire containing multiple-choice types of questions, in order to appreciate the students' prior knowledge in penetration testing and cyber-attack strategies. An informal diagnostic assessment was also performed when the instructor asked questions and discussed the concepts of penetration testing during the *Preparation* stage to appraise in situ the prior knowledge of students.

During the *in-game assessment* students involved in the *il Segreto di Arlecchino* scenario with the gaming objective of capturing the file flag.txt. In this process, HackLearn collected learning analytics that provide insights on the students' efforts and achievements in the game's environment and the scaffolding they required. In the *post-game assessment 1* of the *Conclusion* stage students wrote a short report to explain their in-game activities and to express their views on the HackLearn, whereas in the *post-game assessment 2* they filled the post-game questionnaire presented in section 4.2.

4.2. The post-game assessment questionnaire

The design of the post-game assessment questionnaire was based on the quality characteristics of the framework presented in section 2.3. Table 2 lists the questions of the questionnaire along with the assessed quality characteristics of HackLearn (third column) and a question code (first column). Specifically, the question Q1 aims at assessing students' perceptions on how effective (*effectiveness* characteristic) the HackLearn is in teaching the topics of the penetration testing module of their course. The questions Q2 and Q5 refer to the *engagement* and the *motivation* characteristics as their aim is to assess the degree to which the students were challenged by the mission of the *il Segreto di Arlecchino* scenario and the HackLearn's leaderboard feature. The question Q3 aims at examining how

interesting and motivating (*motivation* characteristic) the HackLearn game is. The question Q4 refers to the *usefulness* and *acceptance* characteristics of HackLearn as it aims at assessing how much students like the adoption of serious games, such as HackLearn, in the university course materials. Concludingly, the question Q4 implicitly refers to the *engagement* characteristic, as the acceptance is linked to the *engagement*. The questions Q6 and Q7 refer to the *effectiveness* characteristic, as their aim is to assess the degree to which students believe that the HackLearn's scaffolding features enhanced their performance [18]. Finally, the questions Q8 to Q10 refer to HackLearn's *usability* and *user satisfaction* characteristic, as the questions Q8 and Q9 aim at assessing the game design aspects (e.g., background, colors, icons), whereas the question Q10 aims at assessing how usable and understandable (*understandability* characteristics) the HackLearn's interface is and how much *user satisfaction* it provides.

Table 2. *The post-game assessment questionnaire*

Id	***Question***	***Quality characteristics***
Q1	The utilization of the HackLearn hacking simulator game helped me to comprehend the Cyber Kill Chain model and the attack patterns hackers use to unleash cyber-attacks.	Effectiveness
Q2	The Harlequin mission of the HackLearn game was a challenging assignment.	Engagement, motivation
Q3	I am interested to have more missions in Harlequin.	Motivation
Q4	I would like other courses and subjects to use serious games with simulations (e.g., networks, programming, management, business).	Usefulness, acceptance, engagement
Q5	I would like the top 10 leaderboard to present the scoring of all my colleagues.	Engagement, motivation
Q6	The hints assist me to complete the mission of the game.	Effectiveness
Q7	The teaching contents assist player to recall and/or comprehend some aspects of the game (e.g., tools' usage, description of attack patterns).	Effectiveness
Q8	I liked the colors and the background of the HackLearn game.	User satisfaction, game design
Q9	I liked the icons of the HackLearn game.	User satisfaction, game design
Q10	It is easy to understand how the game interface works to carry out the mission.	Usability understandability, user satisfaction

4.3. Evaluation parameters

HackLearn's evaluation strategy involved the definition of the evaluation metrics used to measure HackLearn's quality characteristics. The evaluation of the HackLearn's *effectiveness* was performed with respect to the students' prior knowledge on the topics of the penetration testing. For the evaluation of *effectiveness*, the following parameters were considered:

i. The recorded number of steps students performed.
ii. The number of in-game questions students answered satisfactorily (i.e., graded over 60%).
iii. How much students think that the utilization of HackLearn helped them to comprehend the topics of the penetration testing lecture (i.e., answers to question Q1).
iv. The recorded number of hints they acquired per step during the game sessions (*hints per step*). Since students had the possibility to play multiple sessions, the *hints per step* were calculated by considering the maximum number of hints per step from all the sessions students played. For example, if a student requested 4 hints in step 2 of her first session and 1 hint in the step 2 of the proceeding session, it was considered that the student requested 4 hints in step 2.

v. How much students valued the support they had from the game's hints on the Likert scale of the questionnaire they answered in the *post-game assessment 2* of the *Conclusion* stage (i.e., answers to question Q6).
vi. How much students valued the support they had from the game's teaching contents on the Likert scale of the questionnaire they answered in the *post-game assessment 2* of the *Conclusion* stage (i.e., answers to question Q7).
vii. Any comments and suggestions made regarding the effectiveness of the game in the *post-game assessment 1* report of the *Conclusion* stage.

The evaluation of HackLearn's *engagement* and *motivation* characteristics was combined, as the engagement is based on the motivation characteristic [17]. Thus, for their combined evaluation, the following parameters were considered:

i. The number of sessions the students performed.
ii. The total time they spent in the game and the average time they spent per session.
iii. The number of actions they performed in the game.
iv. How much interesting and motivating (i.e., question Q3), challenging (i.e., questions Q2 and Q5) and useful (i.e., question Q4) students valued their experience with HackLearn on the Likert scale in the questionnaire they filled in the *post-game assessment 2* of the *Conclusion* stage.

The *usability* characteristic was associated with how much students valued the easiness of use and the understandability of the user interface (i.e., question Q10 which also related with the *understandability* characteristic), whereas the *user satisfaction* and the *usability* were associated with how much students appreciated the design of the game's interface (i.e., questions Q8 and Q9). The *usefulness* and *acceptance* characteristics were associated with how much students would like the adoption of serious games with simulations in the university courses (i.e., question Q4). The characteristics of *user experience*, *usability* and *user satisfaction* were also associated with the related comments and suggestions students made in the report of the *post-game assessment 2*.

5. The experiment

The HackLearn COFELET game was evaluated in the context of the Networks and Web Applications Security course of the Department of Applied Informatics at the University of Macedonia in Thessaloniki, Greece. 103 fourth year (i.e., final year) undergraduate students participated in the experiment.

For the evaluation of HackLearn, the methodology presented in section 4 was employed. Although the adopted didactic framework proposes a flow of processes for business simulation games, the framework was also applied in the evaluation of HackLearn that is a cybersecurity simulation game. The *Preparation* and *Introduction* stages have been conducted in an introductory lecture, which followed a penetration testing lecture wherein the execution of the HackLearn sessions happened. The lecture was delivered on-line through the Zoom platform due to the Covid-19 pandemic. In the *Interactions* stage learners interacted with the HackLearn game. The *Interaction* stage was conducted as an assignment outside the regular class period. In the *Conclusion* stage learners answered a questionnaire and wrote a short report post to the execution of the game sessions. In the remainder of this section the stages of the HackLearn's evaluation process are presented in more detail.

5.1. Preparation stage

In the first part of the introductory lecture, the students were informed on the aims and objectives of the penetration testing part of the course. Specifically, it was made known to the students that they will learn penetration testing concepts and that they will practice cyber-attack techniques and strategies. The students were also informed that they will

interact with a learning environment which provides the opportunities to experiment safely with cyber-attack techniques and it will scaffold their efforts. Additionally, the concept of ethical hacking and the techniques of penetration testing were discussed, and the necessity of ethical hacking was pointed out.

5.2. Introduction stage

The introductory lecture was delivered to the students presenting the Cyber Kill Chain model and the attack patterns of host discovery, port scanning, password recovery exploitation and authentication abuse. Subsequently, the introductory lecture was followed by the penetration testing lecture in which the usage and the syntax of the nmap, ftp, ssh, searchsploit, msfvenom, metasploit, ifconfig and base64 tools were presented. Additionally, the HackLearn game was introduced to the students and a demo scenario was explained, in which a host discovery attack pattern (i.e., the ICMP Echo Request Ping attack pattern) and a port scanning attack pattern were presented along with the decoding of base64 encoded text. During the demonstration students were informed that HackLearn counts participants' scores based on an advanced assessment facility [1] according to which the assessment facility grades participants' efforts by keeping track of their times, the number of actions they perform, the hints they acquire and the number of times they play the game. The top 10 scores are presented in the game's leaderboard.

After the penetration testing lecture, students had one week to play the HackLearn game. Before their first session with HackLearn, students answered the five (5) multiple-choice questions of the *pre-game assessment* questionnaire, in which they declared their prior knowledge and experiences in the lecture's topics (Fig 10).

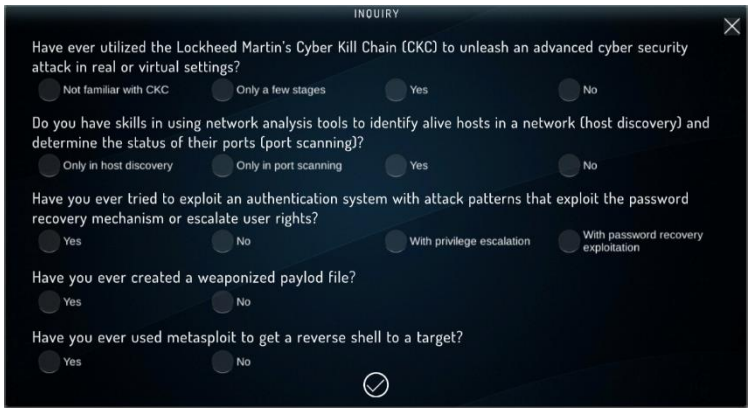


Figure 10. *The pre-game assessment questionnaire*

5.3. Interactions stage

In the *Interactions* stage students created an account as penetration testers, they entered in the game and they followed the interactive tutorial (Fig. 6). Students had the possibility to play the *il Segreto di Arlecchino* scenario several times to achieve the scenario's goal. To do so, students had to develop and implement a strategy that adopts the stages of the CKC model. During the game session, students performed actions and interacted with the game's entities (e.g., network, host, firewall, file system, service) that simulate the behavior of real devices. The student's actions were always followed by the game's feedback as a result of students' activities. The feedback was delivered in textual form through the game's terminal and interface (i.e., score and progress in the progress bar). Therefore, students had the opportunity to refine failed techniques and strategies and to try different approaches. For example, in the step S2 of the *il Segreto di Arlecchino* scenario (Table 1) the students had to change the host discovery ICMP Echo Request Ping attack pattern they initially adopted to find the network's hosts because the game's firewall dropped the ICMP packets. The

Instructors chose to demonstrate the ICMP Echo Request Ping attack pattern in the Introduction stage because it fails in the context of the *il Segreto di Arlecchino* scenario. Thus, the students were led to a cognitive conflict [19].

5.4. Conclusion stage

In the Conclusion stage, students answered the post-game assessment questionnaire containing Likert scale and multiple-choice types of questions and wrote a report. In the report students described their actions, the strategy they employed, their achievements, the pitfalls they identified in the game, the comments on their experiences with HackLearn and suggestions for the improvement of the game. The purpose of the report was to make students reflect on their actions and experiences with HackLearn, and to express their opinion on the game in a more open-ended way than they did with the questionnaire and make suggestions.

6. Results

6.1. Effectiveness

During the *pre-game assessment*, students were asked to answer the questions depicted in Figure 10, based on their prior experiences on applying the CKC and cybersecurity attack patterns and techniques. Instructors chose to ask the students to declare their prior knowledge and not to test it, as it was expected that only a minor percentage of students would have prior knowledge in penetration testing. Besides, instructors had the possibility to preliminary appreciate the students' prior knowledge in the *Preparation* stage. Figure 11 shows the results of the *pre-game assessment* according to which only a minor percentage of students had experiences and knowledge in the penetration testing topics.

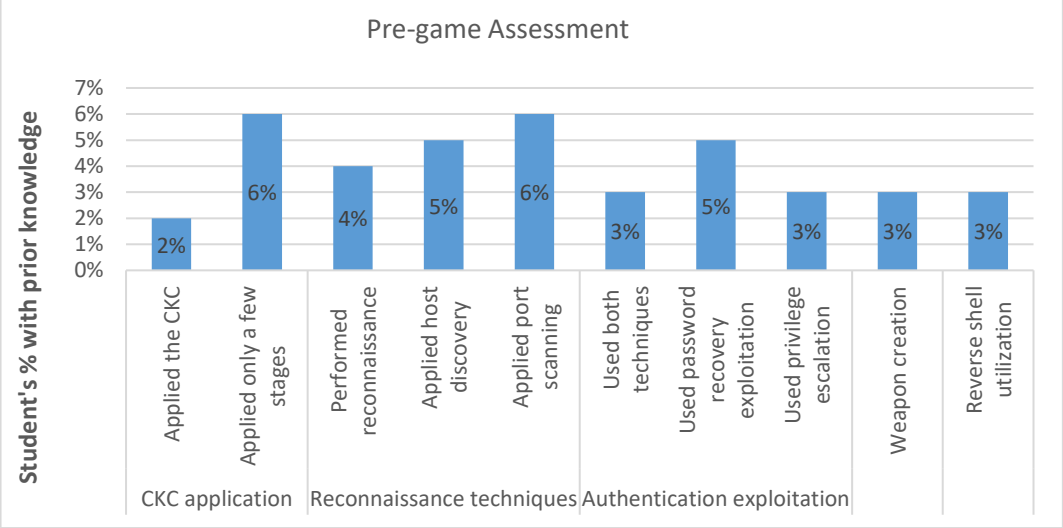


Figure 11. *Pre-game assessment results*

In the *Interactions* stage, 51 students managed to capture the file flag.txt, whereas from the 11 students who declared they had prior experience in penetration testing, 7 students captured the file flag.txt. Almost 66% of the students achieved at least 5 out of the 9 mission's steps (Fig 12).

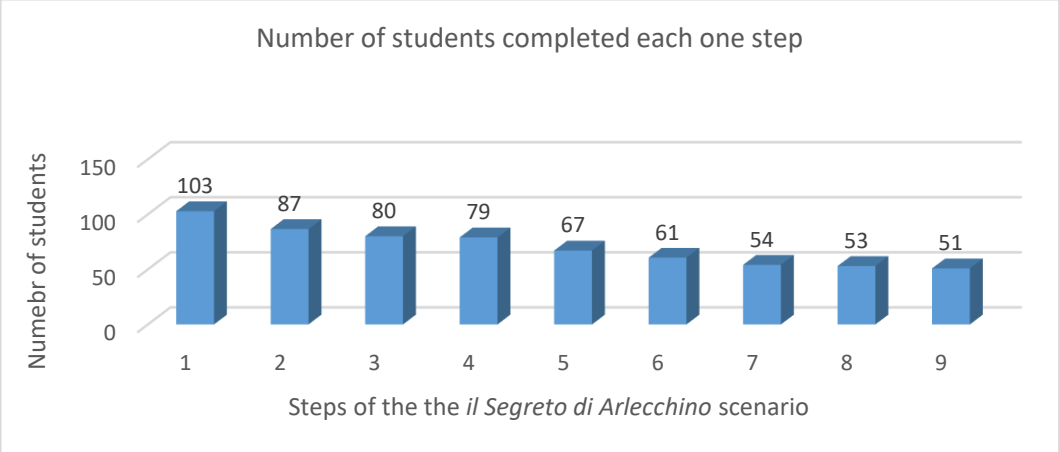


Figure 12. *Number of students per reached step of the il Segreto di Arlecchino scenario*

On average students completed 6,17 steps pe se with standard deviation of 3,18. Each student answered satisfactorily on average 4 questions with standard deviation of 2,66. In the question Q1 of the post-game assessment questionnaire, students showed that they appreciated the usefulness of HackLearn in comprehending the CKC model and the cyber-security attack patterns (Fig 13).

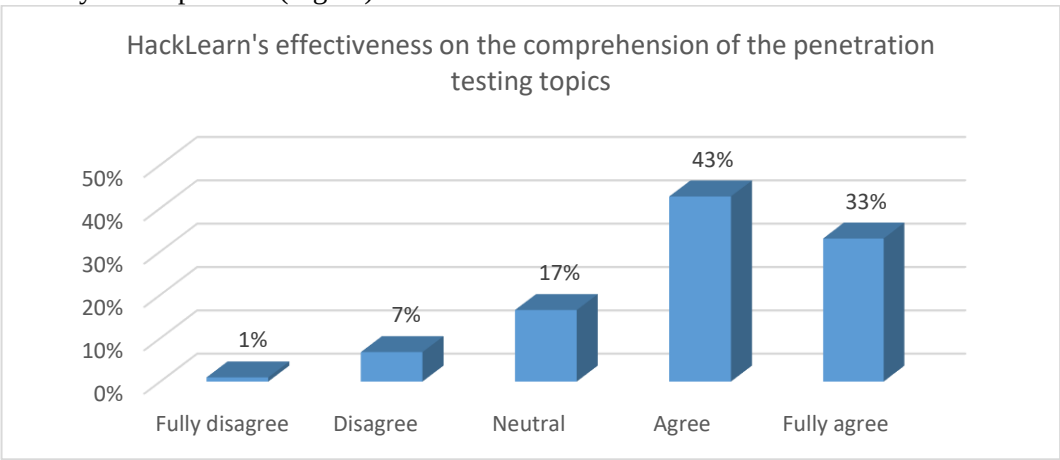


Figure 13. *Percentage breakdown of students' answers to Q1*

Moreover, students showed that they generally appreciated the help they had from the game's scaffolding facilities. Specifically, 69% of the students agreed or fully agreed that game's teaching contents helped them recall and/or comprehend some aspects of the game (i.e., question Q7), whereas 54% agreed or fully agreed that the hints effectively supported them to complete the mission of the game (i.e., question Q6). Though, a considerable percentage of 31% answered that they feel neutral on the support they had from the hints of the game, whereas 16% of the students stated that they disagree or fully disagree that hints helped them to accomplish the mission. According to the game's analytics each student requested 1,49 hints per step with standard deviation of 1,45, whereas the 33% of the students that completed up to the step 4 requested on average 0,92 hints per step.

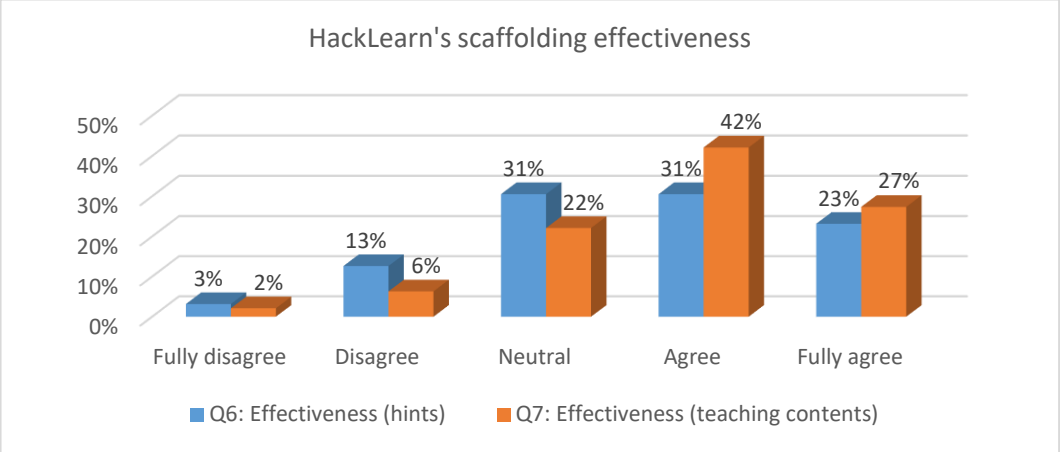


Figure 14. *Percentage breakdowns of students' answers to Q6 and Q7*

In the report of the *post-game assessment 2* of the *Conclusion* stage, more than 60% of the students stated that the game was an efficient and interactive way to learn the topics of the penetration testing module of their course. Some students also stated that their experience with HackLearn raised their awareness of the security policies applied in nowadays (e.g., in the creation of passwords, the protection of accounts). A suggestion that worth's mentioning proposed an enhancement of the game's scaffolding by improving the help option of the *in-game tools* (e.g., `nmap -help`) to provide details on the tool's usage, syntax etc.

6.2. Engagement & motivation

During the *Interactions* stage, 448 sessions were performed and stored in the HackLearn's database. Learners performed an average of 4,36 sessions per se with standard deviation 2,70. On average each user spent approximately 56 minutes in the game (3.396 seconds) with standard deviation approximately 40 minutes (2.452 seconds), and average time 13 minutes per session. Students performed on average 16,78 actions per session with standard deviation 8,71. Moreover, from the 51 students that captured the flag 34 students (i.e., approximately 65%) replayed the mission possibly to improve their records and scores. In the *post-game assessment 1* of the *Conclusion* stage 86% of the students found HackLearn a challenging assignment (i.e., Q2), 66% of the students are interested in playing more scenarios and 92% of the students would like to use simulation games in university's courses. Though, only 45% of the students were interested in finding out through the leaderboard how their colleagues performed in the game.

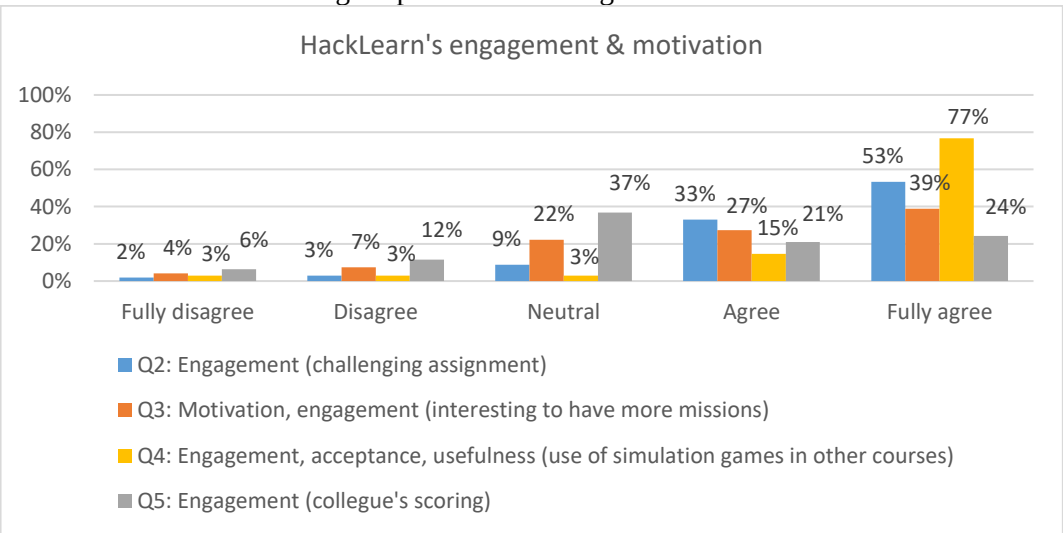


Figure 15. *Percentage breakdowns of students' answers to Q2, Q3, Q4 and Q5*

In the report of the *post-game assessment 2*, most of the students stated that the game was a challenging and interesting experience with clever challenges and they really enjoyed that they learnt new topics in such a practical and efficient manner.

6.3. Usability & user satisfaction

Students showed in the *post-game* assessment questionnaire that they were satisfied with the usability and the game design aspect of HackLearn. Specifically, 70% of the students answered that they found usable the colors and the background of HackLearn, 72% that they understood quickly the meaning of the game's icons (Fig. 16) and 70% stated that it was easy to adopt the manner that HackLearn's works (Fig. 17). However, 27% of the students stated in their report of the *post-game assessment 2* that they experienced connection problems while playing the game and they had to replay the game several times from the first step. At this point, it should be noted that the sessions terminated due to connections problems (terminated sessions) were spotted and excluded from the evaluation process. Additionally, students stated that it was frustrating that the game kept asking answers for the in-game questions, even though students had provided answers in preceding sessions. Students suggested that the game should have a save facility which will save a game session's state and the learners' progress and a load facility that will allow learners to restore their game session and continue from their last check point.

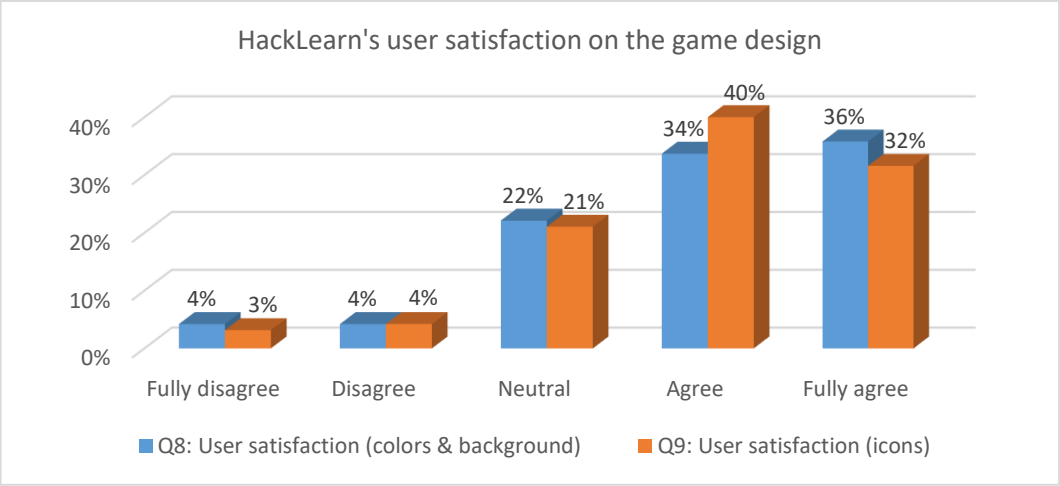


Figure 16. *Percentage breakdowns of students' answers to Q8 and Q9*

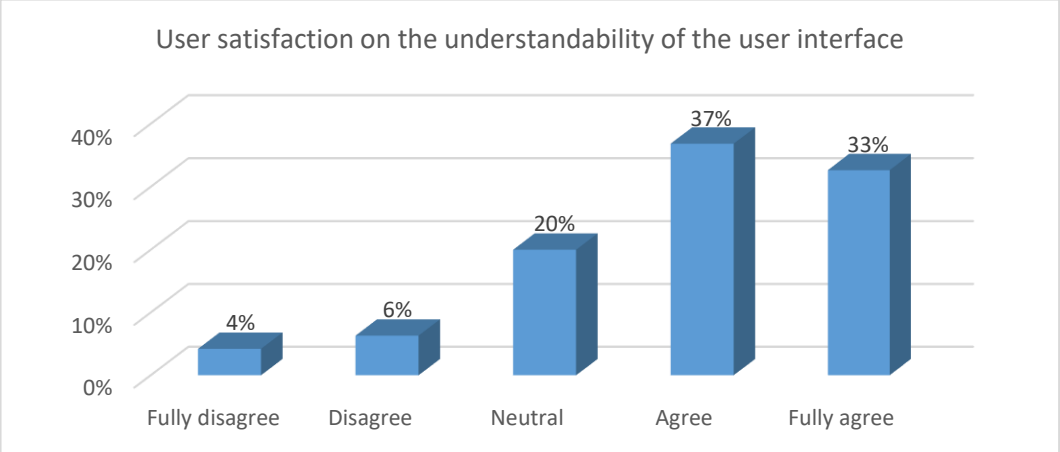


Figure 17. *Percentage breakdowns of students' values on the understandability of HackLearn's user interface*

7. Discussion

The COFELET framework foresees the improvement of cybersecurity education impact, through the development of proper means to deliver effective cybersecurity learning and training. HackLearn is an innovative COFELET game based on modern learning theories and well-known cyber security standards aiming at teaching cyber security concepts while providing hands-on experiences to learners. As HackLearn is the first game of its genre [1], its impact cannot be compared with the impact of other cyber security serious games. However, the results of the presented evaluation can aid in coming to some deductions on HackLearn's impact.

HackLearn has been adopted successfully in a learning approach of a real educational environment, it enhanced a didactic process with many learning benefits, and thus it can be part of the university's course materials. Specifically, according to the HackLearn's analytics, a high percentage of the students were engaged in a game that they played as homework, outside a regular class period. In fact, many students replayed the game several times to achieve the gaming goals or their personal goals (i.e., to increase their scores and make a leaderboard record). The students declared in the *post-game assessment 2* that they considered HackLearn effective in comprehending the module's topics, interesting, challenging, useful and motivating. Many students particularly commented that they enjoyed the HackLearn sessions because it was a challenging task which required critical thinking. Additionally, it is notable that 92% of the students stated that they would like to use serious games with simulations in university's courses (i.e., 77% fully agreed and 15% agreed), a characteristic that shows that students prefer to be active learners instead of passive receivers of information as with traditional teaching methods.

A high percentage of the students stated in the *post-game assessment 1* that the teaching contents and the hints of the game helped them to carry out the mission (i.e., 69% and 54% respectively). However, a considerable percentage of students stated that they did not appreciate the support they had from the game's hints. Besides, the learning analytics show that the students that did not do well (i.e., the students that reached up to the step 4) only requested on average 0.92 per step, whereas one would have expected that they should had used all the support they could get from the game. Thus, more efficient strategies have to be considered for the provision of hints to the learners and especially for learners that find it difficult to function well in the game. Such strategies are the provision of free hints (i.e., hints without score impact) and the formation of attractive and more efficient hints.

In the user satisfaction aspect, although the game has a simple 2D design, most of the students stated that they liked the game design and they used the user interface without difficulties. However, a considerable percentage of the students experienced the session termination problem due to connection problems, as the game could not communicate with the database to store the sessions' analytics and the students' answers to the in-game questions. The connection problem was an intense problem probably due to the instability of the internet during the Covid-19 virus pandemic and in many cases happened due to the students' unstable connection. However, apart from the save and load features suggested by the students, HackLearn can improve the user experience aspect by incorporating a connection examination mechanism and a buffer mechanism. The connection examination mechanism will constantly test the quality of the participants' connection and the buffer mechanism will occasionally store the game's data when temporary connection problems exist. When the connection is stable the buffer mechanism will query its data to the database.

8. Conclusions

The COFELET framework aims at enabling the development of cybersecurity serious games that will enhance the impact of the cybersecurity education. In this study, we

presented the evaluation of the HackLearn COFELET game user experience, which is a scenario-based game aiming at teaching cybersecurity concepts and providing hands-on experiences to the learners. Additionally, we presented some features (e.g., scenarios rational, learners' tasks) of the prototype scenario to provide details on the learners' experiences, the challenges they faced, and the manner they employed cyber security tools, techniques and strategies. We described the manner that HackLearn can be adopted in a real educational setting by adopting the didactic framework [4]. Moreover, we analyzed the methodology we followed to evaluate HackLearn's impact. Specifically, in the presented evaluation process we assessed the game's perceived UX and its effectiveness in teaching the CKC model and the attack patterns hackers apply to unleash their attacks. We assessed how engaged, motivated and satisfied the learners were by HackLearn. The results of our evaluation show that such approaches are very promising since HackLearn was a beneficiary addition in a university's class. Subsequently, our work supports the perspective that serious games can be part of a formal educational system, as students are motivated in learning new topics in more active and creative ways. HackLearn is a hacking simulation game that models and interprets the complex system of a cyberspace in which cyber-attacks take place. Thus, the presented work provides a proof of concept that any real system can be modeled and interpreted in an organized and parameterized learning environment (e.g., serious game), no matter how complex is.

The future work of HackLearn is multi-faced. More scenarios have to be elaborated for different learners' roles and consequently the repository of key elements has to be enriched. Moreover, the HackLearn's scaffolding system has to be upgraded with more features.

References

[1] M.N. Katsantonis, I. Mavridis, "Design and Evaluation of COFELET-based Approaches for Cybersecurity Learning and Training", "Computers & Education," Computers & Security, 2021:102263. https://doi.org/10.1016/j.cose.2021.102263.

[2] M.N. Katsantonis, I. Kotini, P. Fouliras, I. Mavridis, "Conceptual Framework for Developing Cybersecurity Serious Games", In 2019 IEEE Global Engineering Education Conference (EDUCON), 2019 IEEE, pp. 872-881, Dubai, United Arab Emirates, 2019. https://doi.org/10.1109/EDUCON.2019.8725061.

[3] M.N. Katsantonis, P. Fouliras, I. Mavridis, "Conceptual analysis of cybersecurity education based on live competitions", 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 771-779. https://doi.org/10.1109/EDUCON.2017.7942934.

[4] M.C. Utesch, "A Successful Approach to Study Skills: Go4C's Projects Strengthen Teamwork", International Journal of Engineering Pedagogy, 6(1), 2016. https://doi.org/10.3991/ijep.v6i1.5359.

[5] A.J. Abdellatif, B. McCollum, P. McMullan. "Serious games: Quality characteristics evaluation framework and case study", 2018 IEEE Integrated STEM Education Conference (ISEC). IEEE, 2018, https://doi.org/10.1109/ISECon.2018.8340460.

[6] M.N. Katsantonis, I. Mavridis. "Ontology-Based Modelling for Cybersecurity E-Learning and Training", In: Herzog M., Kubincová Z., Han P., Temperini M. (eds) Advances in Web-Based Learning – ICWL 2019. ICWL 2019. Lecture Notes in Computer Science, vol 11841. Springer, Cham. https://doi.org/10.1007/978-3-030-35758-0_2.

[7] M.B. Carvalho, F. Bellotti, R. Berta, A. De Gloria, C.I. Sedano, J.B. Hauge, J. Hu, M. Rauterberg, "An activity theory-based model for serious games analysis and conceptual design", Computers & Education. 87, 166-181, 2015. https://doi.org/10.1016/j.compedu.2015.03.023.

[8] S. Arnab, T. Lim, M.B. Carvalho, F. Bellotti, S. de Freitas, S. Louchart, N. Suttie, R. Berta, A.D. Gloria, "Mapping learning and game mechanics for serious games analysis", British Journal of Educational Technology, 46 (2), 391-411, 2015. https://doi.org/10.1111/bjet.12113.

[9] F.L. Greitzer, O.A. Kuchar, K. Huston, "Cognitive science implications for enhancing training effectiveness in a serious gaming context", Journal on Educational Resources in Computing (JERIC), 7.3: 2, 2007. https://doi.org/10.1145/1281320.1281322.

[10] V.I. Sessa, M. London, "Continuous learning in organizations: Individual, group, and organizational perspectives", Psychology Press, 2015.

[11] MITRE, "Common Attack Pattern Enumeration and Classification (CAPEC)", https://capec.mitre.org, last accessed, 2020. Accessed on: March 30, 2021. [Online]. Available: https://capec.mitre.org/
[12] Lockheed Martin, "Cyber Kill Chain (CKC)", Accessed on: March 30, 2021. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.
[13] S. Caltagirone, A. Pendergast, and C. Betz, "The Diamond Model of Intrusion Analysis" Threat Connect, vol. 298, no. 0704, pp. 1–61, 2013.
[14] B.E. Strom, A. Applebaum, D.P. Miller, K.C. Nickels, A.G. Pennington, C.B. Thomas (2018). Mitre att&ck: Design and philosophy. Technical report.
[15] B. Nilüfer, A. Löffler, R. Heininger, M. Utesch, H. Krcmar, "Evaluation Methods for the Effective Assessment of Simulation Games", In International Conference on Interactive Collaborative Learning, pp. 626-637, Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-11932-4_59.
[16] S.P. Smith, K. Blackmore, K. Nesbitt, "A Meta-Analysis of Data Collection in Serious Games Research", In: Loh C., Sheng Y., Ifenthaler D. (eds) Serious Games Analytic, Advances in Game-Based Learning. Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-05834-4_2.
[17] O. Dele-Ajayi, J. Sanderson, R. Strachan, A. Pickard, "Learning mathematics through serious games: An engagement framework", In 2016 IEEE Frontiers in Education Conference (FIE) (pp. 1-5). IEEE, October 2016. https://doi.org/10.1109/FIE.2016.7757401.
[18] F.D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS quarterly: 319-340, 1989. https://doi.org/10.2307/249008.
[19] T. Mischel, "Piaget: Cognitive conflict and the motivation of thought." Cognitive development and epistemology: 311-355, 1971. https://doi.org/10.1016/B978-0-12-498640-4.50015-9.