IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol.007, LIIR485
DOI: https://doi.org/10.52731/liir.v007.485

Reducing Student Hesitation through a Trial-and-Error Cyber Defense Exercise System for Security Beginners

Ichitoshi Takehara [*], Yuki Kami [*],
Koji Kida [*], Keizo Saisho [*]

Abstract

Cyber defense exercises are important for developing cybersecurity personnel capable of responding to increasingly sophisticated attacks. However, beginners often hesitate to execute commands during exercises because of anxiety about system failures. To address this issue, we developed a cyber defense exercise system that allows students to save and restore exercise states. Trial-and-Error function enables learners to retry operations safely and reflect on the results of their actions. In the evaluation experiment, students who had learned basic Linux commands were divided into two groups: one using the developed function and the other not using it. As a result, the former group tended to execute more commands than the latter. This result suggests that the proposed system reduces hesitation during operations. Interview responses also suggested that the system helped students feel psychologically comfortable and encouraged them to explore different defensive operations. These findings indicate that the proposed system can support learning for beginners by reducing hesitation and promoting reflective practice in cyber defense exercises.

Keywords: Cyber Security Training, Security Exercise System, Trial-and-Error, Rewinding

1 Introduction

In recent years, cyber-attacks have become increasingly sophisticated. Moreover, a shortage of skilled security personnel has been reported worldwide [1]. There is an urgent need for personnel who can perform “emergency responses in the event of security incidents” and “respond to the increasingly sophisticated cyber-attacks” [2]. The development of security personnel is imperative. To address this issue, cyber security exercises have become increasingly important [3].

Among various cybersecurity exercises, cyber defense exercises (CDXs) are widely recognized as an effective approach to cybersecurity training [4]. In these exercises, participants in a training play the role of security-response team members and defend a virtual service from cyber-attacks on an exercise system. For example, to defend the system, participants analyze log files to understand the status of the server and perform operations on the firewall to block unauthorized access. The knowledge and experience obtained through the exercise system contributes to enhancing security capability.

Universities are increasingly focusing on security education through practical exercises. The authors adopt cyber defense exercises for security beginners at Kagawa University. A 40-minute cyber defense exercise is conducted for fifty beginner students who have only learned basic

[*] Kagawa University, Kagawa, Japan

Linux commands [5][6]. For a single cyber-attack, there are multiple defense methods available. It is important for students to acquire practical experience of both success and failure by trying multiple methods using commands. However, because university classes have limited number of classes, the exercise is only conducted a few times. Students are unable to try multiple defense methods against a single cyber-attack. From the observation of students during exercise, we found that they had hesitated to execute commands because they were worried about causing negative effects to services. In this paper, we address the challenge that students have difficulty performing cyber defense exercises without hesitation in executing commands.

To address the challenge, we propose enabling students to reflect on and retry the defense methods they have tried. In order to realize this idea, we have developed a cyber defense exercise system "Prote-kun" that enables Trial-and-Error [7][8]. Students can rewind the state of an exercise, allowing them to execute commands without hesitation. In this paper, we describe the design and development of the proposed exercise system and present the results of an evaluation experiment conducted with students who have recently learned basic Linux commands. Their hesitation to execute commands is evaluated based on the number of commands entered and post-exercise interviews.

2 Related work

Conventional exercises are typically designed to follow a fixed sequence, where students address predefined tasks [9]. Educators have expertise and teach definitions and theorems. It is crucial for the educators to guide students by providing correct answers when errors occur. Students may fail to select the correct answer even if the question content is slightly different, as they cannot independently identify the differences. In recent years, it has become acceptable that students can achieve personal development through self-reflection [10]. To solve tasks in experiments, students utilize their knowledge and experience based on observations of the current state and the results of Trial-and-Error. To realize self-reflection in exercise, several educational and learning systems have been developed. Chiken et al. developed a programming learning system to promote coping skills based on previous compile errors [11]. Hirashima et al. developed an error visualization system based on simulation results with incorrect conditions created by students [12][13][14].

Organized by the European Network and Information Security Agency, "Cyber Europe" is Europe's largest cyber exercise. Participants respond to incidents occurring in the exercise system in cooperation with simulated stakeholders such as victims [15]. There are also similar cyber exercises such as "Mini Hardening" [16] and "Micro Hardening" [17] by the Hardening Project. These exercises involve students working together as a team to impersonate a CSIRT. In reality, only a few students apply commands, leaving the rest without the chance to experiment with commands or reflect on defense methods.

In security exercises, Trial-and-Error is becoming increasingly important. Kodera et al. develop a system that provides iterative learning for security exercises [18]. Shiota et al. develop a learning system that allows students to learn the risks of cybersecurity based on failure experiences through scenarios of man-in-the-middle attacks [19]. Ohta et al. are developing an exercise system named "CABIN", which provides rollback functionality, allowing students to rewind the state of their exercise environment [20]. Skopik et al. propose a concept of "Non-linear Cyber Exercises", which enables the reconsideration of security incidents through branching and rewinding, recognizing that cyber crises are often unpredictable [21].

Hoshino et al. develop a student training system for system administrators [22]. It is important

for students to practice the same operations repeatedly by using the rewind function. Their system can rewind only one step back, so students cannot compare defense methods. Furthermore, this system provides a pseudo server by using PHP and JavaScript. "Prote-kun" uses real Linux servers running on virtual machines, enabling students to save the exercise state at any time and restore it using the rewind function.

A cyber defense exercise is a simulation designed to explore how to respond to cyber-attacks. Gharibi et al. reported that simulation is the most widely adopted teaching methodology in nursing education [23]. In this simulation involves placing the learners in patient care situations created by instructors in order to maximize the learning in real situations that learners may encounter. The result is reported that Simulation-based nursing education increases the clinical competency of nursing students. And repetitive practice increases skills and self-confidence through simulation results in long-term retention.

3 Proposal System

3.1 Concept

We propose a system enabling Trial-and-Error in cyber defense exercises for self-reflection. Security personnel need practical experience by applying some defense methods. To gain substantial experience, they need to review tried defense methods against attacks. In this section, we present an exercise using a Trial-and-Error approach, where students can operate some defense methods against attacks. The conceptual diagram of the exercise is shown in Figure 1. As shown in "Conventional", conventional exercises only allow students to try one defense method against an attack. Students can only try Defense method *A* against Cyber attack *A*. As shown in "Proposal", the exercise with Trial-and-Error approach allows students to try multiple defense methods against the same attack. Students can attempt both Defense method *A* and Defense method *A'* against Cyber attack *A*. Following Cyber attack *B*, students can continue Defense method *B'*, built on Defense method *A'*. Thus, this exercise allows students to save their current state individually as a savepoint and restore it during the exercise.

An example of the Trial-and-Error function is shown below:

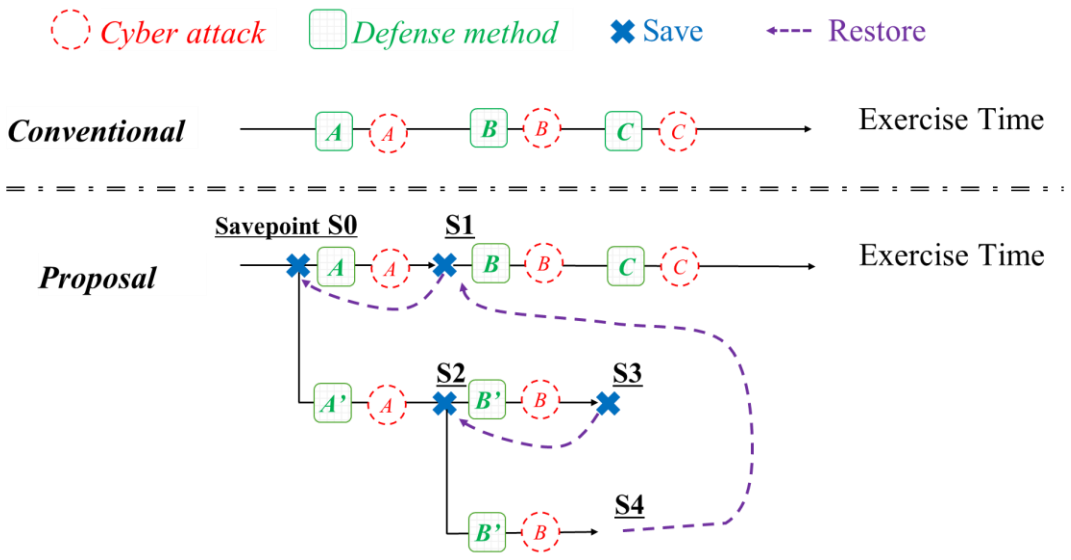


Figure 1: Concept of the Exercise

- The student saves S0 (as savepoint) prior to executing a defense method, considering that the current state could be important in the future.
- The student monitors the result of Defense method *A* against Cyber attack *A*, and then restores the exercise state to S0 for rewinding.
- After restoring to S0, the student tries Defense method *A'* against Cyber attack *A*.

Moreover, the students can restore the state to S1, which is before applying Defense method *B*, based on the results of Defense method *A'* and Defense method *B'*.

Students proceed with the exercise by determining which defense method is more effective against the attack. To do this, the students are provided with the state of an e-commerce site as Defense-Score. Students can reflect on their own defense methods by referencing the Defense-Score. The Defense-Score grows over time as the e-commerce site is accessed during exercise. The Defense-Score does not increase if the e-commerce site cannot meet the requirements of CIA (Confidentiality, Integrity, and Availability). For example, the e-commerce site lacks availability when it cannot be accessed by others.

Therefore, students can proceed with the exercise and execute commands without hesitation, as they are able to rewind the exercise state in case of failure.

3.2 Usage Example

In this section, we present an example of the exercise using the Trial-and-Error approach. The purpose of this exercise is to defend the e-commerce site from cyber-attacks. An e-commerce site is attacked via HTTP requests that exploit vulnerabilities in WordPress [24] plugins. Students can rewind to the point before applying a defense method by saving the state on the system at this moment.

Students can apply defense methods, as follows:

- **Disable Web Access:** The e-commerce site can be protected from attacks using HTTP requests; however, the e-commerce site cannot be provided services. Thus, this method does not ensure availability.
- **Disable Plugins of WordPress:** The e-commerce site can be protected from attacks that use vulnerable plugins; however, its functionality is restricted. Thus, this method does not ensure availability.
- **Upgrade WordPress with Plugins:** The e-commerce site has its vulnerabilities modified and can defend against attacks; however, the site is stopped during the upgrade process. Moreover, students need to perform a data backup and restore data before or after the upgrade according to savepoint to be restored. They also need to verify functionality after upgrade.

By restoring the saved state, students can try the three methods described above and evaluate the results.

In the future, cyber-attacks are expected to become more sophisticated. Through this Trial-and-Error exercise, students can gain experience with various patterns of defense methods and develop the ability to operate them effectively.

3.3 System Configuration

Figure 2 shows the architecture of "Prote-kun". "Prote-kun" consists of the following virtual machines (VMs).

- **DefenseVM:** An e-commerce site is running with vulnerability intentionally for educational purposes. Students defend an e-commerce site from attack by the AttackVM.
- **AttackVM:** It exploits vulnerabilities to attack DefenseVM based on the scenario. The exer-

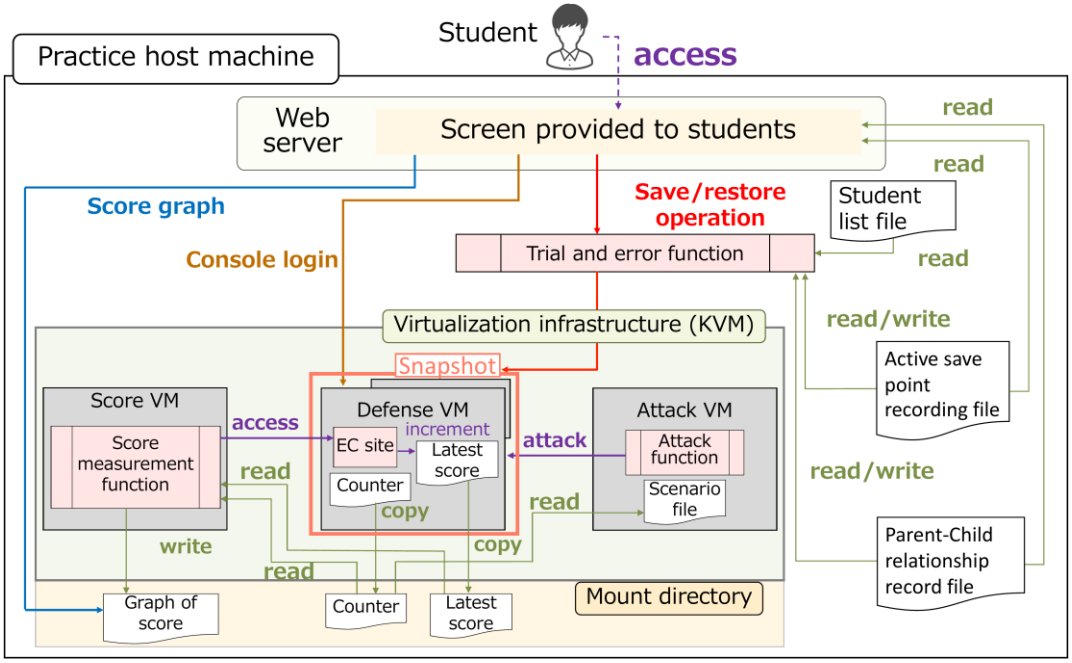


Figure 2: Architecture of "Prote-kun"

cise scenario describes the start time and content of the attack.

● **ScoreVM:** It monitors the risk of vulnerabilities and accesses in an e-commerce site and then calculates Defense-Score from them. Students evaluate a service operation using a calculated Defense-Score.

The Defense-Score transition and attack contents are executed based on the exercise time.

For constructing virtual machines and implementing snapshot function, virtualization platform "KVM" [25] and virtual machine construction software "Vagrant" [26] are used. "QCOW2" [27] which supports the snapshot function is adopted as the virtual disk format.

During the exercise, each student utilizes a Web-based GUI to independently apply the defense methods. Figure 3 shows the web-based interface that students can use to perform the exercise. They use the interface as following:

(1) Students can apply the defense method as commands shown in Panel-A

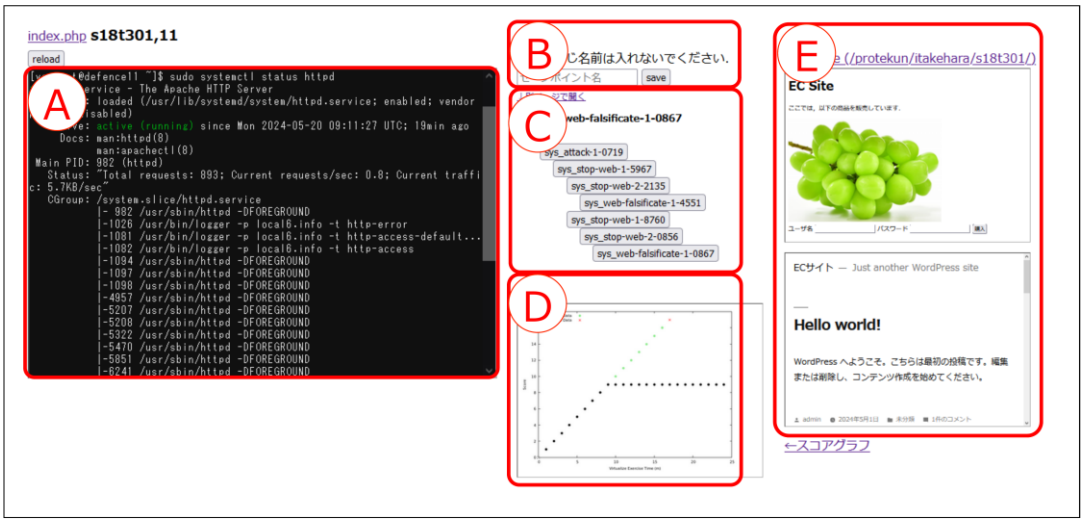


Figure 3: Proposed System GUI

(2) Student access DefenseVM to confirm the service. They can confirm state of Web service Shown in Panel-E.
(3) During exercise, Student can save the state of exercise as a savepoint with a name shown in Panel B and Panel C.
(4) Student can restore the saving exercise state by selecting the button named savepoints.
(5) Student can view the score transitions shown in Panel-D.

3.4 Trial-and-Error Function

This function achieves to save/restore the state of exercise.
● **Save:** It saves the state of the DefenseVM and exercise time when a student clicks the save button with savepoint name in Figure 3-B. The savepoints are shown as a button with related a parent savepoint in Figure 3-C.
● **Restore:** It restores the state of the DefenseVM and the rewinds exercise time when a student clicks the button written a savepoint name.

This function is implemented using the snapshot function of KVM with Vagrant.

3.5 Attack Function

AttackVM attacks the DefenseVM based on Attack scenario that is created by teacher of class. The scenario is configured Attack-start-time, Attack-explanation, and Attack-commands. Figure-4 shows the configuration file. This function launches an attack when the Attack-start-time matches the exercise time, even if it rewound by Trial-and-Error function.

3.6 Score Measurement Function

This function verifies that the availability and integrity of the service are maintained. ScoreVM simulates a customer or user and accesses the DefenseVM. This function checks whether the service is running and has not been tampered with, and calculates a score based on the results. If the service is properly maintained, points are added; otherwise, no points are given. To check the service status, "curl" commands are used to access HTTP/HTTPS on the DefenseVM.

Students can check the transition of their scores using the "Score Transition Graph" shown in Figure 5, which is displayed in Figure 3-D. The x-axis and y-axis of the graph are exercise time and score, respectively. The red dot is the latest score. The numbers in Figure 5 describe the following numbers.
(1) If the DefenseVM isn't attacked, the score increases. The student applies no defense method.
(2) If the DefenseVM is attacked and can't provide service, the score doesn't increase. The student analyzes the attack method and searches for the appropriate defense method to prevent the

```
attack:
  10:   Attack start time
    explain: get root account password
    execdir: /home/vagrant/
    command: ~/bin/attack > password.txt   Attack command
  11:
    explain: send a backdoor using the password obtained by 10
    execdir: /home/vagrant/
    command: ~/bin/sendfile root password.txt backdoor.sh   1step
```

Figure 4: Sample of Attack Configuration File

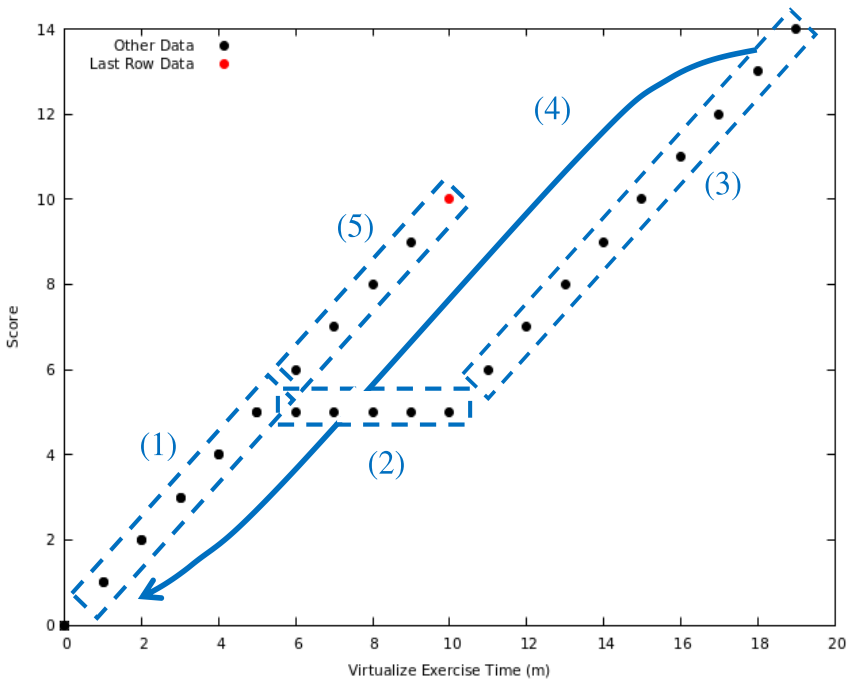


Figure 5: Sample of Score Transition Graph

attack.

(3) The student tries the defense method that he/she discovers, the service gets access from ScoreVM again.

(4) The student wants to try another defense method applied at step (3) before being attacked. He/she restores DefenseVM to exercise time = 0.

(5) The student immediately executes the defense method. The score increases because DefenseVM prevents the attack.

This function uses the same method as the Attack function based on exercise time.

4 Evaluation

4.1 Purpose and Methods

We conducted an evaluation experiment to confirm that "Prote-kun" solved the identified our challenge. We assumed that a higher number of command executions indicates less hesitation during operation. Therefore, the number of executed commands was used as an evaluation metric.

In this evaluation, a 40-minute cyber defense exercise is conducted with seven participants using the proposed system. The participants were students who had recently learned basic Linux commands and had no prior experience in cyber defense exercises. All participants were provided with an explanation of the Cyber Defense Exercise and the Trial-and-Error function. The score status is displayed in the Score transition graph. Participants aim for a linear increase in their scores during the exercise. Four participants enable the Trial-and-Error function; others disable it. To analyze the experimental results, their screens are recorded during the exercise. Afterward, all participants took part in an individual interview for about 20 minutes regarding the Trial-and-Error function.

Table 1 shows the overview of the exercise. The participants act as web server administrators.

Table 1: Overview of Evaluation Experiment

Item	Content
Cover Story	The participant operates an e-commerce site and a blog site, which are targeted by cyber-attacks.
Purpose	Keeps the services running to maintain sales.
Role	Administrator of Web server.
Operation	To operate defense method according to attack.
Feedback	Score transition graph.
Information Source	Cheat sheet about basic Linux commands

The attack scenarios used in the exercise are as follows.

(1) A brute-force password attack is performed to obtain the SSH root user's password. The attacker logs in via SSH and stops the web server twice.

(2) A vulnerability in a content-management-system plugin is exploited to upload a malicious PHP file. The malicious PHP file is accessed to deface the e-commerce site.

These scenarios were configured based on the exercises described in [7][8]. The services operated on the DefenseVM are an e-commerce site and a blog site. The Score function accesses the DefenseVM and adds points when the following conditions are satisfied.

(1) In the e-commerce scenario, the web server is running.

(2) In the blog scenario, the website has not been tampered with.

For condition (2), the website is considered untampered if the pre-downloaded version and the version downloaded at each access are identical.

4.2 Results and Discussion on the Number of Commands

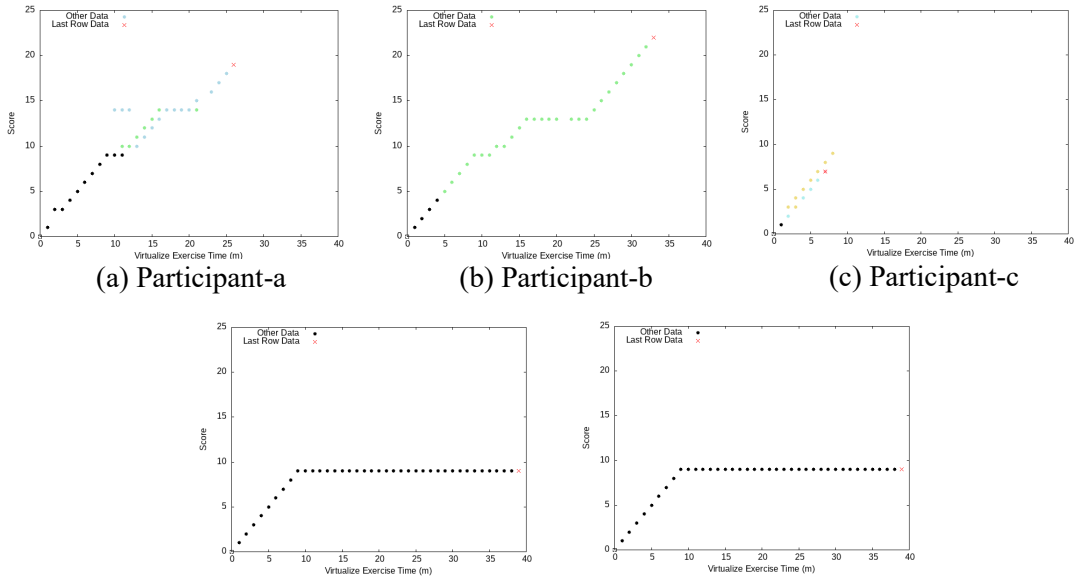
Figure 6 shows a scene of the experiment.

Owing to a measurement error, the number of command inputs could not be measured for one participant in each group. Accordingly, five participants were selected for detailed analysis in the evaluation of command execution numbers.

Figure 7 shows participants' score transition graphs.

(a) This participant executed three save operations and two restore operations. The state was

Figure 6: Scene of the Experiment



(a) Participant-a (b) Participant-b (c) Participant-c

(d) Participant-d (e) Participant-e

Figure 7: Score Transition Graphs of Participants

Table 2: Number of Commands Operated (With/Without Trial-and-Error function)

Trial-and-Error function	Max.	Min.	Ave.
Disable (2 participants)	60	56	58
Enable (3 participants)	74	71	72

saved at the beginning of the experiment. After performing the restore operation, the participant checked the service status and searched logs file. It can be inferred that the participant experienced the same attack twice and attempted to investigate its cause.

(b) This participant executed two save operations and one restore operation. The participant saved the state immediately after the start of the experiment, observed the attacks for about 10 minutes, and then performed a restore. Based on the results of the attack, the participant investigated the logs and searched for the tampered files.

(c) This participant executed seven save operations and thirteen restore operations. The participant frequently used the save and restore functions. This suggests that the functions were used to understand the effects of various commands through observation of their results.

In contrast, the other participants (d) and (e) in the disabled group relied solely on copying and pasting commands from the cheat sheet and failed to restart the service.

Table 2 shows the number of commands executed. When the Trial-and-Error function was enabled, the maximum, minimum, and average numbers of command inputs were 1.23, 1.26, and 1.24 times greater than those without the function, respectively.

4.3 Results and Discussion on the Interviews

After the exercise, all participants were interviewed regarding the items listed below.

(1) What did you learn about cyber-attacks and their defense in this exercise?

(2) What did you think about self-study using this system?

(3) What did you think about the save and restore functions?

(4) What do you think about practicing with this system before the actual operation?

The four interview questions mentioned above were administered to all participants, yielding a total of 28 responses. Interviews were conducted with the Trial-and-Error function disabled

Table 3: Results of the Interview with the Trial-and-Error Function Enabled Group

(a) Positive Comments

Items	Comments
(1)	I realized that if I make mistakes during the exercise, it becomes difficult to carry out subsequent defense actions. Because I could rewind the exercise, I noticed that it was easier to defend when I responded to the attacks properly from the beginning.
(2)	When I got stuck, I was able to rewind and try different defense methods, which increased my options. This time, I tried various approaches.
(3)	They are useful features. It would be even better if I could see what was saved and the time when the save was made.
(4)	Because I can restore the environment even after being attacked, I feel more at ease and show less hesitation in entering commands.

(b) Negative Comments

Items	Comments
(2)	The difficulty level is high, but I think it is effective for learning. A one-week deadline, as in other course assignments, is not sufficient; I would prefer a period of about two weeks.
(3)	I didn't really feel that I was able to use it properly. I didn't have a sense that the system was actually restored to the saved point.
(4)	In practice, it is necessary to strengthen security before an attack occurs, but I couldn't learn that through this exercise.

group under the assumption that the function was enabled. Table 3 shows excerpts from the interview results with the function enabled group. Table 3-(a) summarizes positive opinions from the participants. They commented that they felt psychologically comfortable and were motivated to actively try other patterns. Items (1) and (2) show that participants felt less psychological pressure during the exercise. These comments indicate that participants who felt psychologically comfortable were motivated to try different approaches. Together with the quantitative results showing an increase in the number of executed commands, these findings suggest that the hesitation to execute commands was reduced.

Furthermore, items (3) and (4) suggest that the Trial-and-Error function was regarded as useful and could be enhanced by visualizing save information. Item (3) was commented because the screen in Figure 3-E did not change during the rewinding process (e.g., modifications to configuration files). In addition, the system allowed students to proceed with the exercise while checking the results of each command step by step, which supports gradual understanding and self-reflection. From the analysis of the number of command inputs and the interview results, it can be concluded that the challenge discussed was addressed through the use of the proposed system.

On the other hand, Table 3-(b) presents negative opinions obtained from the interview. Some participants mentioned that they did not clearly perceive the effect of the save/restore function. These comments suggest that beginners may need additional guidance or visual feedback to understand whether their save and restore operations were successfully performed. Furthermore, it was pointed out that the exercise mainly focused on reactive defense after an attack, and that the opportunity to learn preventive defense methods before an attack was limited. In response to this opinion, the learning experience could be improved by developing a curriculum on preventive defense based on this exercise, enabling students to learn how to strengthen servers before an attack occurs.

Table 4: Results of the Interview with the Trial-and-Error Function Disabled Group

Items	Comments
(1)	If an attack is allowed to succeed even once, it is undesirable from a security standpoint. I think I could respond to it more effectively if I were able to use the rewind function.
(2)	Since I cannot be completely sure that my actions are correct, if the Trial-and-Error function were available, I could go back each time and try several different patterns.
(3)	When doing the exercise in the usual way, my motivation gradually decreased, but I think I could maintain my motivation by using the Trial-and-Error function.
(4)	As practice, this system can be used to check the result of each individual command I execute, rather than reviewing the entire exercise.

Table 4 shows excerpts from the interview responses with the Trial-and-Error function disabled group. These responses indicate that participants recognized the benefits of the function. They noted that the ability to restore the exercise state would reduce hesitation during the exercise. Furthermore, item (2) shows that the participants mentioned that they could try multiple approaches more confidently if the function were enabled. These findings suggest that the Trial-and-Error function mitigates students' hesitation in cyber defense exercises.

5 Conclusion

In this study, we demonstrated that the developed cybersecurity exercise system with a trial-and-error function reduced students' hesitation to execute commands.

An evaluation experiment conducted at Kagawa University showed that participants using our system executed more commands than those without it, indicating reduced hesitation in command execution. Interview results also suggested that the system helped participants feel psychologically at ease and encouraged them to explore multiple defensive approaches. These findings demonstrate that the proposed system can promote active participation and deeper understanding in cyber defense exercises.

In future work, we plan to measure the contents of executed commands and file modifications to analyze how learners' approaches to attacks change when the Trial-and-Error function is enabled, in order to evaluate the learning effectiveness of the proposed system. In addition, we intend to perform multi-day exercises with more participants.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP25K17074.

References

[1] ISC2, "How the economy, skills gap and artificial intelligence are challenging the global cybersecurity workforce," Oct. 2025; https://cybergovernancealliance.org/wp-content/uploads/2024/01/ISC2_Cybersecurity_Workforce_Study_2023-1.pdf

[2] NRI SecureTechnologies, Ltd, NRI Secure Insight 2023, Oct. 2025;

https://www.nri-secure.co.jp/download/insight2023-report.

[3] G. Østby, B. Selebø, S. Kowalski, “Training the Trainers for Cybersecurity Exercises - Developing EXCON-teams”. In: Abbas Moallem (eds) Human Factors in Cybersecurity, AHFE (2023) International Conference, vol 91, 2023, pp.111-120, doi:10.54941/ahfe1003725.

[4] E. Seker and H.H. Ozbenli, “The Concept of Cyber Defence Exercises (cdx): Planning, Execution, Evaluation,” 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2018, pp.1–9.

[5] K. Kida, “Seminars to develop information systems specialists who are good at information security | Course introduction | National University 55 Engineering Faculty HP”, Oct, 2025; https://www.mirai-kougaku.jp/lesson/pages/97.php (in Japanese).

[6] Kagawa University, “Kagawa University Academic Affairs System Campus-Xs”, Oct, 2025; https://kyoumusyst.kagawa-u.ac.jp/campusweb/slbssbdr.do?value(risyunen)=2025&value(semekikn)=1&value(kougicd)=E5005150-1&value(crclumcd)=9999 (in Japanese).

[7] I. Takehara, M. Ishizuka, H. Kamei, K. Kida and K. Saisho, “Evaluation of Processing Time in Trial-and-Error Function of Security Exercise System for Security Beginners”, Proc. the 2023 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE’23), 2023, pp.871-876, doi:10.1109/CSCE60160.2023.00147.

[8] I. Takehara, Y. Kami, H. Kamei, K. Kida and K. Saisho, “Development of Score Measurement and Attack Functions in a Security Exercise System Enable Trial-and-Error”, Proc. 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2024, pp.206–211, doi: 10.1109/IIAI-AAI63651.2024.00048.

[9] H. J. Perkinson, “Learning from Our Mistakes: A Reinterpretation of Twentieth Century Educational Theory”, Praeger, 1984.

[10] M. Okamura, “Revisiting Donald Sch¨on’s “Reflection-In-Action”: from an Examination of his “Epistemology of Practice” ”, Annual Report of The Japanese Society for the Study on Teacher Education, vol.26, 2017, pp.64–74 (in Japanese).

[11] K. Chiken, A. Hazeyama and Y. Miyadera, “A Programming Learning Environment Focusing on Failure Knowledge”, The Journal of Institute of Electronics, Information and Communication Engineers, vol. J88-D-1, no.1, 2005, pp.66–75 (in Japanese).

[12] T. Hirashima and T. Horiguchi, “Error-Visualization for Learning from Mistakes”, Transactions of Japanese Society for Information and Systems in Education, vol. 21, no. 3, 2004, pp. 178–186, (in Japanese).

[13] T. Horiguchi and T. Hirashima, “Simulation-Based Learning Environment for Assisting Error-Correction Management of Error-Based Simulation Considering the Cause of Errors”, Transactions of the Japanese Society for Artificial Intelligence, vol. 17, no. 4, 2002, pp.462–472 (in Japanese).

[14] T. Shinohara, I. Imai, T. Tomoto, T. Horiguchi, A. Yamada, S. Yamamoto, Y. Hayashi and T. Hirashima, “Experimental Use of Error-based Simulation for Force on Moving Object in

Science Class at Junior High School", vol. J99-D, no.4, 2016, pp.439–451, (in Japanese).

[15] the European Union Agency for Cybersecurity: Cyber Europe— ENISA, Oct. 2025; https://www.enisa.europa.eu/topics/skills-and-competences-for-companies/cyber-europe.

[16] MINI Hardening Project "MINI Hardening Project (ZANSIN Project) - connpass", Oct. 2025; https://minihardening.connpass.com/ (in Japanese).

[17] Kawaguchi Sekkei, inc., "Microhardening".Oct. 2025; https://www.sec-k.co.jp/mh (in Japanese).

[18] Y. Kodera and K. Chinen, "Design and Implementation of a Cyber Security Training Rewinding Mechanism", Security Psychology & Trust (SPT) of IPSJ, vol. 2021-SPT-41, no. 14, 2021, pp. 1–6 (in Japanese).

[19] K. Shiota,Y. Taniguchi and N. Iguchi, "A Security Risk Learning System in Public Wireless LAN Based on Failure Experiences", IPSJ Journal, vol.65, no.3, 2024, pp.748–753, doi/10.20729/00233258 (in Japanese).

[20] S. Ohta, S. Yasuda, T. Yumura and Y. Takano, "Toward Next-Generation Cyber Exercise Environments", Multimedia, Distributed, Cooperative, and Mobile Symposium, IPSJ Symposium Series, vol. 2016, no. 1, 2016, pp.1776–1782 (in Japanese).

[21] F. Skopik and M. Leitner, "Preparing for National Cyber Crises Using Non-linear Cyber Exercises", 2021 18th International Conference on Privacy, Security and Trust (PST), 2021, pp. 1-5, doi: 10.1109/PST52912.2021.9647795.

[22] Y. Hoshino, K. Notomi, H. Nishimura and H. Shimeno, "A Development and Evaluation of Training Environment for System Administrator Based on Linux Server", IEEJ Transactions on Electronics, Information and Systems, vol. 136, no.7, 2016, pp.986–994, doi: 10.1541/ieejeiss.136.986 (in Japanese)

[23] K. A. Gharabi and J. Arulappan, "Repeated Simulation Experience on Self-Confidence, Critical Thinking, and Competence of Nurses and Nursing Students-An Integrative Review", Sage Open Nurs, vol. 6: l-8, 2020, pp.1-8 doi: 10.1177/2377960820927377.

[24] WordPress Foundation, "Blog Tool, Publishing Platform, and CMS - WordPress.org", Jan 2025, https://wordpress.org/.

[25] KVM, "KVM", Jan 2025, https://www.linux-kvm.org/page/Main_Page.

[26] HashiCorp, "Vagrant by HashiCorp", Jan. 2025, https://www.vagrantup.com/.

[27] Red Hat, "2.4. Storage Formats for Virtual Disk Images — Red Hat Product Documentation", Jan 2025, https://docs.redhat.com/en/documentation/red_hat_virtualization/4.0/html/technical_reference/qcow2.