Skade – A Challenge Management System for Cyber Threat Hunting

Teodor Sommestad, Henrik Karlzén, Hanna Kvist, and Hanna Gustafsson

Swedish Defence Research Agency FOI

Abstract. When cyber security analysts believe their computer network has been compromised, or feel uneasy about potential intrusions, they might initiate a threat hunting process. The success of a threat hunt is largely dependent on the threat hunter's ability to determine what to investigate, sift through logs, and distinguish normal events from threats. However, these abilities are hard to come by, and it is therefore important to find ways to improve peoples' ability to threat hunt. This paper presents the blueprint for Skade, a system to manage threat hunting challenges. Skade is designed to meet a number of established theories in the field of pedagogy: ensuring constructive alignment, motivating trainees by meeting Turner and Paris' six Cs, providing useful feedback, and covering multiple learning dimensions. Mockups of the user interface of Skade and requirements on supporting scenario emulators are presented, e.g. the data they need to provide to enable generation of feedback to trainees. Seven required functions are identified, e.g. the ability to produce assessment questions based on logs from emulators.

Keywords: cyber security · threat hunting · education · cyber range.

1 Introduction

Large parts of our society's critical infrastructure depend on the industrial control systems and cyber-physical systems. There are many potential cyber threats to these systems, and governing bodies often have specific policies addressing this aspect of cyber security. For instance, the European Network and Information Security Agency (ENISA) have released specific guidance on how to build computer emergency response capabilities for industrial control systems [16]. In line with this, critical infrastructure and related cyber-physical systems are reoccurring themes in cyber defence exercises. For example, various departments within the US government exercised incident handling in the exercise Cyber Storm 2022 [13], and in Locked Shields the "typical scenario relates to the disruption or destruction of critical infrastructure by an adversarial actor" [43].

One type of incident handling activity becoming increasingly popular is *threat hunting*. In threat hunting, an analyst works in hypothesis-driven fashion and looks for things that are suspicious in relation to normal events (e.g. are of unusual frequency), have a connection to some threat intelligence (e.g. a known

malware), or otherwise fit the hypothesis about the threat (e.g. some presumed goal of the threat agent). In general, threat hunting is concerned with finding threats that have evaded the detection systems and signatures already in place. Because many industrial control system environments lack standardized security measures threat hunting is especially relevant to those managing such systems. More specifically, a higher portion of the threats against industrial control systems will need to be detected by human analysts. In addition, critical infrastructure's dependency on industrial control systems suggests that advanced persistent threats such as nation states are targeting industrial control systems. These threat actors have resources and patience to wait for the right moment to use or elevate the privileges they have obtained. For example, the 2015 attack on the systems controlling Ukraine's power grid was preceded by months of attacker-activity within the power company's networks [53]. Similarly, threat agents may aim to degrade the industrial process in ways that do not make cyber attacks the obvious explanation for the problems. For instance, Stuxnet is believed to have been active for years before it was detected [38]. Thus, the hypothesis that someone unauthorized has obtained access to the control system network without making this apparent is plausible.

Thus, it can be argued that threat hunting is particularly relevant to those managing systems running critical infrastructure. Unfortunately, threat hunting is inherently dependent on human expertise. Miazi et al. [37] describe threat hunting as a "highly unstructured task that demands deep technical know-how, data analytics savvy, and out of the box thinking". This paper presents the blueprints for Skade, a challenge management system designed to address the need for human competence in threat hunting within the industrial control system community. Skade, which draws its name from the goddess of hunting in Norse mythology, integrates three main components: a user interface for the trainee, an environment emulator, and a threat emulator (cf. Figure 1). These are used to create synthetic threat hunting scenarios where all details concerning the threat and its traces are known. These scenarios and the ground truth associated with them are used to train people in the process of threat hunting.

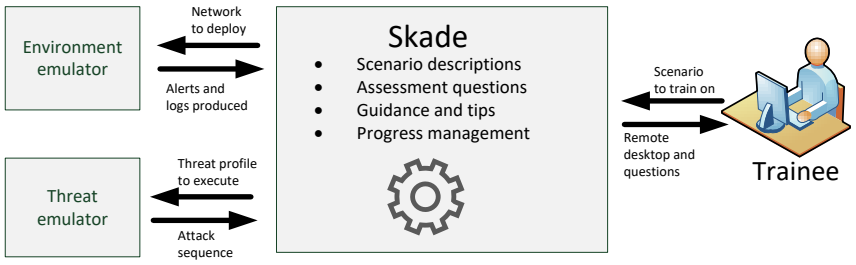


Fig. 1. Illustration of the components of Skade.

Skade's user interface probes trainees for information used to instrument the two emulators, and the user interface is thereafter guided by the user interface to solve the challenge. For example, Skade may probe about the trainee's proficiency level to generate a scenario aimed for this particular proficiency level, probe the trainee with evaluative questions, and assist the trainee with hints in case progress is slow. The contribution of the paper is twofold:

1. Established theories from the field of pedagogy are used to identify how threat hunting training ought to be designed to produce an effective learning experience. This results in a set of hypotheses concerning threat hunting training.
2. The hypotheses are used to identify how the user interface, the environment emulator, and threat emulator should be instrumented to produce training. This results in a blueprint for Skade.

The remainder of this paper is structured as follows. Section 2 provides a brief overview of other initiatives pertaining to threat hunting training, incident scenario generators, and emulators of relevance to threat hunting. Section 3 presents four hypotheses believed to be associated with the learning effect produced in hunting training. Section 4 describes how Skade is designed to address these four hypotheses, and exemplifies how Skade could have been used to realize a recently arranged threat hunting exercise. Section 5 discusses the prospect of Skade as well as future work aimed at developing and evaluating Skade.

2 Related Work

Skade will manage technical challenges designed to train people in the process of threat hunting, specifically people working with computer networks involving industrial control systems. The challenges Skade is designed to manage will be emulated to be technically relevant and realistic. To produce technical challenges or technical environments of this type is by no means a new idea. After reviewing the state-of-the-art in cyber security training for critical infrastructure protection, Chowdhury and Gkioulos [12] "found that delivery methods that offered hands-on experience, in the form of training scenarios and team-based exercises were often preferred over traditional or alternative methods" (such as paper-based teaching and presentations). They also concluded that simulation and virtualization platforms were particularly popular. Similarly, Hajny et al. [21], who reviewed existing curricular guidelines for cyber security, found that many curriculum employ hands-on training and cyber ranges. Thus, to construct virtual environments involving hands-on cyber security challenges is common practice. In line with this, a number of testbeds and emulators specifically focused on cyber environments involving industrial control system have been developed [33, 42, 2, 36].

Skade focuses on building virtual environments containing challenges related to cyber threat hunting. Threat hunting is closely related to incident handling, e.g. the type of processes addressed in exercises such as Cyber Storm [13], Locked

Shields [43], SAFE Cyber [34] and Cyber Czech [49]. However, as described above, the process of threat hunting is highly unstructured and requires considerable technical competence. This is somewhat different from the typical incident handling exercise, which tends to focus on following incident handling processes and collaboration between organizations or team members. Threat hunting challenges tend to focus more on technical analyses. For instance, the threat hunting framework presented in [26] uses the techniques described in MITRE ATT&CK for ICS as basis for log analysis.

There are a few simulations focusing explicitly on producing challenges for threat hunting. Miazi et al. [37] describe experiences from a threat hunting competition arranged on a university campus, concluding that the competition can act as a start for academic threat hunting. Wei et al. [51] developed a university course in threat hunting, with six hands-on assignments. In these assignments, attacks were simulated on virtual machines, and students were given written instructions and pre-prepared questions. The aim was to cover the skills needed during threat hunting across multiple difficulty levels. Both the competition described in [37] and the course described in [51], use pre-defined static scenarios.

A number of platforms have been developed to manage such static scenarios and handle their difficulty etc. Beuran et al. [4] developed CyTrONE, a laboratory scenario management system for cyber security scenarios with automated progression management. The competition platform i-tee [18] has automated handling and scoring for a set of incident handling scenarios in a fictitious cyber environment. The cyber range Kypo [48] is also managing capture-the-flag-tasks, with hints, time limits, and scoring. TopoMojo [10] is another example of a scenario management system, were labs can be built and associated with correct answers. Unlike CyTrONE and i-tee, Skade will manage progress and provide hints to trainees and generate scenarios dynamically by interacting with emulators. Unlike the solutions in CyTrONE, the capture the flag challenges in Kypo, and TopoMojo, Skade focuses on the defensive element and produces threat hunting challenges by emulating threats.

A considerable number of tools and platforms have been developed to emulate cyber environments, and a few have been developed to emulate cyber threats. Examples of environment emulators include ICSTASY [32], Crate [20], CRACK [41], and KYPO [48]; while examples of threat emulators include CARTT [39], SVED [25], Lore [24], and CALDERA [3]. Platforms that combine both environment emulation and threat emulation in an integrated framework have also been developed, e.g. TESTREX [15], Kyoushi [30], and LARIAT [40]. Unlike Skade, these integrated frameworks focus on generating datasets and technical test cases. Skade will instead combine environment emulation and threat emulation in an integrated framework to produce learning outcomes. In other words, Skade aim to combine the ambition of challenge management systems (e.g. TopoMojo) with fully automated scenario emulators (e.g. Kyoushi). Section3 of this paper outlines requirements that arise from this objective.

3 Hypotheses Concerning Threath Hunting Training

We here present hypotheses concerning threat hunting training. The hypotheses are drawn from well-established theories from the field of pedagogy, and concern ensuring constructive alignment, supporting motivating setting, providing feedback and assessment, as well as covering multiple learning dimensions. The underlying theories were chosen to be common in pedagogy, empirically validated, reasonably concretely applicable to threat hunting, and adequately distinct from each other.

3.1 Ensuring Constructive Alignment

Constructive alignment emphasizes both that knowledge is created by the student, rather than merely passed on by the teacher, and that alignment is needed between curriculum objectives, aims, learning activities, and assessments of performance and understanding [6]. Tests also demonstrate that courses designed with constructive alignment in mind foster a deeper understanding in the students [50]. Furthermore, the alignment of learning objectives with learning activity has been found to increase students' motivation to learn, the effort put in, and perceived use of the knowledge [45]. Despite these effects, teachers often forget using constructive alignment, or are not aware of the importance of using it [8].

The usefulness of constructive alignment is widely established in the educational domain, and it has been recommended for security training efforts [11]. Thus, it makes sense to consider constructive alignment when threat hunters are trained. For example, constructive alignment in threat hunting training could start with a clear idea of the purpose of the learning, e.g. training hypothesis-driven threat hunting. From that purpose, a number of learning objectives are defined, e.g. the trainee will be able to 1) understand what hypothesis-driven threat hunting is, 2) use hypothesis-driven threat hunting in Windows environments, and 3) identify known threat actors. From the objectives, a number of learning activities are planned, e.g. 1) watching a recorded lecture on hypothesis-driven threat hunting, 2) practice and perform threat hunting on a small virtual Windows environment and mark machines that are interesting to an attacker, and 3) read about common threat actors in a specific industry. Finally, learning can be assessed in a number of ways, e.g. providing feedback in right/wrong marking of interested machines and grading an essay or quiz relating to threat hunting. Based on constructive alignment theory, it is hypothesized that:

H1 Threat hunting training with constructive alignment will produce larger learning effects than training without constructive alignment.

3.2 Supporting Motivating Setting

Turner and Paris [46] introduced the "six Cs": choice, challenge, control, collaboration, constructing meaning, and consequences. These six features are said to

be critical to creating a motivating learning environment. Based on the theory described in [46], the text below summarizes what each C entails and how it can be addressed to motivate threat hunting trainees.

Choice concerns to what extent students get the opportunity to select activities based on their interests, benefitting their commitment and personal responsibility. In threat hunting training, trainees could be provided a choice of what tasks to focus on in the training. *Challenge* emphasizes the importance of an appropriate difficulty level of the tasks, with tasks that are neither too easy and boring nor frustratingly challenging. In threat hunting training, struggling trainees could be provided clues, in terms of additional threat intelligence. *Control* focuses on student involvement and control over their own learning, e.g. by letting students select tasks and objectives. In threat hunting training, trainees could be provided a selection of scenarios and learning objectives to focus on. *Collaboration* underscores motivation by communication and social interaction, e.g. by having students inspire each other. In threat hunting training, trainees could be provided scenarios that promote collaboration and teamwork. *Constructing meaning* ensures that students understand the value of what they are learning and why, increasing their motivation, e.g. by relating course materials and objectives to real life situations or explaining task rationale. In threat hunting training, trainees could be provided an explanation of constructive alignment (cf. Section 3.1). *Consequence* underlines the students' sharing of their successes and failures, letting them take responsibility of choices in training. In threat hunting training, trainees could be provided their own scores and other trainees' scores for comparison purposes.

There are other ways to classify and describe variables that determine students' motivations. For instance, Epstein introduces the TARGET-framework [17], consisting of the six dimensions Task, Authority, Recognition, Grouping, Evaluation, and Time. However, there is a considerable overlap between these six dimensions and the six Cs. For instance, Task emphasizes a mixture of difficulty among tasks, similar to Challenge in the six Cs. Based on the theory of the six Cs of motivating setting, it is hypothesized that:

H2 Threat hunting training designs that aim to meet the six Cs will motivate trainees more, and produce larger learning effects, than training that does not consider the six Cs.

3.3 Providing Feedback and Assessment

Feedback and assessments are known to have a positive impact on learning [31, 22]. Distinctions are often made between summative and formative assessments, with the former constituting assessment *of* learning, and the latter assessment *for* learning [7]. Summative assessments judge the result after the learning process, and constitute a sort of feedback, while formative assessments give pointers during the training process. Summative assessments indicate when the learning goals are fulfilled, while formative assessments increase motivation and encourage self-assessment [52]. Effective assessment can result in better learning and

make the student "take better ownership of its learning, as opposed to coasting as surface learners" [7]. Research suggests that learning outcomes are higher when the feedback (summative assessment) is given as an explanation, rather than as an evaluation of correctness [28, 35].

The formative assessment in the threat hunting process should reflect how the trainee is performing in the threat hunt. This can be shown in the learning environment as a progress bar, and points or stars given by sub-tasks, e.g. based on how many parts of the threat hunting challenge that have been completed. For the tasks that the trainee failed, hints or correct solutions can be shown to indicate the change in behavior needed for success. Such hints can be made authentic by framing them as threat intelligence, or as anomaly reports by system administrators. The summative assessment at the end of the learning process can indicate to the trainee if the score or level achieved was enough to pass, how the trainee performed compared to other trainees, how the trainee could have done better, and recap the tasks. Based on the theories on feedback and assessments it is hypothesized that:

H3 Threat hunting training with formative and/or summative assessments is associated with greater learning outcomes than training without assessments.

3.4 Covering Multiple Learning Dimensions

Kolb's experiential learning theory (ELT) [29] states that learning is a continuous process where each person enters the learning process with various skills and life experiences. In ELT, Kolb defines two primary dimensions of the learning process. The first dimension represents information gathering, which could be accomplished by either concrete experience or abstract conceptualization. The second dimension describes how the information is processed, either by active experimentation or by reflective observation. The two dimensions are also described as a cyclic learning process, divided in the four phases of concrete experience, reflective observation, abstract conceptualization, and active experimentation.

Kolb's ELT has stood the test of time and has a positive impact on learning [9], has been deemed relevant in cyber security training [27], and has been used to design games in cyber ranges [21]. Figure 2 illustrates how Kolb's process can be related to the "hunting loop" described in [44].

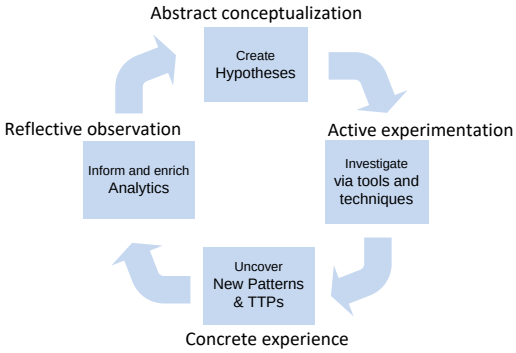


Fig. 2. Kolb's four phases, synthesized with the process of threat hunting from [44]

As Figure 2 suggests, trainees will be required to do abstract conceptualization by constructing hypotheses concerning benign and malicious processes in the computer network. Further, the whole idea of generating hands-on scenarios with emulators is to support active experimentation and foster concrete experience, e.g. to use analysis tools to uncover patterns generated by threat emulators, thus testing the hypotheses and recording the outcome. Finally, reflective observation can be supported, e.g. by encouraging trainees to automate the process with scripts and presenting the ground truth after the challenge. Based on the theory of Kolb, it is hypothesized that:

H4 Threat hunting training that covers all steps in the learning process will produce larger learning effects than designs focusing on individual steps in the learning process.

4 Realization of the Challenge Manager Skade

This section presents how Skade will implement scenarios that address the hypotheses, using an user interface, as well as employing emulators for the environment and for threats. This section will provide concrete examples of trainee interaction and emulator interaction.

4.1 Features

The four hypotheses (H1-H4) presented in Section 3 require a user interface with the ability to present (or not present) information and options. To reflect this, Skade will have a web based user interface where trainees have user accounts and where their progress is recorded. Table 1 summarizes other features of Skade and how they relate to the four hypotheses. To ensure constructive alignment (i.e. H1), threat hunting scenarios in the database will be related to learning

objectives, tasks and requirements, e.g. with a data structure such as Blumberg's course alignment table [8].

Backstories and real-world examples related to the scenario, will also be presented to trainees, in order to construct meaning and motivate learning (H2). Trainees will also be given the option to choose what they should train on, in order to promote a feeling of control (H2).

Summative feedback (H3), in terms of current fulfilment of learning objectives, will be displayed to the trainee at the end of each task. Formative feedback (H3) will be displayed in terms of recommendations and hints. By requesting tips and recommendations, trainees can indirectly lower the difficulty of a task. This mechanism is intended to motivate trainees (H2), by adapting the tasks to be the right kind of challenging, while also offering both choice and control. Recommendations, tips, and options will be related to learning objectives and tasks in order to provide constructive meaning (H2), and ensure constructive alignment (H1). For instance, it will be clearly stated what part of the learning the trainee can skip by choosing to ask for a hint.

Kolb's learning cycle suggests that trainees should be encouraged to go through all steps of the learning process (H4). Skade will therefore include follow-up tasks that encourage trainees to reflect on what they have done. For instance, trainees may be given time to write scripts that automate the threat hunting process they have performed, and thereby be encouraged to recap the more successful parts of their hunting.

The user account of the trainee will record trainee progress. Trainees will be able to compare their accomplishments to other trainees and to predefined benchmarks. This summative feedback (H3) is meant to highlight the consequences (H2) of the learning. Trainees will have the option of measuring their progress on an individual level by taking on scenarios alone, or to form a team with other users to work on scenarios. This option of collaboration (H2) will be made available because it triggers motivation, and because many threat hunting efforts in real life consist of teamwork.

Table. 1. Features in Skade and their relationships to the four hypotheses.

Feature	H1	H2	H3	H4
Presentation of the alignment table for each challenge.	•	•		
Displaying the current fulfilment of learning objectives.	•	•	•	
Backstories and real-world examples related to the scenario.		•		
Giving trainees the option to choose what they should train on.		•		
Enabling optional recommendations and hints at all stages.		•	•	
Presenting the ground truth after the challenge.				•
Follow-up tasks that encourage trainees to reflect.				•

4.2 Functions

All the features described above require the storage of data about scenarios and trainees in a structured manner. The features will also require internal logic that can iterate over scenarios, objectives, and tasks, in order to measure a trainee's progress or a team's progress. More specifically, the internal logic of Skade needs functions capable of providing:

[F1] Textual and visual presentation of scenarios, their objectives, tasks, and requirements.
[F2] Textual and visual descriptions of the emulated networks, e.g. machine names, topology maps, operating systems, users, and settings for log collection.
[F3] Questionnaires for each task, based on the instantiated network, and the threat, e.g. as evidenced by the machines that have been compromised in the network.
[F4] Textual and visual descriptions of the progress of a trainee or team, as provided by querying previously executed scenarios and the objectives the trainees have met.
[F5] Textual tips or recommendations for each task that helps the trainee complete the task, e.g. produce threat intelligence that reveals parts of the attack sequence.
[F6] Textual and visual descriptions of the attack sequence executed by the threat emulator and how this could have been detected, e.g. a bullet point list with time stamps describing the actions taken and artifacts produced.
[F7] Possibility to re-instantiate whole scenarios again, e.g. by instrumenting emulators the same way.

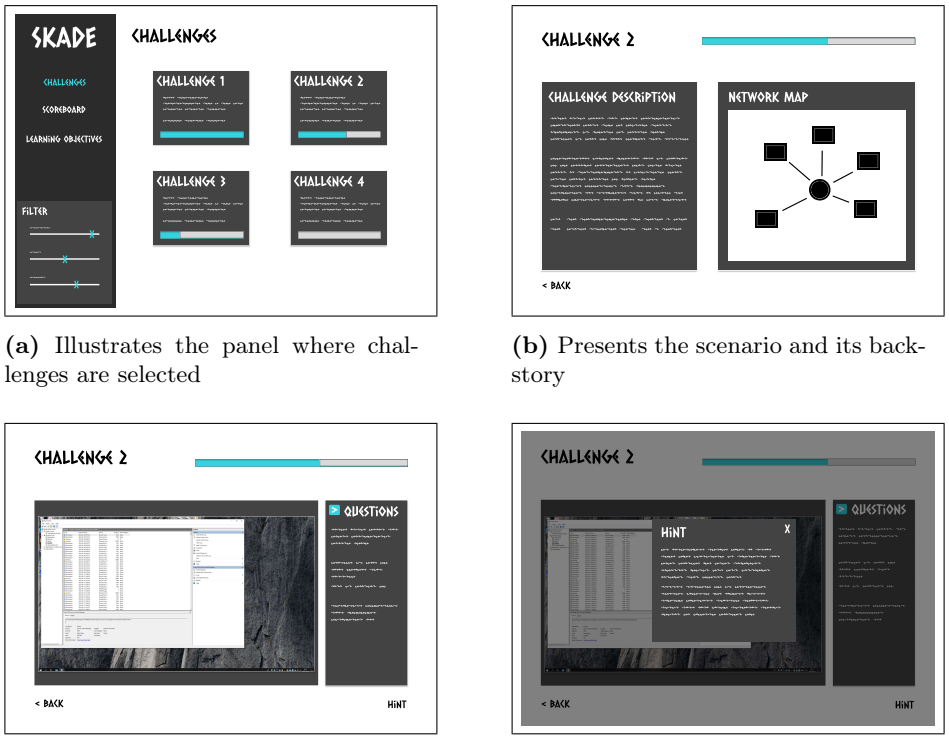
The mapping between the features and the functions can be seen in Table 2.

Table. 2. Features in Skade and their relationships to the seven functions.

Feature	F1	F2	F3	F4	F5	F6	F7
Presentation of the alignment table for each challenge.	•						
Displaying the current fulfilment of learning objectives.			•	•			
Backstories and real-world examples related to the scenario.	•	•			•		
Giving trainees the option to choose what they should train on.							
Enabling optional recommendations and hints at all stages.			•	•			
Presenting the ground truth after the challenge.						•	
Follow-up tasks that encourage trainees to reflect.							•

Functions [F1] enables the presentation of the scenario to the trainee as in Figure 3a and function [F2] is needed to provide the type of background information illustrated in Figure 3b. Functions [F3] and [F4] are needed to provide the functionality related to assessments and feedback as in Figure 3c. Function

[F5] is needed to provide the trainee with hints as illustrated in Figure 3d and to create a reasonable backstory, e.g. providing information about the network and its assets. Function [F6] is required to provide ground truth to trainees after they have completed their hunt, and thereby support reflection and self-evaluation. Function [F7] will allow the creation of follow-up tasks were trainees try again using different methods or try to automate successful parts of a threat hunt.



(a) Illustrates the panel where challenges are selected

(b) Presents the scenario and its backstory

(c) Interaction with machines and challenge questions

(d) Hint provided to a trainee

Fig. 3. Mockups of the user interface.

4.3 Example based on the Nordic-US exercise of 2023

To illustrate the type of challenges Skade will be able to manage, we here present an example drawn from a cyber defence exercise arranged in 2023. The exercise was held in Sweden hosted by The Swedish Civil Contingencies Agency (MSB) [1]. It included participants from government CSIRT (Computer Security Incident Response Team) of the Nordic countries and the USA. The simulated environment was representing a fictive country with vital societal functions and

critical infrastructure that were the target of several cyber attacks, were each attack was treated as separate challenge. In one of these challenges, an insider connected a laptop to the company network and obtained credentials from a domain controller by exploiting the vulnerability CVE-2020-1472 (often referred to as ZeroLogon). Participants were tasked to hunt, report and manage this threat. The text below will explain how Skade could have been used to deliver this challenge in an automated manner, and how the seven functions ([F1]-[F7]) could have been implemented if Skade would have been used.

Function [F1] and [F2] involves giving participants the necessary background information in textual and visual forms. It is straightforward to implement these functions based on the information the exercise management used to instruct trainees (e.g. power point presentations). For instance, Skade could have shown the participants a network topology map and a textual description of the task. The difficulty of this challenge could have been varied by deploying computer networks with more or less logging capabilities or by providing trainees with threat intelligence of different detail. In the Nordic-US exercise standard logging was enabled and the training was initiated by giving participants intelligence suggesting that credentials of the organization had leaked on the dark web.

Functions [F3] and [F4] are straightforward to implement if objectives are defined in a way that is measurable using web forms. Skade could have been loaded with objectives such as: 1) identify obtained credentials by entering the machine they were taken from, 2) identify the MITRE ATT&CK techniques involved in the attack, and 3) attribute the attack to a user or IP address. All of these can be known beforehand or extracted from logs. The attacks in the exercise were scripted using the threat emulator SVED [25]. When the attacker's laptop is connected to the network, SVED produces several logs, for instance as follows:

```
{
    "data": "{\"event\": \"VLAN switch completed.\",
\"new_ip\": \"59.21.4.150\"}",
    "id": 43391680,
    "log_source_id": 5358730,
    "log_source_type": "VLANSwitch",
    "status": "EntityState.SUCCESSFUL",
    "time_stamp": "2023-06-27 15:28:30"
}
```

The IP address (i.e. "59.21.4.150") and the time stamp ("2023-06-27 15:28:30") can be extracted from the log using the following regular expressions in python.

```
r'VLAN switch completed\D*(.*?)\\\"}'
r'VLAN switch completed[\D\d]*\"time_stamp\": \"(.*)\"'
```

Function [F5] requires Skade to have information that can be used to help trainees. The logs from SVED can be used to generate clues to a trainee in a predictable way. For example, intelligence concerning the use of CVE-2020-1472

to target other critical infrastructure, could have been presented as a clue; the time of events could have been read from SVED's logs and presented as a clue; the IP address of the insiders laptop could have been read from SVED's logs and presented as a clue; and the name of the target domain is a parameter of the attack in SVED and could have been given as a clue. Furthermore, the predictability that comes from managing the attack with a threat emulator makes it straightforward to identify logs that are generated by the attack. For instance, in the network of the exercise, the laptop generated logs in the DHCP server when it requested an IP address and the the ZeroLogon exploit could be detected through logs of multiple connection requests and Windows event with ID 4742 with certain content. This could have been stored as clues in Skade. Alternatively, the logs collected in the targeted systems could have been queried using information about the attack (e.g. IP address and timestamp) to identify specific log entries to direct the trainee to.

Function [F6] involves presenting the ground truth in a way that the trainee understands. The logs from SVED's execution contain all the information needed, e.g. machines involved, timestamps and the exploits used. However, to implement this function, Skade would need to process SVED's output and simplify it. For instance, when SVED connects a machine to the network, it produces 15 logs like the one above, including printouts on preparation of different actions and their status at different points in time. Only the completed and successful actions need to be summarized to the trainee.

The use of emulators makes [F7] simple to implement. In this particular exercise the network was emulated using the cyber range Crate [20] and the attack was scripted using the tool SVED [25]. It has already been re-instantiated multiple times with different networks etc.

5 Discussion

This paper has presented the overall idea of Skade, the theory related to the training of threat hunters, and outlined how Skade can be realized using emulators and a user interface. The sections below discuss to what extent Skade meets the requirements of design science suggested by Hevner et al. [23], expand on topics relating to the trainees, elaborate on what the emulators need to cover, give more detail on learning objectives, and outlines a plan to test the hypotheses.

5.1 Skade as a Design Science Effort

Hevner et al. [23] proposed seven guidelines for design science, i.e. research that aims to create new and innovative artifacts. The intention of this project is to create the new and innovative artifact Skade, and the project's compliance with these seven guidelines is therefore of relevance.

The *first* guideline states that a viable artifact must be produced in the form of a construct, a model, a method, or an instantiation. This is straightforward

for the project to fulfill. The Skade system, illustrated in the mockups of Figure 3, intends to be an artifact in the shape of a concrete instantiation. The *second* guideline concerns problem relevance. In Section 1, we argued for the case that threat hunting training is an important and relevant business problem. Section 5.2 further discusses the need for training. *Third*, Hevner et al. stress that the efficacy of a design artifact must be rigorously demonstrated. The evaluation of Skade is far from complete, but plans for validations of utility are outlined in Section 5.5. *Fourth*, design-science research shall make contributions in the areas of the design artifact, design foundations, and/or design methodologies. Skade will be an artifact that solves a previously unsolved problem, i.e. scalable automated training in threat hunting. The *fifth* guideline concerns the rigor of the construction and evaluation of the artifact. Skade has not been constructed or evaluated yet, but Section 5.5 outlines the plans for validation and Section 5.3 outlines the plan for implementation using emulators. The *sixth* guideline stresses that design science is an iterative search process. Accordingly, research on Skade will consider different emulators and solutions within the boundaries given by the theories described in this paper. Further, the boundaries will be set differently if other theories show promise. Finally, the *seventh* guideline concerns communication to both technology-oriented and management-oriented audiences. The overarching project already has a communication plan that covers both of these types of audiences.

5.2 Trainees and Requirements on Challenges

The suitable content for the challenges that Skade provides will depend on the level of expertise of the trainees that use Skade. Our initial analysis is that senior threat hunters will be difficult to please with the type of challenges Skade can provide. This is because experts can be expected to require a high level of realism and fidelity to learning something useful. This would pose extreme requirements on emulated environments, threat emulation, and toolsets provided to trainees. Our initial analysis also suggests that novices, e.g. those unfamiliar with log management, security threats, and basic system administration, will struggle with basic parts of the challenges and gain little from a system such as Skade. Accordingly, Skade will focus on a target audience of intermediate learners.

There is no clear definition of an intermediate learner, but there are several frameworks classifying cyber security practitioners into roles and levels of proficiency. In addition, a survey by Fuchs and Lemon suggests that the most valuable professional background for threat hunting team members, is knowledge in baseline network communications and activity; incident response; threat intelligence and analysis; knowledge in baseline endpoint applications, users and access; and network and endpoint forensics (c.f. Figure 8 in [19]). Based on this, we consider the target audience of Skade to include: senior system administrators from organizations that meet the minimal level of the Threat Hunting Maturity Model (THMM) of organizations [44]; tier 1 and tier 2 of Security Operation Centers [47]; as well as personnel in the roles of Cyber Incident Responder and Cyber

Threat Intelligence Specialist in the European Cybersecurity Skills Framework [14].

5.3 Interaction with Emulators

As described in the introduction, there are a large number of emulators available. The functional requirements of Skade appear to be met by many of these. For example, Kyoushi [30] stores data on the network in configuration files and supports [F4]. CALDERA [3] produces operation reports that support [F5] and [F6]. Skade also requires the possibility of representing meaningful challenges for a threat hunter. Using the number of procedure examples for different attack techniques in MITRE ATT&CK as a proxy for relevance, the following techniques could piece together a relevant partial scenario: initial access via spearphishing attachment (T1566.001), execution via Windows command shell (T1059.003), persistence and escalation via registry run keys (T1547.001), evasion using obfuscated files (T1027), and credential access using keylogging (T1056.001). These types of techniques can be employed in many types of networks and are common in threat emulators. Thus, they do not restrict Skade or threat hunting challenges to a particular set of emulators.

The overall idea of Skade is agnostic to the emulators used, and Skade requires little from the emulators. However, Skade will need to interact with emulators, e.g. send instructions to emulators and interpret logs to generate assessment items. The current plan is focused on the environment emulator Crate [20] and the threat emulator Lore [24]. This choice of emulators is primarily due to practical reasons related to development resources, but also because of the high level of automation that these two emulators offer. Crate has an extensive API for configuration and deployment, which makes it possible to deploy new, diverse, and complex networks, adjusted to fit relevant scenarios. For instance, Crate has scripts that configure collection and signatures in the networks that could be used by Skade. Lore automates the construction of attacks in SVED [25] and the logs can be extracted in the same way as in SVED (the tool mentioned in Section 4.3). It enables multiple profiles with different pre-existing knowledge, differing targets etc. in order to enable the threat hunting scenarios to be created dynamically in a manner appropriate for Skade.

5.4 Learning Objectives and Learning Activities

Section 3 describes hypotheses regarding threat hunting derived from the field of pedagogy. The objectives could relate to planning and communication, or more concrete hands-on-keyboard actions. We envision that Skade cover all three of these, and offer challenges designed for different levels of expertise. Examples of what the learning objectives may entail are described below, together with a brief note on how to adjust the difficulty level of reaching the objectives.

Planning objectives could be to create and evaluate threat or detection hypotheses, in a cyclic manner as described in Figure 2. The hypotheses could be

formed from different focal points, such as identified vulnerabilities, critical systems, crown jewel assets, binaries, indicators of compromise, attack techniques, or threat intelligence. Planning can also include structuring the thought process using the pyramid of pain [5] in order to strike a balance between the most valuable indicators (e.g. attack techniques and tools) and the easiest identifiable indicators (e.g. IP addresses and hash values). Thus, scenarios with different indicators on different levels in the pyramid of pain will need to be emulated.

Communication objectives could include documenting, reporting incidents or communicating with team members. The communication could also concern requests for further information, such as information about vulnerable systems, threat intelligence, or other tools needed, as well as requests to remediate vulnerabilities or perform endpoint hardening. To enable communication learning, Skade will require some way of checking trainee documentation and communication, e.g. by recommending a practice of structuring reports, and automatically checking if trainee reports align with this structure.

Hand-on-keyboard objectives could include the detection of things such as vulnerabilities, insecure practices, misconfigurations, and attack techniques already used in the network. Some examples of the detection of attack techniques in a few attack phases are given in the following. Persistence might be detected by finding which objects use Run and RunOnce or login scripts, as well as which objects that have historically initialized network connections. Command and control (C2) might be detected by looking for anomalies in HTTP requests (e.g. URLs and User-Agent strings), bytes transferred, and duration of connections. Internal reconnaissance might be detected by looking for certain commands spawned by a script (e.g. automated ipconfig). The Skade platform will need to be aware of how trainees could detect things like this in each scenario. In addition, to ensure that training can be transferred to operational contexts, the attack techniques used and the indicators provided will need to be representative of those in operational networks.

A later step in the threat hunting process, as mentioned previously, is to automate each part of the process once the parts have been performed manually. This might include the improvement of automatic detection mechanisms by reducing their false positives and false negatives, or by placing new sensors. To provide such training opportunities, the scenarios, or variants of them, will need to executed on request to test trainee attempts to automate the threat hunt.

One aim of Skade is to offer training for trainees with different levels of expertise by adjusting the difficulty level of the scenario. The difficulty level can be altered by the allotted time to hunt, the provided threat intelligence and logging mechanisms, the complexity and size of the system, the level of background noise, the hunter's familiarity with system, the need to ask for more permissions etc. in the system, and how much the evidence must stack up in order to count as proof.

5.5 Experiment Plan and Tests of Hypotheses

While the prospect of training individuals and teams in threat hunting in a partially automated fashion is appealing, it is not obvious that it is possible to obtain clear learning effects from the type of training that Skade will support. *First*, the threat hunting process is typically thought of as unstructured, making it difficult to create an automated training software for training. *Second*, threat hunting is sometimes said to require some degree of "thinking outside the box", and this is difficult to learn in training. *Third*, threat hunting requires extensive in-depth technical knowledge in terms of normal system behavior, cyberattacks, logging etc. Skade will focus on the ability to combine different kinds of knowledge relating to threat hunting, but it is unclear if this is worthwhile in case individuals lack the various kinds of knowledge that are to be combined. Thus, the utility of Skade will need to be evaluated properly.

The four high-level hypotheses presented in Section 3 can all be tested by comparing Skade to some alternative that is not designed to meet the underlying theory. This alternative could be an instance of Skade purposely modified to be inconsistent with the theory. For instance, H3 states that feedback is important. An experiment can be applied to test the learning outcomes in two conditions: a) training with feedback by Skade and b) training where the feedback is removed. Learning outcomes can be evaluated by simply asking trainees if they learned anything after being exposed to each condition. Previous meta-analyses suggest that feedback improves learning effects with approximately 0.5 standard deviations [28]. Tentative power calculations ($\beta = 0.8, \alpha = 0.05$) suggest that a sample size of 65 trainees will be sufficient in order to detect such effect sizes in a crossover design. The same approach is possible to use for tests of the other hypotheses: the alternative condition for H1 can be scrambled relationships between objectives and tasks; the alternative condition for H2 can be removal of various options for the trainee; and the alternative condition for H4 can be to focus on individual steps.

6 Conclusion

This paper has identified four basic ideas that can be used to guide the design of training in the field of threat hunting: the idea of constructive alignment [6], Turner and Paris' six Cs related to motivation [46], the general idea of providing meaningful feedback, and the four learning dimensions from experiential learning theory [29]. The blueprint of Skade meets these theories, e.g. by presenting challenges in a good way and offering trainees the option to get hints on what to do. A number of publicly available emulators would meet the requirements of Skade. A suitable target audience for a challenge management system is intermediate learners, e.g. senior system administrators. The efficacy of Skade and the design guidelines can be tested in experiments with samples of approximately 65 such trainees.

References

1. MSB hosts international cybersecurity exercise in Sweden (May 2023), https://www.msb.se/en/news/2023/may/msb-hosts-international-cybersecurity-exercise-in-sweden/
2. Almgren, M., Andersson, P., Björkman, G., Ekstedt, M., Hallberg, J., Nadjm-Tehrani, S., Westring, E.: Rics-el: building a national testbed for research and training on scada security (short paper). In: Critical Information Infrastructures Security: 13th International Conference, CRITIS 2018, Kaunas, Lithuania, September 24-26, 2018, Revised Selected Papers 13. pp. 219–225. Springer (2019)
3. Applebaum, A., Miller, D., Strom, B., Korban, C., Wolf, R.: Intelligent, automated red team emulation. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. pp. 363–373 (2016)
4. Beuran, R., Inoue, T., Tan, Y., Shinoda, Y.: Realistic cybersecurity training via scenario progression management. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 67–76. IEEE (2019)
5. Bianco, D.: The pyramid of pain. Enterprise Detection & Response (2013)
6. Biggs, J.: Enhancing teaching through constructive alignment. Higher education **32**(3), 347–364 (1996)
7. Bin Mubayrik, H.F.: New trends in formative-summative evaluations for adult education. Sage Open **10**(3) (2020)
8. Blumberg, P.: Maximizing learning through course alignment and experience with different types of knowledge. Innovative Higher Education **34**, 93–103 (2009)
9. Burch, G.F., Giambatista, R., Batchelor, J.H., Burch, J.J., Hoover, J.D., Heller, N.A.: A meta-analysis of the relationship between experiential learning and learning outcomes. Decision Sciences Journal of Innovative Education **17**(3), 239–273 (2019)
10. Carnegie Mellon University.: TopoMojo: A VM Topology Manager (June 2019)
11. Chanussot, T., Schürmann, C.: Cyber awareness training for election staff using constructive alignment. In: Electronic Voting: 6th International Joint Conference, E-Vote-ID 2021, Virtual Event, October 5–8, 2021, Proceedings 6. pp. 63–74. Springer (2021)
12. Chowdhury, N., Gkioulos, V.: Cyber security training for critical infrastructure protection: A literature review. Computer Science Review **40**, 100361 (2021)
13. CISA: Cyber storm viii: After-action report. Tech. rep. (2022)
14. for Cybersecurity (ENISA), T.E.U.A.: European cybersecurity skills framework. Tech. rep. (2022)
15. Dashevskyi, S., Dos Santos, D.R., Massacci, F., Sabetta, A.: Testrex: a testbed for repeatable exploits. In: CSET (2014)
16. Dufkova, A., Budd, J., Homola, J., Marden, M.: Good practice guide for certs in the area of industrial control systems. European Network and Information Security Agency (ENISA) (2013)
17. Epstein, J.L., for Research on Elementary, J.H.U.C., Schools, M.: Target, an Examination of Parallel School and Family Structures that Promote Student Motivation and Achievement. Report (Johns Hopkins University. Center for Research on Elementary and Middle Schools), Center for Research on Elementary and Middle Schools, Johns Hopkins University (1987)
18. Ernits, M., Tammekänd, J., Maennel, O.: i-tee: A fully automated cyber defense competition for students. ACM SIGCOMM Computer Communication Review **45**(4), 113–114 (2015)

19. Fuchs, M., Lemon, J.: Sans 2019 threat hunting survey: The differing needs of new and experienced hunters. Tech. rep. (2019)
20. Gustafsson, T., Almroth, J.: Cyber range automation overview with a case study of crate. In: Secure IT Systems: 25th Nordic Conference, NordSec 2020, Virtual Event, November 23–24, 2020, Proceedings. pp. 192–209. Springer (2021)
21. Hajny, J., Ricci, S., Piesarskas, E., Levillain, O., Galletta, L., De Nicola, R.: Framework, tools and good practices for cybersecurity curricula. IEEE Access **9**, 94723–94747 (2021)
22. Hattie, J.: The applicability of visible learning to higher education. Scholarship of teaching and learning in psychology **1**(1), 79 (2015)
23. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Management Information Systems Quarterly **28**, 75–106 (03 2004)
24. Holm, H.: Lore a red team emulation tool. IEEE Transactions on Dependable and Secure Computing (2022)
25. Holm, H., Sommestad, T.: Sved: Scanning, vulnerabilities, exploits and detection. In: MILCOM 2016-2016 IEEE Military Communications Conference. pp. 976–981. IEEE (2016)
26. Jadidi, Z., Lu, Y.: A threat hunting framework for industrial control systems. IEEE Access **9**, 164118–164130 (2021)
27. Karjalainen, M., Siponen, M.: Toward a new meta-theory for designing information systems (is) security training approaches. Journal of the Association for Information Systems **12**(8), 3 (2011)
28. der Kleij, F.M.V., Feskens, R.C.W., Eggen, T.J.H.M.: Effects of feedback in a computer-based learning environment on students' learning outcomes. Review of Educational Research **85**(4), 475–511 (Dec 2015). https://doi.org/10.3102/0034654314564881, https://doi.org/10.3102/0034654314564881
29. Kolb, D.: Experiential Learning: Experience As The Source Of Learning And Development, vol. 1. Prentice Hall (01 1984)
30. Landauer, M., Frank, M., Skopik, F., Hotwagner, W., Wurzenberger, M., Rauber, A.: A framework for automatic labeling of log datasets from model-driven testbeds for hids evaluation. In: Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems. pp. 77–86 (2022)
31. Lau, A.M.S.: 'formative good, summative bad?' – a review of the dichotomy in assessment literature. Journal of Further and Higher Education **40**(4), 509–525 (Jan 2015). https://doi.org/10.1080/0309877x.2014.984600, https://doi.org/10.1080/0309877x.2014.984600
32. Lee, D., Kim, D., Lee, C., Ahn, M.K., Lee, W.: Icstasy: An integrated cybersecurity training system for military personnel. IEEE Access **10**, 62232–62246 (2022)
33. Lemay, A., Fernandez, J., Knight, S.: An isolated virtual cluster for scada network security research. In: 1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1. pp. 88–96 (2013)
34. Lif, P., Varga, S., Wedlin, M., Lindahl, D., Persson, M.: Evaluation of information elements in a cyber incident report. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 17–26. IEEE (2020)
35. Mandouit, L., Hattie, J.: Revisiting "the power of feedback" from the perspective of the learner. Learning and Instruction **84**, 101718 (2023)
36. Mathur, A.P., Tippenhauer, N.O.: Swat: A water treatment testbed for research and training on ics security. In: 2016 international workshop on cyber-physical systems for smart water networks (CySWater). pp. 31–36. IEEE (2016)

37. Miazi, M.N.S., Pritom, M.M.A., Shehab, M., Chu, B., Wei, J.: The design of cyber threat hunting games: A case study. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN). pp. 1–6. IEEE (2017)
38. Nakashima, E., Warrick, J.: Stuxnet was work of us and israeli experts, officials say. The Washington Post **2** (2012)
39. Plot, J., Shaffer, A., Singh, G.: Cartt: Cyber automated red team tool. HICSS (2020)
40. Rossey, L.M., Cunningham, R.K., Fried, D.J., Rabek, J.C., Lippmann, R.P., Haines, J.W., Zissman, M.A.: Lariat: Lincoln adaptable real-time information assurance testbed. In: Proceedings, ieee aerospace conference. vol. 6, pp. 6–6. IEEE (2002)
41. Russo, E., Costa, G., Armando, A.: Building next generation cyber ranges with crack. Computers & Security **95**, 101837 (2020)
42. Sitnikova, E., Foo, E., Vaughn, R.B.: The power of hands-on exercises in scada cyber security education. In: Information Assurance and Security Education and Training: 8th IFIP WG 11.8 World Conference on Information Security Education, WISE 8, Auckland, New Zealand, July 8-10, 2013, Proceedings, WISE 7, Lucerne Switzerland, June 9-10, 2011, and WISE 6, Bento Gonçalves, RS, Brazil, July 27-31, 2009, Revised Selected Papers 8. pp. 83–94. Springer (2013)
43. Smeets, M.: The role of military cyber exercises: A case study of locked shields. In: 2022 14th International kypo on Cyber Conflict: Keep Moving!(CyCon). vol. 700, pp. 9–25. IEEE (2022)
44. sqrrl: A framework for cyber threat hunting. Tech. rep. (2018)
45. Stamov Roßnagel, C., Fitzallen, N., Lo Baido, K.: Constructive alignment and the learning experience: relationships with student motivation and perceived learning demands. Higher Education Research & Development **40**(4), 838–851 (2021)
46. Turner, J., Paris, S.G.: How literacy tasks influence children's motivation for literacy. The reading teacher **48**(8), 662–673 (1995)
47. Vielberth, M., Böhm, F., Fichtinger, I., Pernul, G.: Security operations center: A systematic study and open challenges. IEEE Access **8**, 227756–227779 (2020)
48. Vykopal, J., Ošlejšek, R., Čeleda, P., Vizvary, M., Tovarňák, D.: Kypo cyber range: Design and use cases. In: 12th International Conference on Software Technologies. SciTePress (2017)
49. Vykopal, J., Vizvary, M., Oslejsek, R., Celeda, P., Tovarnak, D.: Lessons learned from complex hands-on defence exercises in a cyber range. In: 2017 IEEE Frontiers in education conference (FIE). pp. 1–8. IEEE (2017)
50. Wang, X., Su, Y., Cheung, S., Wong, E., Kwong, T.: An exploration of biggs' constructive alignment in course design and its impact on students' learning approaches. Assessment & Evaluation in Higher Education **38**(4), 477–491 (2013)
51. Wei, J., Chu, B.T., Cranford-Wesley, D., Brown, J.: A laboratory for hands-on cyber threat hunting education. In: Journal of The Colloquium for Information Systems Security Education. vol. 7 (2020)
52. Yüksel, H.S., Gündüz, N.: Formative and summative assessment in higher education: Opinions and practices of instructors. European Journal of Education Studies (2017)
53. Zetter, K., et al.: Inside the cunning, unprecedented hack of ukraine's power grid. Wired **9**, 1–5 (2016)