An ontology-based model for evaluating cloud attack scenarios in CATS – a serious game in cloud security

Tiange Zhao
Siemens AG
Munich, Germany
tiange.zhao@siemens.com

Ulrike Lechner
University of the Bundeswehr Munich
Neubiberg, Germany
ulrike.lechner@unibw.de

Maria Pinto-Albuquerque
Instituto Universitário de Lisboa (ISCTE-IUL)
Lisboa, Portugal
maria.albuquerque@iscte-iul.pt

Didem Ongu
Siemens AG
Munich, Germany
didem.oengue@tum.de

***Abstract*—In recent years, the market of cloud services has been growing rapidly. Consequently, cloud security has become a heavily discussed topic in the industry. If cloud assets are misconfigured, it can lead to severe security issues and be exposured to cybersecurity attacks. It is of great importance that industry practitioners understand the security challenges and their responsibilities to protect cloud assets. We designed a serious game: Cloud of Assets and Threats (CATS) as an enrichment to the traditional training method to empower users of the cloud infrastructure in terms of cloud security awareness. In this work, we propose a new ontology-based model to support the evaluation of the simulated attack scenarios in CATS. We share the implementation details and the algorithm, based on the Common Vulnerability Scoring System (CVSS), which is used in CATS to evaluate the simulated attack scenarios. With this innovative effort, we maximize the level to which the CATS game reflects real-world facts, and provide a reasonable level of abstraction in the serious game. Our work contributes to the body of knowledge by extending the existing ontology and applying it in the evaluator algorithm, which stands at the core of our serious game CATS. We apply CATS in training in the industry and discovered that CATS is a promising approach to help the industry practitioners to understand cloud security concepts and raise awareness about cloud security. Scholars in the academic world benefit from the work by gaining experience in instantiating the design science paradigm.**

***Index Terms*—technology and engineering learning, cloud security, awareness, industry, serious game, shared-responsibility model, ontology-based model**

I. Introduction

Applying cloud deployment, in industrial products and solutions, increases connectivity and improves efficiency. In recent years, the market of cloud services has been growing rapidly. In 2022, the global public cloud services market is expected to grow by approximately 18.8 percent, which amounts to about 490 billion U.S. dollars [24]. Applying cloud service alleviates the working load of the operator in the sense that they don't need to maintain the infrastructure and server. However, cloud assets are prone to multiple security challenges. For instance, the self-service characteristic brings convenience and flexibility but it implies that the responsibility of secure configuration of the cloud asset should be taken by cloud services customer in all three different cloud service models: Infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). It is the responsibility of the cloud service customer to configure the cloud assets, and the cloud services provider only provides the possibility to do so.

A. Motivation and Background

The industry standards and best practices define different roles and responsibilities in securing cloud assets. [15], [16] This message needs to be conveyed to people who configure and deploy cloud assets in their daily work. The traditional way of doing it is through lecture-based classrooms or online training. Many companies organize training regularly with the practitioners. However, with the potential risk of spreading the virus, under the background of COVID-19, training needed to be carried out online. One of the obvious challenges is that it is challenging to keep the trainees engaged and motivated, compared to classroom training. Serious games have been successful in compensating for this disadvantage of online training. Therefore, we explore the possibility of utilizing serious games in training in the industry for raising awareness about cloud security issues and solutions, among industry practitioners.

In our previous work, reporting on the last design cycle of the CATS game, we proposed a prototype and conducted trial runs in an official game event with 108 industrial practitioners, and we received positive feedback for the CATS game. In the current design cycle described in this work, we focus on a refinement of the core evaluation algorithm. In the game, we simulate simplified cloud defense-attack scenarios by inviting the players to apply a couple of cards to build a defense strategy and defend the cloud assets against six given attack scenarios. In this work, we answer the research question of how to improve the current algorithm to better reflect real-world characteristics of cloud security defenses and attacks?

B. Terminology

Before diving into the topic, it is important to define and highlight the terminology we use in this paper and how the terminology relates to our work.

- Defense-in-depth

The use of a defense-in-depth strategy is proposed by Kuipers et al. in their work [17]. It means that instead of relying on one layer of defense in cybersecurity, multiple defenses should be applied. This contributes to a more successful defense strategy and in general, improves protection against cyber threats. Due to the uncertainty of the attacker's move, the defense-in-depth strategy maximizes the coverage of possible attacks and is encouraged in the game.

- Cybersecurity kill-chain

The idea of a cybersecurity kill chain was proposed by Assante et al. in [1]. Instead of targeting systems in a single incident and breach, cyber attackers tend to apply a step-by-step approach. We generalize this result to cloud security: attackers in cloud security incidents plan the attack step-by-step. In the CATS game, the attack scenarios consist of three phases (initial access, launch attack and make impact) instead of a single-step approach.

- The security ontology

The security ontology was developed based on the security relationship model described in the National Institute of Standards and Technology (NIST) Special Publication 800-12 [11]. Fenz et al. proposed a concrete security ontology in their work [6], modeling the security relationship with *Asset, Threat, Control, Vulnerabilities, etc*. In this work, we take the security ontology proposed by Fenz et al. as a starting point and further adjust it to our context: designing a serious game dedicated to raising awareness of cloud security for industrial practitioners.

- Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) [20] is an open framework for communicating the characteristics and severity of software vulnerabilities. CVSS consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic qualities of a vulnerability that are constant over time and across user environments and produce a score ranging from 0 to 10. In our work, we use the base group only to derive the necessary probability for calculation by normalizing the base score to a percentage number. The Base group considers the Exploitability metrics and the Impact metrics. The Exploitability metrics reflect the ease and technical means by which the vulnerability can be exploited. The Impact metrics reflect the direct consequence of a successful exploit. By applying certain defense actions, the base score can be updated. In our work, we assign different *effects* to different defense cards and update the CVSS score accordingly. More details regarding the algorithm will be introduced in the last part of section IV.

C. Organization of this work

This paper is organized as follows: in the first section, section I, we introduce the motivation and background of our research and provide an explanation for the relevant terminology we use in this work. In section II, we present existing work that is helpful or similar to the endeavor of this work. In the method section, section III, we elaborate the research question that we aim to answer through this work and share the scientific method we apply to carry out our research, as well as our experiment setup. In section IV, we introduce in detail the improvement in algorithm as an output of applying the research method. In section V, we present the game events that we have organized, and the approach we take to validate our design. In section VI, we conclude our work by summarizing the limitations and contributions and highlighting the future work planned as the next step.

II. Related Work

With the advent of cloud computing, new security threats arise and cloud security has also become a concern in the information technology industry. Zhang et al. mentioned in their work [26] that users expect a reliable and secure experience, but there is always a risk factor in information technology if it is not thoroughly protected and secured. Zhang et al. [27] analyzed security impacts of cloud security for both customers and operators and concluded that a large part of the existing cloud technology depends on the security provided by the vendors. Cloud vendors, operators, and security vendors provide the means to deliver security in the cloud, but it is left to the practitioners in the industry to configure the cloud assets, thus these practitioners need to learn how to utilize the available infrastructure to configure the cloud assets, and in particular to configure them in a secure way. According to Singh et al [23], who did a survey on cloud security issues and challenges, there is no standard way to build a Service Level Agreement to reduce risk level for cloud customers with providers. To compensate for any leakage caused by the absence of security standardization from the vendor side, practitioners' awareness should be enhanced. Serious games are widely used in education in the cybersecurity area. There are plenty of successful examples of serious games designed for the purpose of raising cybersecurity awareness. Shostack provides a partial list of existing board games for cybersecurity in his webpage [22], which is a list of serious games successful in their fields, and that provide knowledge and raise awareness about cybersecurity. This list has grown over time. The most important reason contributing to the rapid growth in this field is that serious games provide us with an abundant training environment with a great variety of hands-on exercises that are helpful to keep the training participants engaged and focused, just as Susi et al. mentioned in their work [25]. However, still there is a lack of qualification in specific topics of cloud security [29]. The COVID-19 outbreak impacts industries globally [21], and it contributes to an increase in the importance of cloud security awareness, since more applications are deployed in the cloud. To prevent the spreading of the virus by continuing with hybrid training (combining online and onsite training), we adapted our game events to online-only activity as a part of our continuous development method.

In our work, we generalize the concept regarding IT security awareness proposed by Hänsch et al. [12] to cloud security awareness. In their work, they classify IT security awareness into three dimensions: perception, protection, and behavior. With this work, our aim is to raise awareness of security challenges in cloud computing via serious games by using real-world attack scenarios in a simulated environment. Our work includes a framework that allows the players to apply defense cards to build a defense plan against the given cloud security attack scenario. In the backend, the defense plan built by the players is evaluated by an algorithm, whose output is a probability of how likely the defense will withstand the given attack. To develop this algorithm, we extended the ontology proposed by Fenz et al. [7] in their work. Additionally, we consider different roles and responsibilities in defending cloud assets. The attack scenarios are abstracted from real-world hacking activities against known cloud assets and the mapping between the defense and attacks are derived based on the MITRE ATT&CK cloud matrix [4]. In this work, we focus on the extension of the ontology and evaluator algorithm, which is based on the central vulnerability scoring system (CVSS) [20].

In our previous work, we have released a serious game: Cloud of Assets and Threats (CATS), as an enrichment for cloud cybersecurity training [31]. The main purpose of the game is to raise awareness, among industry practitioners, about cloud security issues and solutions. With the support of the feedback we gathered after eight game events with 108 players, who are practitioners in the industry, we continue developing the CATS game.

In this work, we focus on developing a sophisticated evaluation algorithm for the CATS game with the help of CVSS. We combined the MITRE attack [4] and defense models with CVSS [20] for the evaluator algorithm. The MITRE Cloud Matrix [4] groups real-world cloud attack incidents according to their tactics and techniques. We used the MITRE Cloud Matrix in CATS with real-world attack scenarios. In our previous work, we used a straightforward implementation for the CATS evaluator, without considering the individual differences in the mapping of defenses and attacks, but considering all mapping pairs equally instead. By introducing CVSS calculation [18], it is possible to consider the mapping of each defense-attack pair in a scientific and systematic way individually. For this, we get inspiration from CVSS vectors provided by the U.S. National Vulnerability Database (NVD) [8].

III. Methods

In this section, we introduce the method applied to carry out this research work.

A. Research question

In our work, we are dedicated to answer the following research question:

RQ: How to improve the current algorithm to better reflect real-world characteristics of cloud security defenses and attacks?

We would like to understand the important elements to the success of improving the core algorithm of CATS game and we propose our game prototype, validate and improve our design with design iterations. Our work shows a road map to the design and implementation of a useful serious game in the industry. To answer this question, we should identify possible ways to evaluate (Figure 1) and analyze whether the improvement in design is valid and whether the game serves its purpose of raising awareness about cloud security for industrial practitioners. We describe our way of validating the relevant results collected. Figure 1 suggests the new evaluation flow. Blue boxes represent starting and ending points of the evaluation flow. Yellow boxes represent input and output values. Green boxes represent the basic steps of the evaluation flow. The red diamond shape shows a decision on evaluation flow. Evaluation flow continues on each step of the attack plan until every defense is processed for all attacks in the current step of the attack plan.

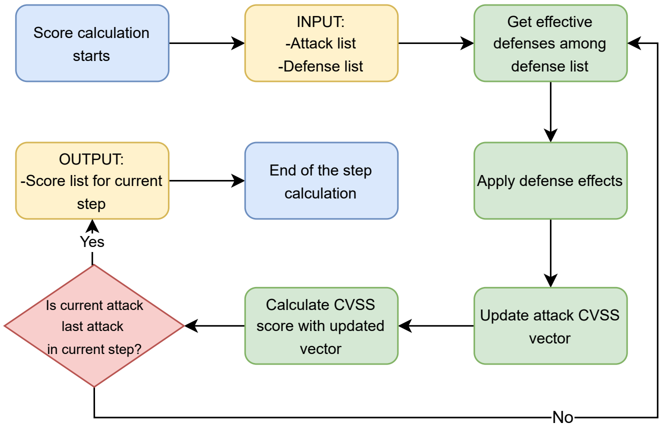
Score calculation starts
INPUT:
-Attack list
-Defense list
Get effective defenses among defense list
Apply defense effects
Update attack CVSS vector
Calculate CVSS score with updated vector
Is current attack last attack in current step?
Yes
No
OUTPUT:
-Score list for current step
End of the step calculation

Fig. 1. New suggested evaluation flow.

Finally, we would like to focus on the details of the abstraction process when improving the current algorithm for such a game and reflect the real-world scenario. The difference between a game and reality is that in the game the irrelevant factors should be simplified or discarded. However, in the meantime, the game should reflect the facts in real-world attack and defense activities. As a game has certain constraints, we would like to identify those constraints. We consider our work as a useful instantiation of converting real-world facts about cloud security into a serious game in which the attack scenarios are simulated and players can decide on different approaches to deal with the given attack scenario. Therefore, it provides inspiration for researchers of how such adaptation can be conducted.

B. Applied Research Methods

We applied the design science research paradigm [13], [14] proposed by Hevner et al. According to Hevner et al., the core of design science research is the cycle of Design & Implement and Justify & Evaluate. We designed and implemented the serious game artifact and organized game events among our target group for justification and evaluation. In the previous design iterations, we proposed the initial game design and validated it with game events attended by industrial practitioners

[31]. In this design iteration, we focus on the refinement of the core evaluator algorithm and the extended security ontology adapted to CATS.

C. Experiment setup, sample, and briefing of participants

We setup our experiments with industrial practitioners within a training framework. In our experiments, we invite the participants to play the CATS game and collect the game dynamic data. In the end of the game events, we distribute a questionnaire to collect feedback.

1) General information on the experiments setup: CATS is a serious game designed to enrich the training material used in the IT industry. The purpose is to raise awareness about cloud security challenges among industrial practitioners. It is available as an online board game on digital platform dedicated for CATS. On the digital platform, the players can join as single players or team up with approximately 4 players per team. We pre-defined six different attack scenarios derived from real-world cloud attack activities [4]. In the game events already performed, it normally took less than 60 minutes to solve all six attack scenarios. The game focuses on the defensive skills of building a defense strategy against the given attack scenario. To build a defense strategy, the player is supposed to pick 6 cards from the defense cards pool and assign them to the correct responsibility: the technical responsibility (4 cards) or the business responsibility (2 cards). In total, there are 23 cards in the defense cards pool for the players to choose from. More details about the game are introduced in our previous work [28]–[30].

2) Game process: As the game starts, we prepare a tutorial scenario for the players to get familiarized with the game platform. Then, the players can select another attack scenario to proceed. In each scenario, a predefined attack is shown on the game interface, and the players are supposed to solve the scenario by building a defense plan, using the available cards in the defense card pool, and submitting the defense plan to the backend. The defense plan will be assessed by the evaluator algorithm, which returns a success rate. The success rate describes the probability that the submitted defense plan withstands the given attack scenario. If the calculated success rate reaches a predefined threshold, the player would get the notification that the scenario is solved and the player can move to the next scenario. If the success rate does not reach the threshold, players can adjust their defense plan by replacing cards or switching cards assigned to the business or technical responsibility according to the hints shown on the game interface. The players are allowed to repeat this step until the scenario is solved. The game is finished if all attack scenarios are solved.

3) Questionnaire: We designed the questionnaire with questions listed in table I. The respondents answer the questionnaire in 5-level Likert scale: Strongly Disagree, Disagree, Neutral, Agree and Strongly Agree. The goal of the questionnaire is to evaluate the level of cloud security awareness of the participants after playing the game and collect feedback from the participants for the improvement in the next design iteration.

TABLE I
QUESTIONNAIRE

No.	Questions
Q1	Playing this cloud security game helps me to understand roles and responsibilities.
Q2	Playing this cloud security game helps me to understand cloud attacks and defenses.
Q3	I benefit from the collaboration with teammates in this cloud security game.
Q4	I benefit from the discussion with teammates in the cloud security game.
Q5	I feel my cloud security know-how has improved by playing this cloud security game.
Q6	I would recommend this cloud security game to other colleagues.
Q7	Our strategy for cloud security will improve by repeatedly playing this cloud security game.
Q8	I think it is hard to calculate the actual probability of a successful defense.
Q9	I think it is hard to consider all relevant factors for a successful defense.

IV. Algorithm Design

In this section, we first introduce the existing CATS game and its important design elements. Then we show the adaption we made to the existing security ontology and the refined evaluator algorithm.

A. The adapted ontology

We adapted the security ontology proposed by Fenz et al. [6]. The ontology obtained is shown in figure 2. The blue rectangles symbolize the concepts which exist in the *original* work of Fenz et al. and the green rectangles are the *adapted* concepts that we extended to apply the ontology to CATS. We provide a brief description of each concept.

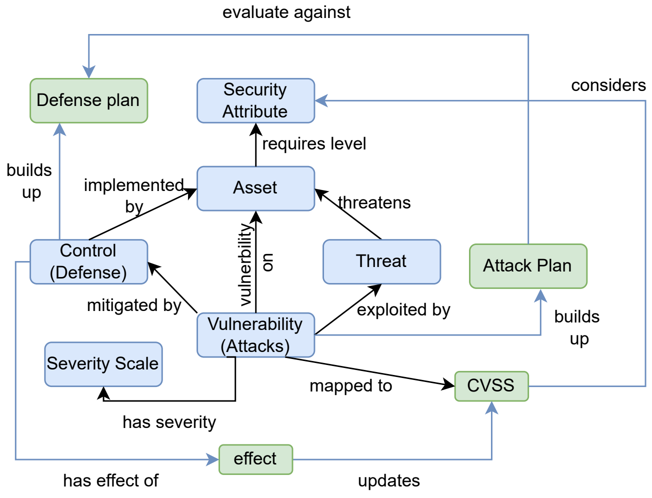

Fig. 2. The ontology extended on work of Fenz et al.

- Security Attribute (*Original*) ⇒ defines which security attributes (e.g. accountability, availability, confidentiality, integrity, reliability, or safety) can be affected by a certain threat. In our work, we consider only confidentiality, availability, and integrity, as used in impact metrics in the CVSS base group.
- Asset (*Original*) ⇒ can be a tangible asset or intangible asset in the original ontology. In our work, we consider mainly the intangible cloud assets, e.g. cloud applications, data, user accounts, etc.
- Vulnerability (*Original*) ⇒ is the absence of a proper safeguard that can be exploited by a threat. Vulnerabilities on *Asset*s exist and could be mitigated by *Control*s. In our work, we map the attacks derived from MITRE ATT&CK cloud matrix [4] to a vulnerability and assign a CVSS score to it.
- Threat (*Original*) ⇒ The threat taxonomy comprises natural (e.g. earthquake, monsoon, or lightning), accidental (e.g. hardware failure or liquid leakage), and intentional (e.g. theft or alteration of software) threats at the highest level, followed by a detailed classification. The threat threatens the assets and the vulnerability is exploited by the threat.
- Control (*Original*) ⇒ has to be implemented to mitigate an identified vulnerability and to protect the respective assets by preventive, corrective, deterrent, recovery, or detective measures. In our work, we map *Control*s to the defenses derived from MITRE ATT&CK cloud matrix [4] and these can be used to reduce the harm of a given *Vulnerability*.
- Severity Scale (*Original*) ⇒ of each vulnerability concept in the original security ontology is rated by a three-point scale (high, medium, and low) in order to enable a machine to interpret the significance of the vulnerability. Since we included the CVSS score, in our work we used the five-point scale according to CVSS vulnerability metrics [19] (None, Low, Medium, High, Critical). The vulnerability with a base score of 0.0 gets the severity scale *None*. The vulnerability with a base score of 0.1-3.9 gets the severity scale *Low*. The vulnerability with a base score of 4.0-6.9 gets the severity scale *Medium*. The vulnerability with a base score of 7.0-8.9 gets the severity scale *High*. The vulnerability with a base score of 9.0-10.0 gets the severity scale *Critical*.
- Defense Plan (*Adapted*) ⇒ In our game, the players are required to combine six *Control (Defense)* cards to build a defense plan. It is a strategy to block the given attack scenario. A full defense plan consists of two defense cards assigned to business responsibility and four defense cards assigned to technical responsibility.
- Attack Plan (*Adapted*) ⇒ In our work, the players are shown six different attack scenarios (attack plan), which describe the step-by-step approach and symbolize the kill chain. The attack plan consists of three steps: initial access (2 *Vulnerabilities (Attacks)*), launch attack (3 *Vulnerabilities (Attacks)*), and make impact (1 *Vulnerability (Attack)*).
- CVSS (*Adapted*) ⇒ In our game, we use the CVSS base score to support the probability calculation in the evaluator algorithm. We map each different *Vulnerability (Attack)* to a CVSS score, some of them are directly available as examples linked to existing incidents on the cloud matrix [4]. When direct mapping is not available, we create and assign a CVSS base score with all the vectors.
- Effect (*Adapted*) ⇒ In our game, it is possible to mitigate the vulnerability by applying individual *Control (Defense)*. The *Effect* describes how the defense mitigates the vulnerability. For example, we assume there is a vulnerability Network Service Discovery [2] mapped to the Common Vulnerability Exposure (CVE) cve-2020-1206 [5]. The vector in base metric of cve-2020-1206 is: Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N with a score of 7.5 (High). We apply the *Control (Defense)* against it, whose effects are AV-1 and S-1. ”AV-1” means to reduce the attack vector (AV) by one category, so the AV is updated from Network (N) to Adjacent (A). ”S-1” means to reduce the Scope (S) by one category. Since the Scope of the original vulnerability is Unchanged (U), which is already the lowest category and cannot be further reduced, therefore ”S-1” does not have any impact on the given vulnerability. By applying the control, the vulnerability's vector is updated to CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N with a score of 6.5 (Medium).

B. The evaluator algorithm

The workflow of the evaluator algorithm is described in figure 3 from a high level. Blue boxes represent starting and ending points of the main flow. Yellow boxes represent input and output values. Green boxes represent the basic steps of the main flow. The purple box represents a generalization for more than one basic steps. The red diamond shape shows a decision on the main flow. The game continues according to flow until the final score reaches the threshold. When the players submit their defense plan to the backend through the game interface, the evaluator starts to work. The evaluator runs on the server side and collects input and feeds the output to the frontend.

The inputs for the evaluator algorithm are 1) the submitted defense cards assigned to the business responsibility; 2) the submitted defense cards assigned to the technical responsibility; and 3) the given attack scenario consisting of 3 steps (initial access, launch attack and make impact). The submitted data is then pre-processed in the data processing step. In this step, the attack scenario is parsed into 3 steps. Additionally, the mapping of the defense to the business or technical responsibility is checked. If the defense card is assigned to the wrong responsibility, it will be removed from the submission and invalidated. If so, this defense card will not have any influence on the calculation in the next step. In the next step of the calculation, the effect of the valid defense card is considered individually, and the effects are applied to the CVSS base vector of the attack one after another. If the lowest level in the CVSS base vector is already reached, the vector will not be further reduced. In that case, the applied defense card is not helpful to the mitigation of the attack and the player should consider replacing this card in the next submission. When all the effects of all the defense cards are applied to all the attacks, the algorithm takes the mitigated CVSS base score rating to the next step: combine step-based scores. In this step, the mitigated CVSS base score is firstly normalized to a percentage number, e.g. the base score 6.5 is normalized to 65% and 10.0 is normalized to 100%. As described in the previous part, each attack scenario has three steps following the idea of a cybersecurity kill-chain. Each step has a different number of attacks: initial access (2 cards), launch attack (3 cards), and make impact (1 card). Within one step, the mitigated attacks are considered in parallel, which means the attack success rate for this step will only be reduced to 0%, if all attacks within this step are mitigated to 0%. If there is one undefended attack with 100% of attack success rate, the attack success rate would be 100%. That case symbolizes that the attack finds an open vulnerability and applies an exploit, therefore bypassing this step without any trouble. We assume the normalized percentage of an attack m in step n is p_{nm}, then the attack success rate of step n p_n is:

$$p_n = 1 - \prod_{1 \ldots m} (1 - p_{nm}) \tag{1}$$

The defense success rate of step n is:

$$\prod_{1 \ldots m} (1 - p_{nm}) \tag{2}$$

Then we consider the three steps in serial order. If an attacker attacks the cloud asset in a three-step approach, he or she must succeed in all the steps to finally compromise the cloud asset. Therefore, if we assume the final attack success rate is p, then:

$$p = \prod_{1 \ldots 3} (p_n) \tag{3}$$

The final defense success rate p_{final} can be derived:

$$p_{final} = 1 - p \tag{4}$$

In the next step, the final defense success rate p_{final} is normalized to a percentage number and if this rate is higher than or equal to the threshold, this attack scenario is solved and the player can move on to solve the next attack scenario. If the final defense success rate does not reach the threshold, it means the defense strategy that the player submitted can be improved further and the player is supposed to adjust the strategy according to the available hints and do a new submission. In CATS game, the players are encouraged to try different combinations and we do not limit the number of attempts.

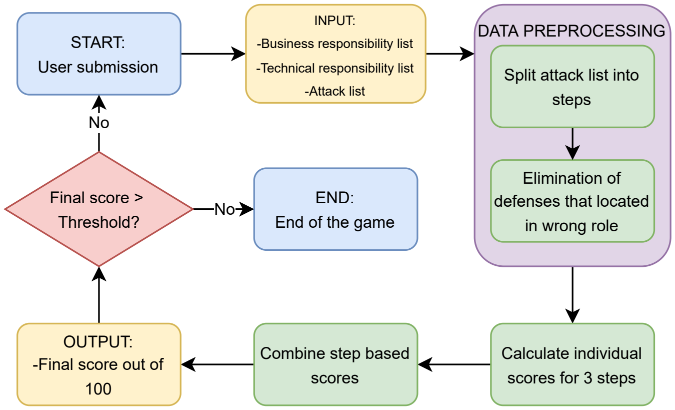
START:
User submission
INPUT:
-Business responsibility list
-Technical responsibility list
-Attack list
DATA PREPROCESSING
Split attack list into steps
Elimination of defenses that located in wrong role
No
Final score > Threshold?
No
END:
End of the game
OUTPUT:
-Final score out of 100
Combine step based scores
Calculate individual scores for 3 steps

Fig. 3. Evaluator algorithm.

V. Findings

A. Game Events

CATS is a serious game designed for practitioners in the industry to raise awareness about cloud security issues and solutions. As Hevner et al. mentioned in the design science paradigm, it is important to continuously evaluate the design artifact. In our work, we evaluate the game with game events. We organized eight game events in the year 2022 with participants from the industry. We aimed to reach a better design for our game by continuously organizing game events and keeping to improve and develop our game design. To get a meaningful result, we collect game dynamic data from the CATS game events and we invited the players directly to give us feedback in a survey or in an open discussion right after the game event. There are two ways in which we organized the game events. The first option is to combine the game events with training. CATS is designed to be an enrichment of the training material to keep the participants engaged and provide them with opportunities for hands-on exercises. In this first option, the game event occurs right after the training, where the basic concepts about cloud security are introduced. In such game events, we limit the play time to one hour and the participants join as single players. The second option is to deploy CATS attack scenarios as a category of tasks in a CyberSecurity Challenge (CSC) [9], [10] event. CSC is a type of event similar to Capture-the-flag (CTF), where players work in teams and get points by solving attack scenarios in CATS. CSC lasts for a full day. Based on our observation, the participating teams tend to focus on and solve CATS challenges within a certain period of time. In CSC, we typically have 4-5 teams with 3-4 players on each team. It is important to note that within one team, players might have different technical backgrounds and reported that the communication with the teammates is beneficial, as well as the possibility to take on different tasks, such as CATS challenges and other coding challenges.

B. Collected data

The data is collected from two sources. The first is the passively recorded game dynamic data, which includes each submission made to the backend and its timeline. The data we collected is shown and ranked in table II. The second data source is the questionnaire we distributed right after the game events and the direct feedback we get from the participants in the open discussion. The collected results are summarized in table I.

TABLE II
DEFENSE CARDS AND RANKINGS

Defense Card	Theoretical Ranking	Game Event Ranking
Account Management	1	9
Network Segmentation	2	3
Restrict Permission	3	10
Logging & Monitoring	4	4
Asset Management	5	13
Filter Network Traffic	5	9
Password Policy	7	4
Audit	7	6
MFA	7	14
Critical Data Protection	10	11
Update Software	10	17
Information Encryption	10	15
Backup Concept	13	14
Application Isolation and Sandboxing	13	9
Vulnerability Scan	13	14
OS Hardening	13	14
IDS	13	7
Remove Unnecessary Feature	13	17
Application Developer Guidance	19	19
User Training	19	18
Account Use Policy	19	6
Software Configuration	22	20
Code Signing	23	21
Spearman’s correlation coefficient		0.67

C. Analysis of collected data

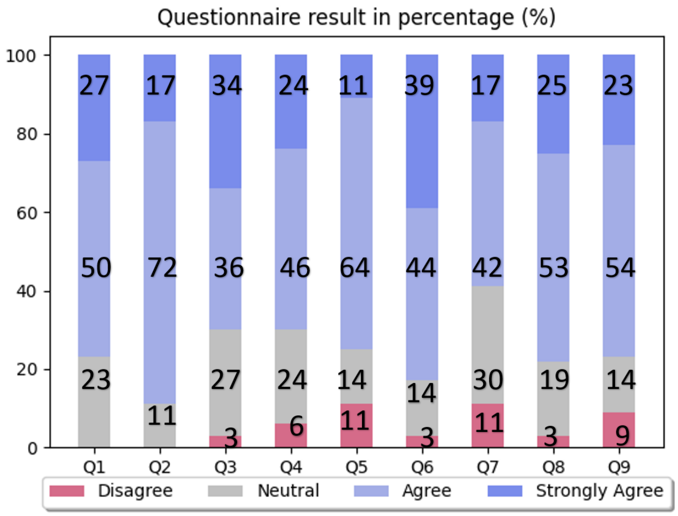

Fig. 4. Questionnaire result.

We analyze game dynamic data as shown in table II and the questionnaire result as shown in figure 4.

We planned the first and second scenarios as warm-ups by simplifying these two scenarios. We observe that participants make a higher number of submissions in the first and second scenarios. Then in the other scenarios, they can succeed with a lower number of submissions. We can interpret it as participants learning about the game logic in the early scenarios and applying the lessons learned in the next scenarios.

Another analysis approach is examining the frequency of each submitted card and comparing the frequency ranking against the most valuable and effective defense cards in the theoretical ranking (Table II). Every defense card affected a different number of attack cards on CATS. This makes some defense cards more valuable. A theoretical ranking can be derived based on the usefulness across all attack scenarios. We use Spearman’s rank correlation coefficient [3] to observe a correlation between ranking in theory and collected data from CATS game events. Spearman’s ρ has a range between ”-1” and ”1”. The value ”-1” suggests that the two compared ranking lists are negatively correlated. That is the case when one list is the reserve of the other. ”0” suggests there is no correlation between the two lists. ”1” suggests that the two compared ranking lists are perfectly correlated. That is the case when two lists are identical. After eight game events, we come to 0.67 for Spearman’s ρ (including the learning phase), which suggests a moderate correlation as shown in table III. This is a positive indicator that players’ performance in the game supports our expectancy. The players understand the game logic and grasp the fundamental concept of cloud security. Another important observation is that for every game event in consecutive submissions for each scenario, participants discover the defense cards with higher effect and are more likely to use more effective defense cards in later scenarios. At this point, we decided to use CVSS to give participants a more precise insight into the effects of defense cards and provide the simulated evaluation in finer granularity, e.g. consider the differences in the mapping defense/attack pairs.

TABLE III
DEGREE OF CORRELATION ACCORDING TO SPEARMAN’S RHO

Range	**Degree of Correlation**
$0<\|\rho\|<0.3$	Weak
$0.3<\|\rho\|<0.7$	Moderate
$\|\rho$—>0.7	Strong

We distribute questionnaires after each game event. We list nine statements and the statistics in table I, which help us to gain insight into the impact the CATS game poses on participants regarding cloud security awareness (Table I). We ask the respondents to rate nine statements in the questionnaire in table I. The respondents were asked to answer whether they ”Strongly Disagree ”, ”Disagree”, ”Neutral”, ”Agree”, or ”Strongly Agree” with the statement. As is shown in the figure 4, the majority of the respondents agree or strongly agree with most of the statements. The figure only shows four out of five level of Likert scale because in the answers we collected, no one replied ”Strongly Disagree” for any of the asked questions. There are a couple of points we would like to highlight:

- The overwhelming majority agree that playing CATS helps them to understand the roles and responsibilities in building a cloud defense strategy. They also find it helpful for them to learn about cloud defenses and attacks.
- The overwhelming majority benefit from the teamwork and the collaboration with the teammates, and think their know-how has been improved by playing CATS.
- Most of the respondents agree or strongly agree that they would like to recommend CATS to other colleagues. Some considered playing CATS repeatedly to improve their strategic thinking in defending cloud assets.
- Most of the respondents also see the difficulty in calculating the probability of whether a certain defense strategy would withstand the given attack and it is also difficult to consider all the relevant factors.

D. Discussion

In this paper, we aim to provide our answers to the research question:

RQ: How to improve the current algorithm to better reflect real-world characteristics of cloud security defenses and attacks?

We followed the design science research paradigm proposed by Hevner et al. [13], [14]. The core of design science research is the cycle of Design & Implement and Justify & Evaluate as a design strategy. The positive responses from the participant feedback from consecutive game events show that our continuous design and implementation method has achieved promising results. The design of our artifact is validated by the game event we organized. By implementing the feedback from the participants we could further improve the designed artifact. Our work has shown that the design science research paradigm is a viable solution to designing a cloud security serious game for industrial practitioners to raise awareness of cloud security issues and best practices.

In our work, we collect data from game events in two different sources: 1) game dynamic data and 2) surveys and feedback in open discussions. We propose a mathematical approach to measuring the players' performance in the game event and we use the measure as evidence of reflecting the level of understanding in cloud attacks and defenses. In our data, we extracted a moderate correlation between the theoretical value and the game dynamics, which implies that the player agrees to the game logic and yet, there is a knowledge gap between their understanding of the cloud security and the model we designed based on MITRE ATT&CK cloud matrix [4] and CVSS calculation [20]. Besides the points mentioned above, we used the survey to collect direct feedback from the participants and the results are overwhelmingly positive in terms of the three dimensions of awareness: perception, protection, and behavior [12]. We find success in applying those two approaches as a way to validate and evaluate such a serious game.

As mentioned in section II, there are various ways to build a security ontology to model the characteristics of cloud assets. The ontology proposed by Fenz et al. inspires us to extend it to a cloud security ontology that can be quantified and applied in the evaluation of a cloud security defense plan. With the calculation of the probability of a defense plan to withstand an attack, the evaluator algorithm is the core of CATS. Another point that we address in our work is to refine the algorithm and derive a quantified method to consider the subtle difference in the individual mapping between the cloud defense- attack pairs. With the help of CVSS, we upgraded our evaluator algorithm in the CATS game from our previous work [29]–[31]. In the current state, participants not only have to cover all attacks in the attack plan with defense cards, but they also have to reach an acceptable protection level in general. Participants can gain a better insight into the power of defense cards thanks to this evaluator algorithm, and to the calculation process supported by the Common Vulnerability Scoring System (CVSS). With this innovative effort, we maximize the level to which the CATS game reflects real-world facts, and provide a reasonable level of abstraction in the serious game. The real-world simulation facilitates utilization in an industrial environment with practitioners, regardless of their technical backgrounds, and serves the purpose of raising awareness of cloud security challenges and solutions.

E. Match and Contribution

This work contributes to ICE 2023 conference, and in particular to the topics: - Industry 4.0, and - Innovation in Engineering Management Education, Research, and Learning. In this work, we present an innovative product for the education of industry IT practitioners - a refinement, with a sophisticated evaluation algorithm, of an innovative serious game to raise cloud security awareness: CATS. This work contributes to the competent use of cloud computing, hence its contribution to the Industry 4.0 topic. Concerning the Field of Interest of the IEEE Technology and Engineering Management Society (TEMS) our work contributes to the practice of implementing cloud technology, specifically through the education and training of the IT industry practitioners that should use it.

VI. Conclusion

In this section, we conclude our work by listing the limitations and the scope of our work; summarizing the existing framework and the conducted research, and providing an outlook into future work.

A. Limitations

The work introduced in this paper is an extension of our previous work [29], [30] on the CATS prototype as a serious game in an industrial setting with the purpose of raising awareness about cloud security. In this work, we explain the details with game design and introduced the ontology approach we applied to further refine the core evaluator algorithm. We provided the evaluation data for the previous work, but only limited number of game events are organized for the new evaluator algorithm. The primary target group for CATS is the practitioners in the industry, which is a focused target group. It limits the number of samples we could take for the evaluation.

B. Concluding Remarks

Our work is guided by the design science paradigm with the purpose of designing and improving a useful artifact in the industry: the CATS serious game. We improve the design of CATS in multiple design interactions and, in each iteration, we evaluate and validate our implementation by organizing a game event with practitioners in the industry. For the evaluation of the game, we propose a mathematical method to measure the performance of the players. Our work provides inspiration for the usage of design science research in designing serious games and contributes to the understanding of the impact of serious games in the industry.

Additionally, we extend the existing body of knowledge on security ontology through the adaptation of the ontology proposed by Fenz et al. [6] to our case. This provides a valuable experience of the application of such ontology.

Furthermore, we apply the CVSS calculation in the core evaluator algorithm of our serious game to enable the quantified calculation of the probability of whether a defense strategy would withstand a certain attack scenario. A similar approach can be useful in other vulnerability simulation environments.

C. Future Work

In the future, we would like to organize game events to test the new features and the ontology-based evaluator algorithm with industrial practitioners and collect direct feedback from them. By analyzing game dynamic data that are passively collected and the feedback that is actively collected, we can validate the adaption of the evaluator we developed with game events data and further improve the game features in the future.

Acknowledgements

This work is partially financed by Portuguese national funds through FCT – Fundação para a Ciência e Tecnologia, I.P., under the projects FCT UIDB/04466/2020 and FCT UIDP/04466/2020. Furthermore, the third author thanks the Instituto Universitário de Lisboa and ISTAR, for their support. We acknowledge funding for project LIONS by dtec.bw. We acknowledge funding for project CONTAIN by the Federal Ministry of Education and Research under the project number 13N16581 and 13N16585.

References

[1] Michael J Assante and Robert M Lee. The industrial control system cyber kill chain. *SANS Institute InfoSec Reading Room*, 1, 2015.

[2] MITRE ATT&CK. Network service discovery.

[3] Leomarich Casinillo and Ginna Tavera. On the dark side of learning calculus: Evidence from agribusiness students. *IJIET (International Journal of Indonesian Education and Teaching)*, 5:52–60, 01 2021.

[4] The MITRE Corporation. Cloud matrix, 2023.

[5] National Vulnerability Database. CVE-2020-1206.

[6] Stefan Fenz and Andreas Ekelhart. Formalizing information security knowledge. In *Proceedings of the 4th international Symposium on information, Computer, and Communications Security*, pages 183–194, 2009.

[7] Stefan Fenz and Thomas Neubauer. Ontology-based information security compliance determination and control selection on the example of iso 27002. *Information & Computer Security*, 2018.

[8] Christian Fruhwirth and Tomi Mannisto. Improving cvss-based vulnerability prioritization and response with context information. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009.

[9] Tiago Gasiba, Ulrike Lechner, and Maria Pinto-Albuquerque. Cybersecurity challenges for software developer awareness training in industrial environments. In *Innovation Through Information Systems*, Lecture Notes in Information Systems and Organisation, pages 370–387, Cham, 2021. Springer International Publishing.

[10] Tiago Espinha Gasiba, Ulrike Lechner, and Maria Pinto-Albuquerque. Cybersecurity challenges: Serious games for awareness training in industrial environments, 2021.

[11] Barbara Guttman and E Roback. An introduction to computer security: the nist handbook, 1995-10-02 1995.

[12] Norman Hänsch and Zinaida Benenson. Specifying IT security awareness. In *2014 25th International Workshop on Database and Expert Systems Applications*, pages 326–330. IEEE, 2014.

[13] Alan Hevner. A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19, 01 2007.

[14] Alan Hevner, Salvatore March, and Jinsoo Park. Design science in information systems research. *Management Information Systems Quarterly*, 2004.

[15] ISO27017. ISO/IEC 27017:2015 Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services, 2015.

[16] ISO27018. ISO/IEC 27018:2019Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors, 2019.

[17] David Kuipers and Mark Fabro. Control systems cyber security: Defense in depth strategies. Technical report, Idaho National Laboratory (INL), 2006.

[18] Peter Mell, Karen Scarfone, and Sasha Romanosky. Common vulnerability scoring system. *IEEE Security & Privacy*, 4, 2006.

[19] National Vulnerability Database. Vulnerability metrics, 2023.

[20] NIST Organization. Vulnerability metrics, 2023.

[21] Ambika Selvaraj, Vishnu Radhin, Nithin KA, Noel Benson, and Arun Jo Mathew. Effect of pandemic based online education on teaching and learning system. *International Journal of Educational Development*, 2021.

[22] Adam Shostack. Tabletop security games + cards, 2022.

[23] Ashish Singh and Kakali Chatterjee. Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications*, 79:88–115, 2017.

[24] Statista. Market growth forecast for public cloud services worldwide from 2011 to 2023**, 2022.

[25] M. Johannesson T. Susi and P. Backlund. Serious games: An overview, 2007.

[26] Zhang Yandong and Zhang Yongsheng. Cloud computing and cloud security challenges. In *2012 International Symposium on Information Technologies in Medicine and Education*, volume 2, 2012.

[27] Ni Zhang, Di Liu, and Yunyong Zhang. A research on cloud computing security. In *2013 International Conference on Information Technology and Applications*, pages 370–373, 2013.

[28] Tiange Zhao, Tiago Gasiba, Ulrike Lechner, and Maria Pinto-Albuquerque. Raising awareness about cloud security in industry through a board game. *Information*, 12(11), 2021.

[29] Tiange Zhao, Tiago Espinha Gasiba, Ulrike Lechner, and Maria Pinto-Albuquerque. Exploring a Board Game to Improve Cloud Security Training in Industry. In *Second International Computer Programming Education Conference*, volume 91, 2021.

[30] Tiange Zhao, Ulrike Lechner, Maria Pinto-Albuquerque, and Ece Ata. Cloud of Assets and Threats: A Playful Method to Raise Awareness for Cloud Security in Industry. In *Third International Computer Programming Education Conference*, volume 102 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[31] Tiange Zhao, Ulrike Lechner, Maria Pinto-Albuquerque, Ece Ata, and Tiago Gasiba. Cats: A serious game in industry towards stronger cloud security. In *Ubiquitous Security*, pages 64–82, Singapore, 2023. Springer Nature Singapore.