Education and Information Technologies
https://doi.org/10.1007/s10639-019-09942-y

Supporting Cybersecurity Education and Training via LMS Integration: CyLMS

Razvan Beuran, Dat Tang, Zheyu Tan, Shinobu Hasegawa, Yasuo Tan, Yoichi Shinoda

Received: 19 March 2019 / Accepted: 7 June 2019

Abstract Cybersecurity education and training are being conducted on an ever-increasing scale, as most organizations need to improve their readiness in dealing with the more and more frequent cyberattacks. However, most systems used for such education and training purposes are built from scratch, are highly customized, and often proprietary. This is true especially for complex activities that include hands-on practice, such as Capture The Flag (CTF) competitions and realistic cyber range training. Moreover, the specificities of these platforms create an important overhead, both for instructors, who need to develop training content and learn how to use them, and also for trainees, who need to each time adjust to a different platform.

In this paper, we present our approach of integrating cybersecurity training activities, both for technical and awareness training, with Learning Management Systems (LMSs). In particular, our system—named CyLMS—provides integration from content point of view with most LMSs through the use of the SCORM format for packaging the training content. Moreover, additional CyLMS modules make possible a tighter integration with the Moodle LMS, a widely-used e-learning platform, for tasks such as automatic activity management and hands-on environment access. In this way, both instructors and trainees benefit from standard interfaces for checking the training content, answering questions, managing the results, etc. The paper includes an evaluation of CyLMS from a functionality, user and performance perspectives that demonstrates its applicability to actual training activities. While so far we have only used CyLMS in the cybersecurity context, the platform is sufficiently generic to be applied to other education activities, as a learning content management tool that facilitates training content creation and sharing.

Keywords Cybersecurity education · Learning content management · Learning management system · Hands-on training · Cyber range

Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

1 Introduction

Large-scale cyberattacks have occurred worldwide more and more frequently in recent years, and with ever greater consequences. Some of the most severe cases in the past five years include: (i) a security breach at Yahoo in 2014 that compromised the accounts of 1 billion users, and is considered the largest discovered breach in the history of Internet; (ii) a DDoS attack conducted in October 2016 on the Dyn DNS provider in the United States, with traffic exceeding 1.2 Tbps, the largest DDoS attack to date that resulted in the inaccessibility of several high-profile web sites; (iii) the WannaCry ransomware campaign in May 2017, which infected over 400,000 computers in 150 countries in the largest ransomware attack yet.

Cybersecurity education and training, especially via hands-on activities, are essential for making sure that such security incidents can be prevented and handled adequately—see [3] for a study on this topic. Various companies currently provide training platforms, and national governments also firmly support such activities. For instance, CYDER (Cyber Defense Exercise with Recurrence) [15] is a program coordinated by the National Institute of Information and Communications Technology in Japan that provides regular hands-on training to IT personnel of national and local government organizations. The Hardening Project [27] is a security contest organized by the Web Application Security Forum in Japan, in which teams of security experts and IT professionals compete with each other in terms of the service level they can provide for a realistic e-commerce virtual company. Internationally, the SANS Institute provides as paid programs both live courses and online training via a set of interactive learning scenarios that should help professionals to develop and master real-world cybersecurity skills [19].

Most cybersecurity education and training programs employ custom platforms for presenting content to trainees, which are sometimes built from scratch and often proprietary. This means that trainees have to adjust to each platform every time they take part in a different training activity. In addition, this situation creates a large overhead for educators, who need to learn how to use new interfaces as they develop training content, manage trainees history and results, and so on. For some training programs, such as Capture The Flag (CTF) competitions, the relatively standard form in which they are conducted made possible the development of open-source platforms, but even in this relatively narrow field there are still many alternatives, none of them very mature [13].

Our goal of is to make possible wide-scale education and training programs that reach young people in universities, colleges and even high schools; this is in contrast with existing training programs, which typically only target security professionals in organizations, companies or in the military.

In this context, we have designed and implemented CyLMS (Cybersecurity Training Support for LMS), an open-source set of tools that integrate cybersecurity training features with existing LMS software in order to facilitate security-related education and training activities. This is made possible through the use of a text-based representation for the training content that is automatically converted to the de facto standard format for LMS named SCORM (Shareable Content Object Reference Model) [1]. A tighter integration is also available, in particular with the Moodle LMS—which is perhaps the most widely-used learning platform—for features such as managing automatically the learning content, and facilitating trainees' access to the associated training environment.

The LMS integration functionality provided by CyLMS makes it possible for educators to employ systems they are already used to for managing and conducting training activities, following trainee progress, and so on. Moreover, as CyLMS is open-source software and was released via GitHub [7], new features and extensions can be easily added to it. In case an open-source LMS such as Moodle is used, functionality can be further extended through a deeper integration with that LMS. The functionality that CyLMS provides makes it easier for learners as well to take part in training activities, as they use the same interface and principles shared with other courses.

The main contributions of the present paper are:

- Introduce the motivation behind our work on CyLMS and the overall architecture of the system (Section 2).
- Present the design and implementation of the core module of CyLMS, which converts training content descriptions to SCORM format (Section 3).
- Discuss the other modules of CyLMS that provide more advanced integration features, such as automatic content management, training environment integration, and interactive functionality (Section 4).
- Conduct an evaluation of CyLMS from functionality, user and performance perspectives (Section 5).

Although CyLMS was only used so far in the context of cybersecurity education and training, we note that the platform is sufficiently generic to be applied to other education activities, as a general-purpose learning content management tool that facilitates training content creation and sharing.

The remainder of this paper is organized as follows. In Section 2 we discuss general cybersecurity training principles, as well as the overall design of CyLMS and the advantages it offers. Then, in Sections 3 and 4, we present details about the training content representation, and the manner in which the tighter integration with LMS is achieved. Next, in Section 5, we discuss the functionality and performance characteristics of CyLMS. The paper continues with related work, and ends with conclusions, acknowledgments and references.

2 Motivation and Overview

In this section we briefly outline the main characteristics of cybersecurity training, then present an overview of our solution for addressing the identified issues.

2.1 Cybersecurity Training

The main goal of cybersecurity education and training is to enable learners to have the correct *behavior* when encountering particular security-related situations. Therefore, such activities focus not only on providing theoretical knowledge, but also on cultivating the necessary skills and abilities. Consequently, experiential learning through hands-on interactive practice is essential for an effective education process. Repetitive training is also an important characteristic, which makes it desirable to have automation features that facilitate such kind of training.

The two components that are required in order to perform hands-on cybersecurity training are as follows:

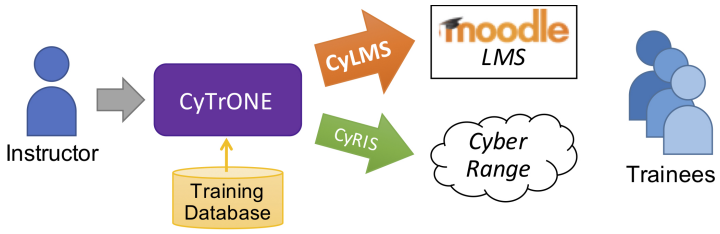

Fig. 1 Architecture of the CyTrONE training framework.

1. *Training content*: An explanation of the training activity, including steps that need to be carried out, questions to be answered, and so on
2. *Training environment*: Realistic resources and network setup that learners can access in order to perform all the procedures required for answering the questions they have received; also called "cyber range"

The integrated cybersecurity training framework named CyTrONE was conceived as a complete solution for facilitating large-scale security training by automating the management of these two components [5]. Instructors interact with CyTrONE, which employs additional tools to accomplish its functions, as shown in Figure 1. The resources necessary for education and training activities are made available in the form of a training database.

The main motivation behind our research was to create a module that, upon instructor's request, takes training content from the CyTrONE database and imports into the Moodle LMS. Consequently, we designed CyLMS so that it can be integrated into the workflow of CyTrONE to provide all the LMS-related functionality needed for cybersecurity training activities in terms of training content representation, conversion to LMS format and import into the LMS, as well as facilitating trainees' access to the cyber range.

An additional module, the cyber range instantiation system CyRIS, is used to create on-demand a network environment with the required characteristics associated to a certain training content [4]. CyRIS gets a cyber range description as input for setting up virtual machines (VMs), both with basic functionality—such as necessary accounts, tools, files and program execution—and also with security-related properties/content via features such as firewall configuration, malware emulation, attack emulation and traffic capture. Once the cyber range is set up, the trainees can consult the instructions provided via the LMS, investigate the cyber range according to those instructions, then submit their answers also via the LMS.

2.2 Training System Requirements

Although the training environment or resources may not be needed for those forms of education or training that do not imply hands-on practice, the training content is of utmost importance in all circumstances and for all kinds of learning. This is because both the training content itself and the way in which it is presented to learners are key in making sure that the training activity is effective.

First of all, training content that is clearly expressed and well adapted to a given group of trainees ensures that their learning process is efficient and smooth.

Therefore, we claim that any cybersecurity training system must provide enough *flexibility* for educators to be able to achieve their goals at any given moment. This is because there is no one-size-fits-all solution for training content. Thus, the way in which the activity is explained and the questions are expressed may need to be adjusted to the knowledge and skill background of a certain group of learners. Moreover, it is also possible to imagine different training content for the same training environment, for instance, for beginners versus for advanced users, guided versus free learning, attack versus defense training, and so on. Hence, one key requirement is to give educators the freedom they need in order to create and adapt training content as they see fitted for their students.

Regarding the manner in which the training content is presented to learners, one should make sure that there is no steep learning curve or hindrance for simply accessing the training content, an action that is unrelated to the training itself and should be made as straightforward as possible. A *familiar* user interface is the easiest way to ensure that user experience issues are kept to a minimum. While there may be multiple opinions as to what is the most appropriate user interface for cybersecurity training, we believe that Learning Management Systems, which are already widely used in education, are a suitable solution for most training activities, either directly or via small modifications, as we shall detail later.

To summarize this discussion, we propose the following three requirements for cybersecurity education and training platforms:

1. Provide an easy-to-use (if possible, familiar) user interface both for educators and learners for managing, delivering and interacting with courses.
2. Facilitate content creation, so that educators can easily create and modify content depending on a given audience and specific objectives.
3. Support hands-on activities, so that learners can practice their skills in realistic conditions following the experiential learning paradigm.

2.3 Our Approach

CyLMS was designed based on the above requirements, thus making it possible to meet our goal of supporting cybersecurity education and training. To address Requirement #1, we decided to integrate our system with LMS software, which intrinsically solves this issue, since at least the LMS concepts should be familiar to most educators and learners. As for the other requirements, CyLMS uses a two-pronged approach to address them, as follows:

- Employ as input format a text-based representation of the training content—that is easy to modify and share—and convert it automatically to the standard SCORM format, which can be imported into most modern LMS platforms → Requirement #2.
- Deploy a set of tools that provide a tighter integration with the LMS software, so that tasks such as training management, cyber range access, and interactivity can be automated/simplified → Requirement #3.

An overview of the CyLMS functionality is shown in Figure 2. In a first stage, the input file containing the training content representation is converted to a SCORM package, a process that will be discussed in more detail in Section 3. The text-based representation of the training content allows educators to focus on the

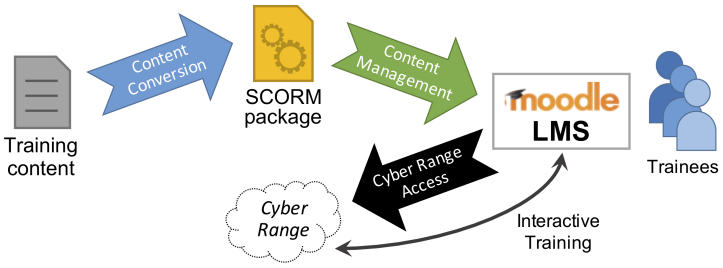

Fig. 2 Overview of the functionality provided by CyLMS.

actual content itself, and mostly ignore its formatting, similar to what LaTeX does for typesetting. The conversion to SCORM format makes it possible to import the training content into most LMS software, either manually or automatically.

In a second stage, the additional integration functionality with the LMS provided by CyLMS—in particular regarding automated content management and access to the training environment—ensures an improved user experience (see Section 4). Although not mandatory for conducting cybersecurity training via CyLMS, such features further facilitate the training process by reducing the management overhead for educators and improving the user experience for trainees.

In the following sections we shall provide details about the manner in which each of the aforementioned features of CyLMS were implemented.

3 Content Representation

One of the most important aspects related to facilitating cybersecurity education and training is a flexible content representation. LMS platforms use custom formats for representing content, hence a more generic format is needed if we consider the need to make it possible to use multiple LMS applications without restrictions.

Non-platform-specific representations can be either in text or in binary formats. The existing text-format representations for LMS content have various limitations on the type of content that can be represented, and their support in various LMSs is uneven, as it will be discussed in the overview of related work (Section 6). On the other hand, binary formats may be better supported, but it is more difficult to produce files in that format. SCORM (Shareable Content Object Reference Model) is a binary format that was designed specifically in order to address interoperability, portability and content reusability issues [1]. The main advantage of SCORM is that it became a de facto standard, and is well supported for content import by most modern LMS platforms.

Due to such considerations, we decided on a hybrid approach for our system. Thus, the input of CyLMS is an original text-based format using the YAML syntax, an easy to manage file format that is also extensible by nature if new features need to be added [9]. Then, a converter tool that we implemented produces an equivalent SCORM package by using a package template to control the presentation style of the training content. In this manner we are able to combine the flexibility of the input YAML representation with the versatility of the SCORM format to facilitate training content representation.

In what follows, we shall first discuss the YAML representation used by CyLMS, and then detail the implementation of the YAML to SCORM conversion module.

3.1 Input Format

We have selected the text-based YAML format for representing training content in CyLMS because of the following advantages it has:

- Easy to view and modify via any text editor
- Both human and machine readable
- Flexible representation form, allowing inclusion of HTML code (e.g., for changing style or adding figures)
- Straightforward versioning and difference checking
- Small size, easy to archive and transfer in order to share the training content

For the purpose of representing training content using YAML, we have introduced a series of custom keys in a hierarchical structure. At the top level, the key `training` is used to indicate that the YAML file represents training content. Each training activity within the file includes an identifier, a title, an overview and a list of questions, each represented by corresponding key labels (`id` for the activity identifier, and so on). In its turn, each question included in a training activity contains a body, optional choices, the correct answer and optional hints, again identified via predefined key labels (`body` for the question body, etc.).

In Figure 3 we show an example of a training content representation file for CyLMS. The file includes the following:

- A training activity with the id "Example" for which a title, an overview, and two questions are provided.
- The first question, with the internal id "EX-1", is a typical question for which the learner needs to fill in the answer; three hints are provided for this question, ranging from somewhat vague to very specific.
- The second question, with the internal id "EX-2", is a multiple-choice question for which the trainee needs to select the correct answer from the four specified alternatives; this question as well includes three hints.

This example content representation file is included with the CyLMS code to illustrate its features, and demonstrates that content created using this format is both easy to read and manage. Although this example is not usable for actual training activities, we note that more realistic training content is included with the cybersecurity training framework that we introduced in Section 2.1.

3.2 Conversion to SCORM Package

The main role of the conversion module of CyLMS is to "translate" a content representation given in the YAML format discussed in Section 3.1 to a standard SCORM content package. Such packages are self-contained ZIP archives that contain all the files needed to deliver the learning content via an LMS. In particular, any SCORM package includes: (i) an XML manifest file; (ii) all the schema/definition files referenced by the manifest file; (iii) all the resource files used by the SCORM package and its learning activities.

```
---
- training:
  - id: Example
    title: Example training questions
    overview: |
    <p>These are two example questions that demonstrate how to define training
content for CyLMS. Please check how the questions are displayed after they are
converted to a SCORM package and uploaded to Moodle.</p>

    questions:
    - id: EX-1
      body: What is the name of the Linux distribution used by this server?
Indicate only the name, without version or architecture, e.g., CentOS.
      answer: Ubuntu
      hints:
      - The directory <code>/etc/</code> contains various files with information
regarding a Linux distribution.
      - One of the most relevant files has a name ending in "release".
      - <code>$ cat /etc/*-release</code>

    - id: EX-2
      body: What is the name of the account you are logged in as?
      choices: root, admin, guest01, trainee01
      answer: trainee01
      hints:
      - The account name is typically the same with the name of the user's home
directory.
      - "Have you ever asked yourself: Who am I?"
      - <code>$ whoami</code>
```

Fig. 3 Sample training content representation that includes two questions, one of fill-in type and one of multiple-choice type.

Note that we employ a SCORM package template for the purpose of the conversion, and the content description from the input YAML file is used to fill-in accordingly information into the template. This approach simplifies the conversion process, as it is not necessary to generate all the files in the SCORM package structure, and also gives more control to educators, who can modify the template as they find suited, for instance, to change the presentation style, add logos, etc.

The SCORM package template needs to be prepared in advance of running CyLMS for the first time. For users' convenience, we included a script that generates a custom template based on a publicly available resource, the "SCORM 2004 Basic Run-time Package" provided on the SCORM.com web site [17]. Our script downloads the sample package, then customizes it to create the template by removing unnecessary files and adding some specific CyLMS files. Educators can either employ the resulting template as such, or adapt it as needed, for example, to change content appearance and style via HTML/CSS mechanisms. It is also possible to use a completely different template, as long as the CyLMS-specific files are included, as explained next.

For a given input YAML file, the format conversion involves the following steps (see Figure 4 for an overview):

1. Check the existence of the top-level key `training` to validate the file type; also check the presence of the training section keys `id`, etc.

Fig. 4 Overview of the conversion process of YAML-based training content representation to SCORM package.

2. Copy the SCORM template prepared in advance to a new directory.
3. For each item in the questions list (if any) do:
   (a) Check the presence of the question keys `body`, etc.
   (b) Add question data to the file "questions.js" in the new SCORM directory, either as a fill-in or as a multiple-choice question.
4. Add training data to the following files in the new SCORM directory: "assessmenttemplate.html" for the training title, overview and level, and "imsmanifest.xml" for the training id.
5. Create a ZIP archive of the new SCORM directory, thus producing the SCORM content package that represents the output of the conversion module.

The conversion from YAML format to SCORM package described in this section was fully implemented based on the algorithm presented above by using the Python programming language (the corresponding file in the released source code is called `cnt2lms.py`).

Once the SCORM package is generated by CyLMS, it can be manually imported into the LMS of choice for conducting the training. For the Moodle LMS, however, we provide deeper integration mechanisms that simplify training management, as it will be described in the next section.

4 LMS Integration

Through our system's integration with LMS software, and in particular with the Moodle LMS, we aim to further simplify the way in which cybersecurity training is conducted. The current implementation covers the following main aspects:

- Provide the ability to automatically add/delete training content into the LMS
- Facilitate trainees' access to the cyber range via links included in the LMS activity page
- Make it possible to conduct interactive training activities directly via the LMS interface

4.1 Content Management

Although it is possible to import SCORM packages—such as the ones generated by our system—manually into an LMS via its web interface, to facilitate content management we enabled the import/removal of SCORM packages without any user intervention. While this CyLMS functionality is currently limited to the Moodle LMS, we believe that it could be extended to other LMS software if needed.

The following steps are performed when adding content to the Moodle LMS (see Figure 5 for an overview):

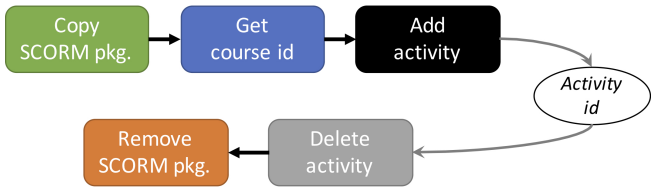

Fig. 5 Overview of the content management process in CyLMS.

1. Copy the generated SCORM package to the Moodle host into the appropriate repository directory.
2. Get the internal course id for the Moodle course into which content should be imported.
3. Add the SCORM package as a new activity in the given course and section, properly configuring its name and properties (SCORM file path, update frequency, etc.).
4. Extract the internal activity id for the newly-created activity by parsing the previous operation output, and return it to the caller.

A configuration file is used to provide to CyLMS the necessary parameters for the above operations, such as the LMS host and repository used at step 1 in the algorithm, and the name of the course and section identifier needed at steps 2 and 3, respectively. The related functionality was fully implemented in Python (the corresponding file in the released source code is `cfg_mgmt.py`).

In order to interface with the Moodle LMS, CyLMS employs the open-source software named `moosh` [14]. Moosh stands for "Moodle Shell" and makes it possible to perform many Moodle management tasks via the command line. For instance, we use the `moosh` command `course-list` to determine the course id for a given course title, and the command `activity-add` in order to add a SCORM package as a Moodle activity. Note that we had to do a minor patch of the `moosh` source code in order to display the internal activity id for newly-created activities, since this id is necessary when deleting those activities (see below).

The opposite operation, deleting content, is implemented in CyLMS via the following steps (see lower half of Figure 5):

1. Delete the Moodle activity with the id provided by the caller (typically stored when the said activity is added into the LMS).
2. Remove the SCORM package file associated with the deleted activity from the repository directory on the Moodle host.

For the content removal functionality as well we rely on the `moosh` tool. In particular, we use the command `activity-delete` at step 1 above in order to actually remove the activity from the Moodle database. Information about the Moodle host and repository needed at step 2 is retrieved from the aforementioned configuration file options.

Once a SCORM package is imported into Moodle, it will be displayed as shown in Figure 6. For this example we used as CyLMS input the sample training content representation presented in Figure 3. Note how the training overview is displayed at the top of the page, followed by the questions. For illustration purposes, we have already revealed the hints for "Question 1", but hints are initially hidden from trainees, as is the case for "Question 2" in the figure.

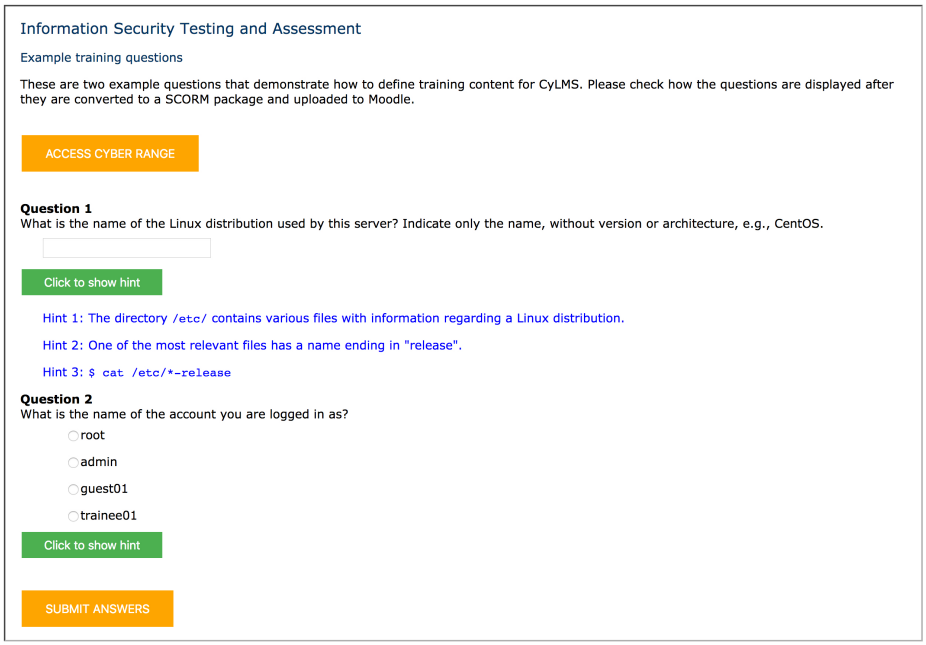

Fig. 6 Screenshot of an activity added to Moodle by using as CyLMS input the sample training content shown in Figure 3.

The LMS integration functionality from content management point of view presented in this section was fully implemented using the Python programming language (the corresponding file in the released source code is `lms_mgmt.py`).

4.2 Cyber Range Access

Most modern cybersecurity training programs use cyber ranges as training environments in which the participants can apply their skills to solve realistic hands-on problems. If an LMS is used as the primary interface for trainees to receive instructions, consult questions and submit answers, the trainees could also be given access to the cyber range via the LMS interface, thus improving their user experience. The three components used in order to enable this functionality are:

- A button on the main page of the SCORM package (in our case, inserted via the HTML `button` tag into the file "assessmenttemplate.html")
- A function associated with this button that will open a new HTML page (in our implementation, by using the `window.open()` function in JavaScript)
- A daemon process that will mediate the access to the particular cyber range that is to be used in a given training session (see below for details)

The screenshot in Figure 6 includes below the training overview the button `ACCESS CYBER RANGE` that was created using the mechanisms presented above. The code that needs to be inserted into the SCORM package file for displaying this button and opening a window when the button is clicked is minimal. The

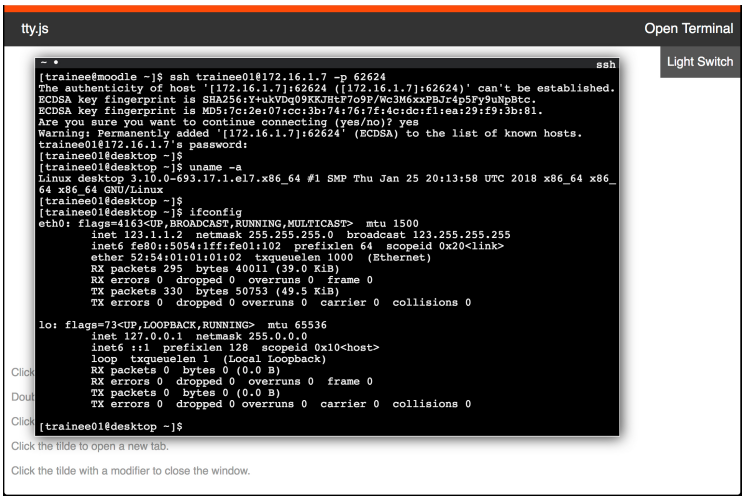

Fig. 7 Screenshot of the terminal window opened via the LMS with support from `tty.js`; trainees can use the terminal to log in to the cyber range and issue various shell commands.

third component, the daemon process that mediates access to the cyber range was implemented differently depending on the access method.

Terminal Access For basic investigations in the cyber range, command-line access via an SSH terminal is the most robust and versatile alternative. To support such access via the LMS interface we have employed the `tty.js` terminal emulator for web browsers [12]. After trainees access the `tty.js` daemon web page by clicking on the connection button, they are able to open a typical terminal window, in which they can type commands for logging in to the cyber range and operating inside it (see Figure 7). An optional Python program collects the information necessary to login into the cyber range, and uses it to generate shell scripts that trainees can execute in order to log in. The collected information originates now from the creation notification provided by the cyber range instantiation system CyRIS mentioned in Section 2.1, but even if the training environment is prepared in a different manner, such a script could be created.

Remote Desktop Access For training related to web vulnerabilities, one needs to access targets in the cyber range via a web browser. Trainees can accomplish this task by remotely viewing the desktop interface of the cyber range machines. This access method takes more bandwidth than terminal access, hence the user experience may be subpar on slow connections, but it is the most straightforward way[1]. For remote desktop functionality we have employed `noVNC`, which is a VNC client that can be used from within web browsers [16]. One `noVNC` instance is associated with one cyber range machine, hence multiple such instances are required in order to enable remote desktop access to multiple machines. A script is used to collect

[1] The technique of port forwarding also makes it possible to access remote web servers in a closed environment, such as a cyber range, via one's own PC browser, hence serving the same purpose and providing a better performance, albeit with potential security risks.

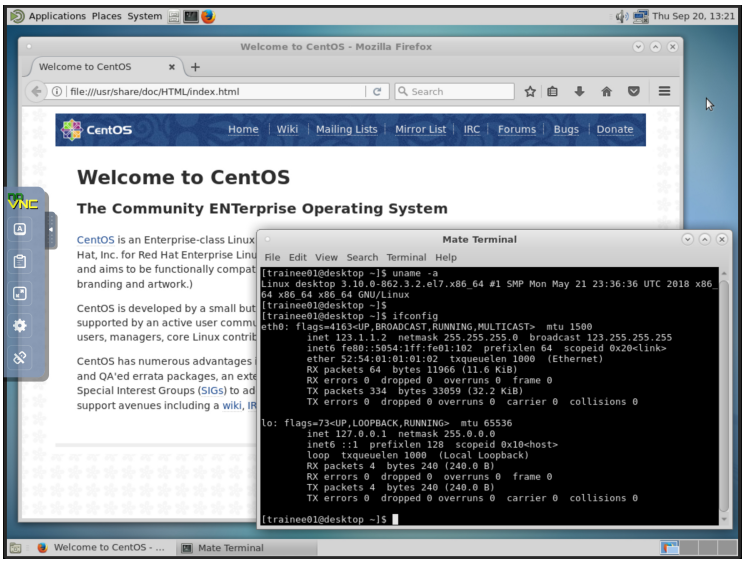

Fig. 8 Screenshot of the remote desktop window opened via the LMS with support from `noVNC`; trainees can use this interface to directly access the GUI environment of the VMs.

login information regarding the cyber range, and allow the trainees to use the VNC client to connect to the appropriate cyber range machine (see Figure 8).

Method Comparison The reason why we provide two methods for accessing the cyber range is that each of them has its advantages and disadvantages, which makes it necessary for educators to choose the most appropriate method for certain given circumstances (see Table 1). The main characteristics of each access method are:

- Terminal access via `tty.js` is lightweight, as it uses text-based information representation for TTY-mode access, hence it has low bandwidth requirements, and a low expected interaction latency. Moreover, this method only requires one instance of the daemon process for all trainees, since actual access is done inside the terminal via the SSH command. Additionally, copy-paste functionality is available for interacting with the terminal.
- Remote desktop access via `noVNC` is on the other hand more resource heavy, as it uses graphical information representation to support both GUI- and TTY-mode access. Furthermore, one instance of the daemon process is required for each VM to which access is required. Copy-paste functionality is mediated via a special "clipboard panel", hence it is more cumbersome (and unavailable for machines without a GUI environment, which are accessed in TTY mode).

4.3 Interactive Training

Interactivity is an important characteristic in many learning theories, and we wanted to also bring such functionality to cybersecurity training. For this purpose we have modified a SCORM package so as to present interactive training

Table 1 Comparison of the Cyber Range Access Methods

Main Characteristics	`tty.js`	`noVNC`
Information representation	Text	Graphical
Supported access mode	TTY	GUI & TTY
Bandwidth requirements	Low	High
Expected interaction latency	Low	High
Daemon process granularity	System level	VM level
Copy-paste functionality	Direct support	Clipboard panel

sessions that a user can access via the LMS interface. Our research on interactive cybersecurity training is discussed in detail in [23], and we only summarize here the main features.

The interface design for interactive training that we did was guided by the recommendations in [28] for adding interactivity to web-based learning: display important information on the top and in the center of the interface; use buttons with bright colors to get attention to them; and provide orientation clues for the training sessions. A screenshot of our interactive training interface as displayed in Moodle is shown in Figure 9; the main elements are, from top to bottom:

- Brief overview of the training activity displayed at the top of the page
- Pull-down list of security vulnerabilities the trainees can choose from as topics for the training session
- Buttons to create and restart the cyber range, as well as to request that an on-demand attack is conducted
- Progress bar to indicate the advancement of the cyber range creation process (which can take several minutes)
- Short description of the chosen vulnerability, updated dynamically in accordance with the user selection
- Detailed instructions regarding the steps the trainee should carry out as part of the training
- Button for connecting to the cyber range via the terminal, so as to manage and investigate the target machine before/after the attack

When an interactive training session ends, a message is displayed in the LMS window (not shown in Figure 9). The training outcome is considered to be a *success* in case the trainee managed to secure the target machine before the attack, and a *failure* in case the target machine fell victim to the attack despite the trainee's efforts. The activity can be repeated as many times as necessary.

The interactive features discussed in this section are implemented by simply adding several PHP scripts and related resources directly into a SCORM package. The PHP scripts are used to retrieve the vulnerability description, control the cyber range (creation, restart, attack request), show the creation progress, check the attack outcome, and so on. Note that the interactive training functionality of CyLMS is not yet integrated into the GitHub release, but we are exploring ways to make it available in the near future. Although the training activity described above is just an example, we believe that this approach has great potential for being developed into a more generic and powerful feature of CyLMS.

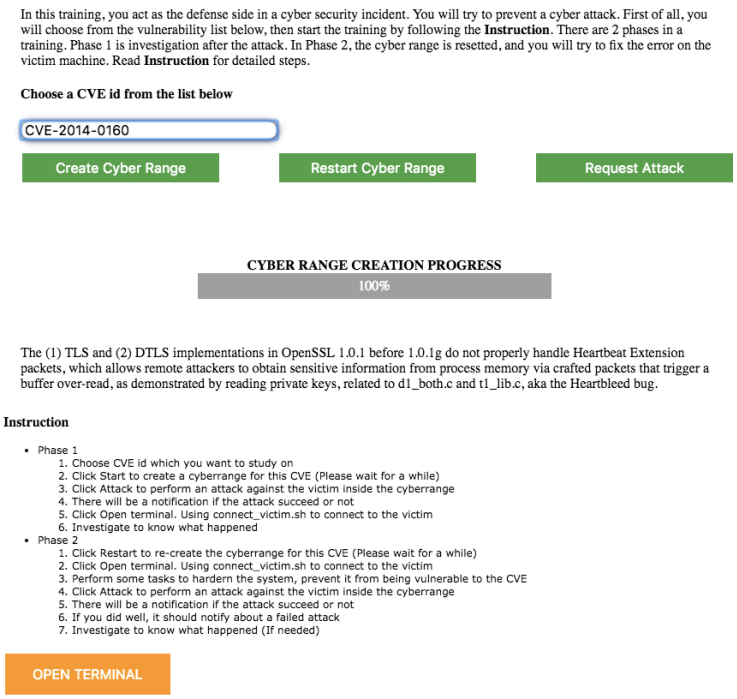

Fig. 9 Screenshot of the interactive training interface as displayed in the Moodle LMS [23].

5 Evaluation

In what follows, we first validate the functionality provided by CyLMS, then we present a user evaluation of its features, and finally we discuss its performance characteristics.

5.1 Functionality Validation

Requirement Evaluation Based on the requirements that we have formulated in Section 2.2, CyLMS provides LMS integration features that facilitate cybersecurity education and training. As CyLMS augments the functionality of the LMS, the combined platform provides additional capabilities that are not otherwise available (see Table 2). First of all, the combined setup "LMS + CyLMS" inherits the familiar course management and delivery functions of the LMS, as well as its online learning features, thus addressing Requirement #1. Second, CyLMS makes possible easy content creation via the use of YAML-based input, and also automates the process related to LMS activity management, hence meeting Requirement #2. Last but not least, hands-on training is supported through a simplified access to the training environment, and interactive practice sessions are made possible by integrating training environment management features into the LMS interface, thus addressing Requirement #3. This analysis demonstrates that our system, in-

Table 2 CyLMS Functionality Evaluation

Main Characteristics	LMS	LMS + CyLMS
Course management & delivery	◯	◯
Online learning features	◯	◯
Easy content creation		◯
Automated activity management		◯
Hands-on training support		◯
Interactive practice sessions		◯

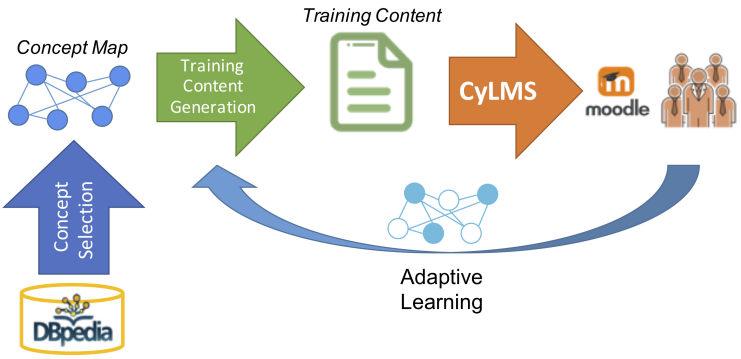

Fig. 10 Overview of our approach to security awareness training.

tegrated with an LMS platform, provides a unified interface that enables us to fulfill the most important characteristics of cybersecurity education and training.

Implementation Validation The CyLMS implementation was repeatedly validated according to the functionality of each of its submodules. For instance, the conversion to SCORM package detailed in Section 3.2 was validated by asking more than 10 users to create training content in the YAML format that we defined, both in English and Japanese, and making sure each time that the generated SCORM packages were correct. Similarly, the LMS content management features discussed in Section 4.1 were fully validated by repeatedly adding and deleting to a Moodle setup activities based on SCORM packages that were generated automatically as described above, and verifying each time that those activities were correctly displayed in the LMS.

Other Applications In addition to using CyLMS together with the CyTrONE framework presented in Section 2.1, we were also able to use it for another application in the field of *security awareness training*, thus further demonstrating its usefulness. An overview of our approach in this case is provided in Figure 10, which includes the processing stages detailed below (for more technical details about items 1 and 2, please see our previous paper [22]):

1. *Concept selection*: Use a publicly available Linked Open Data (LOD) dataset, in particular DBpedia [8], to retrieve all the concepts that are related to a

given *keyword*, which represents the main topic of that training activity. The Page Rank algorithm proposed in connection with hypertextual web search [6] is used to determine the degree of relevance of the retrieved topics with respect to the input keyword, for further filtering/ordering purposes.
2. *Training content generation*: Generate quizzes made of multiple-choice questions—a format often used in awareness training courses—for instance, by asking questions of the form *"What is C?"*, where C represents any particular concept that must be learned. For choices we use a template of the form *"... is D."*, where D denotes a concept definition obtained again from DBpedia, with the right definition being used in addition to a number of incorrect ones.
3. *Training content import*: Employ CyLMS to automatically import the training content into the Moodle LMS, which is then used to conduct the training activity, so that the entire training process is automated.
4. *Adaptive learning*: Optionally one could employ a *learner model* to estimate the ability of each particular learner, and an *instruction model* to adaptively select the most suitable questions for that learner depending on his/her model data, thus improving the training effectiveness.

The discussion above illustrated how the use of CyLMS can simplify the design of a training system, by letting developers focus primarily on the training content and overall process. This is because CyLMS enables them to use LMS software as a front-end interface in an effortless manner, even for a relatively complex system as the one we presented. Although we have so far applied CyLMS only to the context of cybersecurity education and training, many of its capabilities are more generic than that. For instance, many fields could benefit from the easy content creation made possible by CyLMS, or the activity management automation features. Furthermore, in case a network environment is associated with a given learning activity, then the hands-on facilitation mechanisms of CyLMS could be used to mediate access to that environment. Therefore, we believe that our contribution encompasses a larger area than just security education.

5.2 User Evaluation

Content Creation The first type of users of CyLMS are instructors, and their first mission is to develop training content. CyLMS was used by such instructors within the framework of several projects, the most important of which are listed below:

- Create training content based on the US NIST Technical Guide to Information Security Testing and Assessment [20]. We were thus able to validate that CyLMS content representation can cover all the three classes of techniques discussed in the mentioned guide: review, target identification and analysis, and target vulnerability validation.
- Develop training content related to the security courses taught at the Tokyo Metropolitan College of Industrial Technology in Japan. Senior students and professors collaborated to create content on topics ranging from web application vulnerability assessment to penetration testing, thus further validating the characteristics of our framework.
- Create training content based on the subjects of "Registered Information Security Specialist Examination", an official certification exam administered twice

a year by the Information-technology Promotion Agency (IPA) in Japan. The content, developed in partnership with a commercial company, demonstrated the applicability of CyLMS to real-life training scenarios.

Discussions with the content creators who participated to the above activities and were not directly involved in the development of CyLMS helped shape the current content specification format, and their feedback confirmed its usefulness for various education purposes. These discussions will also influence the future development of CyLMS, such as a request to be able to separate long training activities into "sections", each with its own page in the SCORM package, a feature that we are considering for the next release of the system.

Training Activity While the three projects mentioned above highlight the applicability of CyLMS to diverse circumstances in terms of content creation, we also want to mention the use of CyLMS for an actual training session that was conducted in March 2019 as part of the course "Literacy in Information Security Management" given at our institution. A total of 26 students attended this course, and all of them took part in the said training. The activity used content adapted from the IPA exam-based training content mentioned above, and the full CyTrONE framework presented in Section 2.1 was employed to manage the training. The activity took place successfully, and the training scores handled through the Moodle LMS were part of the final student grades for that course, thus validating the use of CyLMS to conduct training for actual university courses.

This training activity allowed us to also get the perspective of the second category of CyLMS users, the trainees, regarding our system. The training content itself was evaluated by students very positively, and it was decided that this training will take place annually from now on. Furthermore, interviews with the attending students revealed that their training experience via CyLMS was smooth and straightforward. The only problem encountered by two of them was that question hints were not correctly displayed in a certain version of the Microsoft Edge browser running on Surface tablets; we are currently working on fixing this bug before the next software release.

5.3 Performance Evaluation

Given that CyLMS provides several additional features over a standard LMS, we conducted a performance evaluation of its implementation, so as to make sure that its execution burden and processing time have no negative influences on the web server load and users' experience.

All CyLMS operations are straightforward in terms of required processing power, being mainly related to basic text processing, network transfer and file operations. To determine its performance characteristics, we have conducted several experiments with CyLMS using as input the file that was shown in Figure 3; this file includes 2 questions, having 24 lines and 1090 bytes. Note that we have conducted similar experiments with more realistic training content as well, including input files having 10 questions per set, 90 lines and up to 7,800 bytes; the results we obtained are comparable, therefore we omit their detailed presentation. All experiments were performed on a Dell R430 server with dual 14-core

Table 3 CyLMS Performance Evaluation

CyLMS Operation	Processing Time [s]
Conversion to SCORM package	0.1
Moodle activity creation	1.7
Moodle activity removal	1.5

Intel Xeon E5-2660 v4 CPU at 2.00 GHz and 128 GB memory, which we typically use in cybersecurity training activities.

In Table 3 we provide several indicative performance evaluation results, which are averages over 5 experiment runs for each operation. First of all, our measurements show that the conversion operation, which is the main text processing step, only takes about 0.1 s to complete, hence it is very fast.

The Moodle activity creation operation is composed of two main steps: copying the SCORM package file remotely via SCP, and importing the activity into Moodle via remote command execution over SSH. Consequently, the measured duration of this operation also includes authentication and network communication time. In our experiments, Moodle was run as a virtual machine on the Dell server, and the average RTT between the host and the Moodle VM—as shown by the PING command—was of 0.343 ms. In such circumstances, we determined that activity creation takes around 1.7 s, which we consider to be a reasonable time given the remote operations involved. For reference, simply executing the command `ls` over SSH in the same environment took about 0.3 s.

As for the Moodle activity removal operation, it also has two components: removing the activity from Moodle, and deleting the SCORM package file, both executed over SSH. The results we obtained are similar to those for activity creation, with an average time of about 1.5 s, which is still a reasonably short time. Our assumption as to why the removal takes a shorter time compared to activity creation, even though it is composed of two steps as well, is that for activity creation the SCORM package needs to be transferred via SCP, whereas there is no such file transfer operation for removal.

Regarding the other features of CyLMS, such as facilitating the access to the cyber range, although we have not conducted any objective evaluation, we mention that the time it takes to display the terminal or VNC access web pages is comparable to any other browser operation.

6 Related Work

Our approach to facilitating cybersecurity education and training is based on two main pillars: supporting training content creation tasks via a user-friendly input format, and providing integration capabilities with LMSs as means to leverage the functionality of existing e-learning platforms. The following subsections detail activities related to these aspects.

6.1 Content Representation

We consider that content representation is a major issue regarding cybersecurity training, and one which is often ignored or given a lower priority by researchers.

Text-based formats are the easiest to edit directly, and, in the context of e-learning platforms, AIKEN [24] and GIFT [25] are the most basic of them. AIKEN is human readable, however is restricted to multiple-choice questions, hence it lacks flexibility. GIFT is a lot more versatile, and can be even exported from Moodle, but is a lot less easy to read. The Moodle XML format [26] is also a versatile quiz format, but, being based on the XML standard, is mainly intended as a machine-readable representation for import/export tasks.

Our use of YAML as a basic format for training content representation in CyLMS combines the advantages of human-readable formats, such as AIKEN or GIFT, with the extensibility and structure of a machine-readable format, hence providing the best features of both worlds. Nevertheless, using directly such a format would require creating appropriate plugins for the target LMSs, and it would give educators no control over the manner in which the content is actually displayed. This is why, in a second stage we make use of the binary format SCORM.

Even tough every LMS platform, such as Moodle or Blackboard, has a native format for content representation, most of them also support SCORM as import format, and this is the reason that motivated our decision to use it in CyLMS. The SCORM format [1] is being maintained by the Advanced Distributed Learning Initiative program, and is being widely used already in practice, thus having become a de facto standard for e-learning, which is a major advantage.

6.2 LMS Integration

As explained already, `moosh` [14] is a tool that provides a command-line interface for managing Moodle, and which we use in order to achieve some of the LMS integration functionality presented in this paper. Nevertheless, there are also efforts to create more generic LMS integration mechanisms. Thus, GLUE! [2] is an architecture for achieving an integration of various external tools with multiple existing LMSs. This approach requires the creation of "adapters", both for the tools and the LMSs, that are connected via a specific management module; hence, it is suited for typical learning tasks, when there is a high probability to reuse the adapters for external tools. However, we consider that in the field of cybersecurity training, where there is a huge diversity of tools and no standardization, the specialized approach that we propose is more appropriate and less complex.

Reference [18] presents work on the creation of a middleware for integrating "web labs", defined as programs that allow learners to execute experiments remotely, with LMSs. The approach is simpler compared to GLUE!, and it is based on the use of the SCORM standard as means for providing generality in terms of supported LMSs, similar to our approach. However, this work also focuses mainly on the integration mechanisms, with no effort related to discussing, for instance, how the SCORM packages are to be generated. We consider that our holistic approach, which also takes into account the need to facilitate the task of content creation, is as a result superior from the user experience point of view.

In this context we would also like to mention the Learning Tools Interoperability (LTI) standard developed by the IMS Global Learning Consortium [11]. This standard is intended as a generic framework for managing the interactions between LMSs and external tools, and the most recent version, LTI 1.3, focuses on improving the security specifications of the standard. Nonetheless, in niche areas like cybersecurity training, there may be limited advantages of following such a complex standard, and dedicated solutions such as ours are preferable. Potential implementers should always consider the trade-offs involved, and the fact that this standard is only available via membership, hence it is not well-suited for non-commercial education efforts in small institutions.

In addition to the training programs, such as SANS Netwars [19] and the CTF competitions, that we have mentioned already, there are also specific efforts towards conducting such activities via integration with LMSs. For instance, the authors of [21] propose a tight integration of Moodle with "virtual labs", environments designed for making possible hands-on sessions. The tight integration makes it so that security training appears as a specific type of activity in the Moodle interface. Yet, there is no discussion about automation, and the activities are defined at virtual machine level, with no description about how the training content (e.g., questions trainees need to answer) is created or displayed. CyLMS, on the other hand, takes into account all these important factors.

Another effort towards cybersecurity training via e-learning, SecLab, is described in [10]. A main difference with respect to our work is that the authors created their own e-learning platform, hence did not leverage the advantages of established LMSs such as Moodle. The paper discusses in detail the training content presently available in SecLab, but does not consider extensibility from this point of view, hence we conclude that SecLab provides less flexibility than CyLMS.

7 Conclusion

In this paper we have presented CyLMS, a support system for cybersecurity education and training that leverages the advantages of LMS software to facilitate the related activities. The features of CyLMS can be split into two categories, related to training content representation, and to a tighter integration with the LMS.

Regarding training content representation, the two main characteristics are as follows: (i) Use a text-based representation that simplifies content editing, sharing and versioning, while offering sufficient content-creation flexibility to educators; (ii) Convert the training content to a standard SCORM package, which can be imported into most modern LMSs.

As for the tighter integration with the LMS, the following functionality is currently supported by CyLMS: (i) Automatically add and remove activities into the Moodle LMS based on the converted SCORM package; (ii) Provide functionality for accessing the associated training environment via the LMS interface, thus improving the user experience; (iii) Include other LMS integration code into the SCORM package that make possible to conduct interactive sessions, in which trainees can take control of the training conditions, so as to improve the training effectiveness and reduce the instructor workload.

The evaluation of our system started with a functionality validation that showed the improvements that the combination of LMS software and CyLMS

brings to cybersecurity training, how the implementation itself was validated, and the applicability of CyLMS to other areas, such as security awareness training. The user evaluation that followed demonstrated that CyLMS can be used to develop training content and conduct actual training activities in higher education institutions, with positive feedback being received both from educators and students. Furthermore, the performance evaluation indicated that CyLMS has a very low processing overhead, under 2 seconds in our experiments, hence its installation would not have a significant impact on existing LMS setups.

CyLMS is under active development, and we are currently working on extending the representation format to give more content-creation freedom to educators (e.g., multiple correct answers per question). Other future work refers to the interactive training features of CyLMS, in particular to providing an API-like mechanism for associating LMS interface elements (buttons, etc.) with custom user code for more flexibility.

Acknowledgements The authors would like to thank Muhammad Harith bin Noor Azam for the initial implementation of content management via `moosh`, and Masanori Sunagawa for the prototype implementation of remote desktop access via `noVNC`. This work was supported by JSPS KAKENHI Grants Number 17K00478 and 17K00479.

References

1. Advanced Distributed Learning (ADL) Initiative: SCORM Overview. https://www.adlnet.gov/scorm
2. Alario-Hoyos, C., Bote-Lorenzo, M.L., Gomez-Sanchez, E., Asensio-Perez, J.I., Vega-Gorgojo, G., Ruiz-Calleja, A.: GLUE!: An architecture for the integration of external tools in Virtual Learning Environments. Computers & Education **60**(1), 122–137 (2013)
3. Bartnes, M., Moe, N.B., Heegaard, P.E.: The future of information security incident management training: A case study of electrical power companies. Computers & Security **61**(Section C), 32–45 (2016)
4. Beuran, R., Pham, C., Tang, D., Chinen, K., Tan, Y., Shinoda, Y.: Cybersecurity Education and Training Support System: CyRIS. IEICE Transactions on Information and Systems **E101-D**(3), 740–749 (2018)
5. Beuran, R., Tang, D., Pham, C., Chinen, K., Tan, Y., Shinoda, Y.: Integrated Framework for Hands-on Cybersecurity Training: CyTrONE. Elsevier Computers & Security **78C**, 43–59 (2018)
6. Brin, S., Page, L.: The Anatomy of a Large-scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems **30**(1–7), 107–117 (1998)
7. Cyber Range Organization and Design (CROND): GitHub Repository for CyLMS. https://github.com/crond-jaist/cylms
8. DBpedia Association: DBpedia Website. https://wiki.dbpedia.org/
9. Evans, C.: The Official YAML Website (2017). http://www.yaml.org/
10. Ghiglieri, M., Stopczynski, M.: SecLab: An Innovative Approach to Learn and Understand Current Security and Privacy Issues. In: Proceedings of the 17th Annual Conference on Information Technology Education (SIGITE '16), pp. 67–72 (2016)
11. IMS Global Learning Consortium: Learning Tools Interoperability Website. http://www.imsglobal.org/activity/learning-tools-interoperability
12. Jeffrey, C.: tty.js: A terminal for your browser, using node/express/socket.io. https://github.com/chjj/tty.js/
13. Md. H. Noor Azam, Beuran, R.: Usability Evaluation of Open Source and Online Capture the Flag Platforms. Tech. Rep. IS-RR-2018-001, Japan Advanced Institute of Science and Technology (JAIST) (2018)
14. Muras, T.: Moosh Official Website. https://moosh-online.com/
15. National Institute of Information and Communications Technology, Japan: Cyber Defense Exercise with Recurrence (CYDER) (in Japanese). https://cyder.nict.go.jp/

16. noVNC Development Team: noVNC: HTML VNC Client Library and Application. https://github.com/novnc/noVNC
17. Rustici Software: Sample SCORM Packages. https://scorm.com/scorm-explained/technical-scorm/golf-examples/
18. Sancristobal, E., Castro, M., Harward, J., Baley, P., DeLong, K., Hardison, J.: Integration View of Web Labs and Learning Management Systems. In: Proceedings of IEEE EDUCON 2010 Conference, pp. 1409–1417 (2010)
19. SANS Institute: SANS NetWars Training Courses. https://www.sans.org/netwars/
20. Scarfone, K., Souppaya, M., Cody, A., Orebaugh, A.: National Institute of Standards and Technology – Technical Guide to Information Security Testing and Assessment (2008)
21. Soceanu, A., Vasylenko, M., Gradinaru, A.: Improving Cybersecurity Skills Using Network Security Virtual Labs. In: Proceedings of International MultiConference of Engineers and Computer Scientists (IMECS 2017) (2017)
22. Tan, Z., Hasegawa, S., Beuran, R.: Concept Map Building from Linked Open Data for Cybersecurity Awareness Training. In: Proceedings of Japanese Society for Artificial Intelligence (JSAI) Special Interest Group on Advanced Learning Science and Technology Workshop (SIG-ALST83), pp. 1–6 (2018)
23. Tang, D., Pham, C., Chinen, K., Beuran, R.: Interactive Cybersecurity Defense Training Inspired by Web-based Learning Theory. In: Proceedings of IEEE 9th International Conference on Engineering Education (ICEED 2017), pp. 103–108 (2017)
24. The Moodle Project: AIKEN Format. https://docs.moodle.org/en/Aiken_format
25. The Moodle Project: GIFT Format. https://docs.moodle.org/en/GIFT_format
26. The Moodle Project: Moodle XML Format. https://docs.moodle.org/en/Moodle_XML_format
27. Web Application Security Forum: Hardening Project (in Japanese). https://wasforum.jp/hardening-project/
28. Woo, Y., Reeves, T.C.: Meaningful interaction in web-based learning: A social constructivist interpretation. The Internet and Higher Education **10**(1), 15–25 (2007)