published as: J. Hu, Christoph Meinel, M. Schmitt, Ch. Willems: A Tutoring System for IT-Security;
Proceedings of the 3rd World Conference in Information Security Education (WISE 2003);
IFIP WISE 2003, Monterey (California, USA), 2003, pp. 51-60.

A TUTORING SYSTEM FOR IT SECURITY

Ji Hu, Michael Schmitt, Christian Willems, and Christoph Meinel
University of Trier, Research group "Institute for Telematics",
Bahnhofstraße 30-32, 54292 Trier, Germany
{hu,schmitt,willems,meinel}@telematik-institut.de

Abstract: Due to the many vulnerabilities of today's computer systems, IT security education has become an important topic. For that reason, a new tutoring system is developed at the Institute for Telematics, Trier, that allows users to gain knowledge about security technologies and tools via a web browser interface. Unlike other systems, this tutoring system does not provide a restricted simulation environment. Instead, guided exercises are performed on a real system (Linux). In this paper, the user interface and architecture of the tutoring system as well as some implementation aspects and future enhancements are described.

Key words: IT Security, Tutoring System

1. INTRODUCTION

Today's IT systems are suffering from various kinds of hacker attacks. Pitifully, many of these attacks succeed because people do not know about the vulnerabilities of their systems and how to defend against attacks. Therefore, IT security education has become a hot topic.

Recently, many universities have integrated lectures on computer security into their curricula. For example, teleteaching courses are currently held at the University of Trier, Germany. One of the courses is a joint project of the University of Trier and the Beijing University of Technology, China, in which students of both countries are made familiar with aspects of Internet security.

Besides knowing the theory, practical experience and the acquisition of practical skills are essential. Therefore, the Institute for Telematics develops

In Proceedings of the 3rd World Conference on Information Security Education (WISE 2003), June 2003, Monterey, CA, USA, pp. 51-60.

a tutoring system for IT security, called *Lernplattform IT-Sicherheit* (LPF; English: learning platform IT security). The LPF is based on web technologies so that users can interact with it via a web browser. It offers both a German and an English frontend.

The LPF is designed for Linux. This operating system has been chosen because it is freely available and gains increasing popularity. It also provides a vast number of open source security and hacker tools. This makes it possible to ship the LPF as a complete system that does not require additional, commercial software. Currently, the LPF is based on SuSE Linux 8.1, the Apache web server, and the PHP and Perl script languages. For practical exercises, various security tools and auxiliary programs are used such as OpenSSL, the Nmap port scanner, the Nessus security scanner, John-the-Ripper (a password cracker), and the Snort intrusion detection system.

The lectures of the LPF cover general security topics as well as specific aspects of the Linux operating system. The range of topics includes cryptography and secure email, authentication, firewalls, intrusion detection, viruses, and security scanning. In order to follow the course, a user is required to have some rudimentary knowledge about Unix concepts (e.g., how to enter commands in a shell, how to create a subdirectory or delete a file).

For every topic, the LPF provides a set of exercises. These guided exercises are performed on the real Linux system with standard tools rather than in a closed but restricted simulation environment. This approach allows users to easily apply their knowledge to production systems later.

Document structure. This paper is structured as follows: Section 2 provides background materials and discusses related work. The system architecture of the LPF and its functional components are described in Section 3. Finally, Section 4 gives a summary and presents future enhancements.

2. RELATED WORK

Intelligent tutoring systems (*ITS*) are the next generation of *computer-based tutoring* (CBT) systems (Sleeman and Brown, 1982). An ITS is able to communicate with a user to assess her results and to teach the user in a manner that is appropriate according to her knowledge and skills (Persché, 1997).

There are a few known projects dealing with security training. The Chalmers University of Technology, Sweden, has a long-term project in which users evaluate the security of a target system by attacking it (Lindskog et al., 1999). However, the exercises are guided by an instructor instead of a computer-based training system.

The *ID-Tutor* (Rowe and Schiavo, 1998) and the ITS described in (Woo et al., 2002) are computer-based tutoring systems for becoming acquainted with intrusion detection. They allow users to perform practical tasks in a simulation environment. The user's answers are evaluated by comparison with predefined or fixed criteria.

The ID-Tutor creates audit files with information on user login and executed user commands. The user has to decide whether an intrusion has occurred and, in case of an intrusion, she must resolve the problems. The ID-Tutor provides a simple interactive operating environment in which the student chooses Unix commands from a menu instead of entering commands on a console. Unfortunately, the generated audit files merely cover a few selected problem cases. In addition, the ID-Tutor does not record and analyze the student's performance.

The ITS described in (Woo et al., 2002) is similar to the *ID-Tutor* but it has an intelligent tutoring system architecture and generates its missions from a knowledge base. The ITS provides an interactive, virtual Unix system with a command line interface. However, the virtual operation system provides only a limited set of commands and a simple directory structure. In order to finish a mission successfully, students have to follow a strict problem resolving path.

3. SYSTEM ARCHITECTURE

The primary tasks of an ITS are the modeling of (1) the knowledge of the domain (domain model), (2) the user (student model), and (3) the pedagogical strategies (tutor model). Furthermore, the development of an ITS must focus on the creation of a powerful user interface (Sleeman and Brown, 1982).

The architecture of the LPF is designed according to these ideas. The main components are illustrated in Figure 1. The *content library* corresponds to the domain model. It is an IT security knowledge base represented by a collection of web pages and scripts. The *navigator* is the corresponding entity to the tutor model. It guides the user through security topics in the content library, prepares and assigns exercises to the user, and processes feedback from the user. The *user object* is the counterpart of the user model. It keeps track of the user's knowledge at every stage in the learning process. The web browser provides the student with a user-friendly and uniform interface. In order to perform the exercises, the user must use a command shell or an X window application.

Figure 1. System Architecture

3.1 THE CONTENT LIBRARY

The content library is a knowledge base that consists of three types of contents:

1. descriptions of security concepts
2. descriptions of security tools
3. security exercises

The first two types represent declarative knowledge that is presented to the user in hypertext form mixed with multimedia objects, such as graphics, images, or animations. The security exercises reflect procedural knowledge related to how a security task is performed. The implementation of the procedural knowledge is much more complicated because it must be presented in terms of scenarios. This means, we must carefully design the steps and actions in the exercises.

The organization of the content library is illustrated in Figure 2. The basic unit is a section. A section represents a complete learning item or several closely related items. A section is also the basic unit for measuring the user's performance. Completing a section means the user has succeeded in acquiring the given knowledge or skills. The LPF has three types of sections including concepts, tool usages, and exercises. They correspond to the three

types of knowledge described above. Every section consists of one or more pages. The exercise sections comprise some additional scripts.

Multiple sections are combined in a chapter that represents a security topic. In most cases, a chapter introduces some security concepts first. Then it explains some tools or commands. Afterwards, the user is asked to perform some practical exercises.

Technically, for each section, there is a description file that provides the navigator (see below) with meta information, including the section type, the number of pages, the title, etc. A chapter description file specifies how a chapter is organized. Finally, a profile description file defines which chapters are reasonable for which type of user. Based on these description files, the navigator is able to construct a hierarchy of chapters and sections and to produce appropriate hyperlinks. The use of description files makes it very easy to adapt user profiles and the structure of chapters in the future, without having to touch the individual sections. Instead, only items in the description files have to be changed for that purpose. Furthermore, it allows to define several chapters that share some common sections.

3.2 THE NAVIGATOR

The main goal of the navigator is to deliver the materials in the content library in a structured manner to the user. As a gateway to the tutoring system, the navigator is also able to track every communication with the user and analyze her learning process.

The navigator knows exactly which pages belong to a section and which sections belong to a specific chapter. The navigator creates linked web pages that are displayed in the web browser of the user (see Figure 2). It also decides whether the user has finished a section successfully and where to continue at the end of a section.

When starting the LPF, the user must register herself or login into the LPF with a valid account. Then, the navigator provides a list of available chapters from which the user can choose. When the user enters a chapter, the navigator creates a navigation bar on the left side of the web page that lists all sections and represents their type by a small icon (see Figure 3).

Help information is available for each exercise. This information is provided in a popup window as a series of questions and answers (see Figure 4).

Figure 2. Content organization and navigation

3.3 EXERCISE MANAGEMENT

Exercises are specified as Perl and PHP scripts in the content library and interpreted by the navigator. An exercise takes place in three steps (cf. Figure 2):

In the first phase, the working environment, i.e. the Linux operating system, is configured. For example, if the user is to perform security scans, a set of services are activated so that the user gets reasonable results.

The next phase deals with generating questions or tasks and passing them to the user. Where possible, these tasks are created dynamically. I.e., all users do the same type of exercise but with different detailed content. For example, for password cracking, a Unix *passwd* file (that the users must decrypt) is generated at run-time. Exercises must be created in such a way that the degree of difficulty does not vary. For instance, passwords created in the *passwd* file are selected randomly from the same dictionary and can be cracked in similar time.

After the user completes her task, the LPF evaluates her result.

In some cases, it may also be necessary or useful to generate some background load *during* an exercise. For instance, the tutoring system might instantiate a *telnet* session while the user practices network sniffing so that some critical passwords can be observed in the data packets.

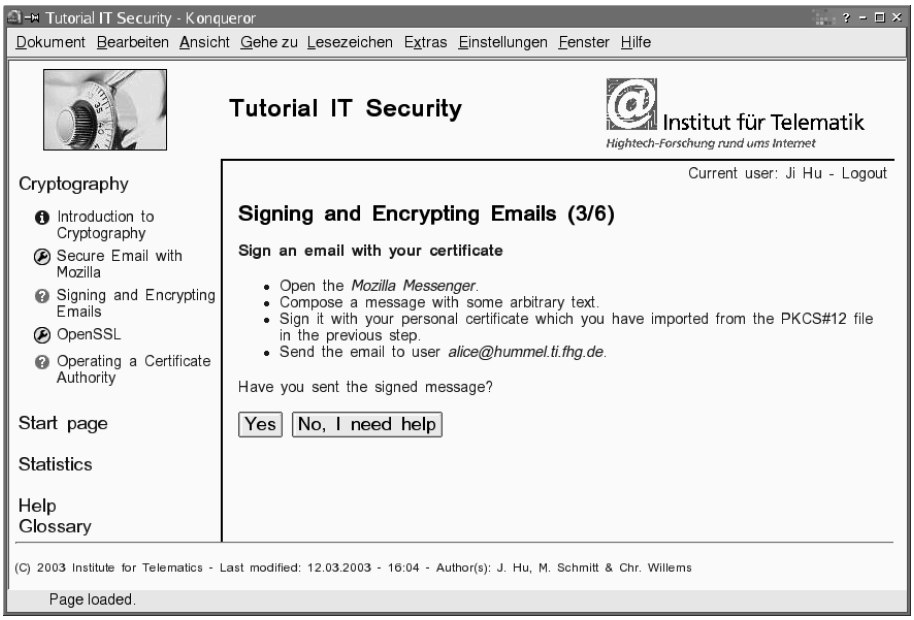

Figure 3. Exercise example: Secure email

Example. The preparation, execution, and result analysis of exercises in a real system environment is a complicated task for which many technical problems have to be solved. The effort needed to set up a proper environment is illustrated by an exercise for secure email. Its purpose is to make users familiar with certificates so that they can sign and encrypt emails. The LPF user interface for this exercise is shown in Figure 3.

Phase 1: Preparation of the working environment:
- clear previous settings and create the necessary working directories
- set up a local mail server
- create a virtual user *Alice* for email communication with the user, i.e. create a Linux account and configure the email settings
- create a certificate authority by OpenSSL commands
- issue certificates to Alice and the user, install the certificate for Alice, and guide the user to import the certificate to the mail client

Phase 2: Generation of exercise:
- create a random message with the signature on behalf of Alice
- send the message to the user
- ask the user to

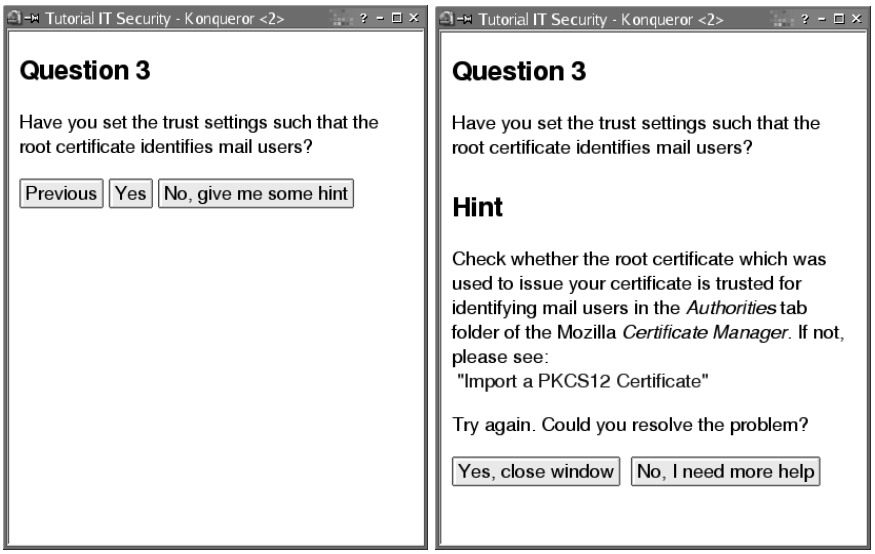

Figure 4. Help information

1. verify the signature attached to the message, accept and trust Alice's certificate
2. reply to Alice's message with her own signature
3. send a message encrypted with Alice's certificate to Alice

Phase 3: Evaluation of the answer:
- fetch a message from Alice's mailbox
- verify its signature to see whether it has been sent from the user and matches with the original
- try to decrypt the message with Alice's private key corresponding to her certificate
- record the (un-)successful completion of the exercise in the user object

3.4 THE USER OBJECT

An important feature that distinguishes an ITS from other tutoring systems is that it "knows" the individual users and tracks their performance. In the LPF, this feature is supported by the *user object*.

The user object contains two types of data: One is static personal information, such as name, password, language, and profile. The profile places the user into one out of (currently) three categories: administrators, end users, and students. The LPF defines different sets of security topics for each category.

The second class of data includes the records of completed sections, the time needed to complete an exercise, the link to the last visited web page, help requests and their frequencies, and so on. In order to complete a topic, the user must pass all exercises in a chapter. A user can repeat an exercise until she finally finds a correct solution. The data in the user object are used to analyze the user's performance and to present statistics on the current status.

Technically, user objects are implemented by a PHP class. When a user logs in or logs out, the user object is read from/written to a floppy. Thus, even if the user corrupts the Linux systems by some exercise, her test results are still accessible.

4. SUMMARY AND OUTLOOK

In this paper, we have presented a web-based tutoring system for IT security. The LPF improves the existing security education activities by several features: it offers users a real system environment instead of a limited simulation environment; it has a navigation mechanism that presents contents and creates exercises dynamically; its user interface is based on a web browser interface. The tools and programs needed in the exercises are available via the browser or already installed on the system.

As of April 2003, the main framework of the LPF is completed. In addition, 5 out of 11 planned chapters are available. These chapters deal with authentication, secure email, certificates and PKIs, data security, network services, and port scanning.

The web server requires root rights for the preparation of exercises. Sometimes, even the user needs to execute privileged commands to perform certain exercises. This introduces the risk that the user spoils the system. Since the LPF runs on the same operating system on which the user performs her exercises, this may lead to a complete break-down. Currently, the LPF must be reinstalled on hard disk in such a case. We plan to combine it with a Linux system that can be booted from CD and does not require any hard disk installation. Another approach that is investigated is the use of virtual machines on top of a base Linux system (Dike, 2000). We are also examining the possibility to provide the LPF as an online tutoring system.

Furthermore, we are looking for a way to provide a pure browser-based user interface where external applications are displayed inside the browser. This would avoid the necessity to switch between different application windows during exercises. A possible solution is to embed a Java applet into the web page that connects to a Linux server (Cao et al., 2002).

REFERENCES

Cao, Jiannong, Chan, Alvin, Cao, Weidong, and Yeung, Cassidy (2002). Virtual Programming Lab for Online Distance Learning. In *Proceedings of the First International Conference*, ICWL 2002, pages 216-227, Hongkong, China.

Dike, Jeff (2000). A user-mode port of the Linux kernel. In *Proceedings of the 4th Annual Linux Showcase & Conference,* page 63, Atlanta, GA. Usenix.

Lindskog, Stefan, Lindqvist, Ulf, and Jonsson, Erland (1999). IT Security Research and Education in Synergy. In *Proceedings of the 1st World Conference on Information Security Education*, Stockholm, Sweden.

Persché, Richard (1997). Immediate Feedback During Online Lectures. Master's thesis, Institute for Information Processing and Computer Supported New Media of the Graz University of Technology, Graz, Austria.

Rowe, N. C. and Schiavo, S. (1998). An Intelligent Tutor for Intrusion Detection on Computer System. *Computers and Education*, pages 395-404.

Sleeman, D. and Brown, J.S. (1982). *Intelligent Tutoring Systems*. Academic Press Ltd., London.

Woo, Chong-woo, Choi, Jin-woo, and Evens, Martha (2002). Web-based ITS for Training System Managers on the Computer Intrusion. In *Proceedings of the 6th International conference* ITS 2002, pages 311-319, Biarritz, France and San Sebastian, Spain.