Forschungsbericht Nr. 03–06

Design and Implementation of a PHP-based Web Server for the Tele-Lab IT Security

Michael Schmitt, Ji Hu, Christoph Meinel
Forschungsprojekt Institut für Telematik
Fachbereich IV – Informatik
Universität Trier, 54286 Trier, Germany
michael.schmitt@teststep.org, {hu,meinel}@ti.uni-trier.de

Design and Implementation of a PHP-based Web Server for the Tele-Lab IT Security

Michael Schmitt, Ji Hu, Christoph Meinel
michael.schmitt@teststep.org, {hu,meinel}@ti.uni-trier.de

December 2003

Faculty IV – Computer Science
University of Trier, DE-54286 Trier, Germany

Abstract

The Tele-Lab IT Security project aims at specifying and implementing a web-based, intelligent tutoring system that allows computer science students, system administrators, and end users to get familiar with the basics of IT security. It provides a powerful, real-life working environment in which users can develop and practice solutions for problems of their every-day work with only little support by the teaching staff.

This technical report focuses on the web-browser part of the Tele-Lab. It discusses the design and the implementation of a web framework that is responsible for user administration and for presenting the teaching contents in a user-friendly manner.

1 Introduction

Due to the increasing use of the Internet, the secure operation of IT systems has gained vital economic and social importance. Accordingly, the awareness and education of users will play an important role in the future.

As a consequence, more and more universities integrate information security into their curricula. In this connection, it is not sufficient to teach only the theoretical foundations of information security — the users must also gain practical experience.

However, practical education by means of a dedicated computer laboratory typically leads to big administration problems, because disturbances are likely to occur. Furthermore, for financial reasons only few institutions can afford a physically separated test network with many heterogeneous systems. On the other hand, if students operate in

the main network with administrator privileges, this implies serious security risks. For these reasons, an approach is needed that reduces the amount of administration.

Currently, IT security education is assisted by computers in four ways:

- Multimedia teaching contents (e.g., Fraunhofer, 2002)
- Software tools (e.g., CrypTool, Esslinger and Eckert, 2002)
- Tutoring systems for specific problem areas of IT security (e.g., Woo et al., 2002)
- Secured computer networks for practical exercises

Unfortunately, no attempt is known so far that tries to combine all four directions in one system.

The Tele-Lab IT Security project aims at specifying and implementing a web-based, intelligent tutoring system that allows computer science students, system administrators, and end users to get familiar with the basics of IT security. It provides a powerful, real-life working environment, in which users can develop and practice solutions for problems of their every-day work. In contrast to other existing tutoring systems that operate in restricted simulation environments, the users gain practical experience on a real system with standard applications.

The Tele-Lab IT Security is a multi-disciplinary computer science research and development project that was initiated at the chair of Prof. Dr. Christoph Meinel at the University of Trier. It combines cognitions about information security (concepts, tools, and applications), intelligent tutoring systems, Internet and WWW technologies, operating systems, and education.

The concepts that have been elaborated so far were published in several conference and journal papers (Schmitt et al., 2003; Hu et al., 2003, 2004). This report focuses on the web-browser part of the Tele-Lab. It discusses the design and the implementation of the web framework that is responsible for user administration and for presenting the teaching contents in a user-friendly manner. As such, it deals as a technical reference for the ongoing development of the Tele-Lab It Security, as well as for projects with a similar objective.

This paper is structured as follows: In section 2, an introduction to the system architecture of the Tele-Lab IT Security is given. Section 3 discusses the support of user profiles. The structuring of the teaching contents by means of meta data is explicated in section 4. Functions of the web framework that ease the development of multimedia teaching contents are described in section 5. The creation and validation of HTML documents for the Tele-Lab IT Security are discussed in section 6. The technical report concludes with a short summary in section 7.

2 System Architecture

For the Tele-Lab IT Security, an infrastructure and architecture is elaborated that both allows to present the teaching contents in a suitable way and ensures secure training.

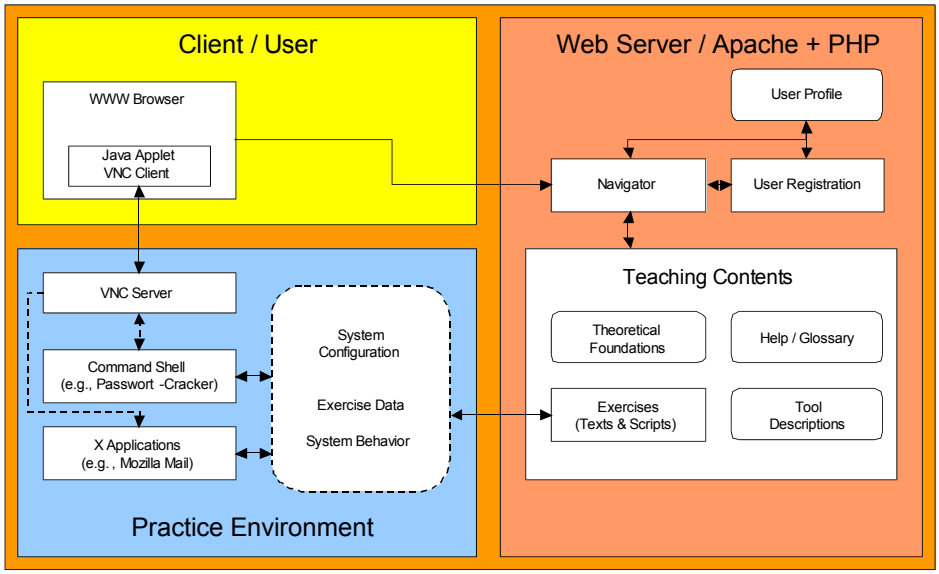

Figure 1: Tele-Lab IT-Security system architecture

The overall system architecture is presented in figure 1. As illustrated, the Tele-Lab is divided into three logical units: the client, the practice environment, and the web server.

On the user's side, a web browser serves as the interface to the tutoring system. It communicates with a web server that is responsible for outputting the teaching contents and administrating the user profiles. Based on structural information about the lectures, the web server creates dynamically linked HTML pages that provide the user with many-folded but also controlled navigation facilities.

Technically, the web server is an Apache server running on a SuSE Linux operating system. To create HTML pages dynamically, the PHP Hypertext Preprocessor (Apache Software Foundation, 2003) is used. PHP is a general-purpose scripting language that is especially suited for web development. Its main advantage in comparison with other scripting languages such as Perl is that it can be embedded into and mixed with plain HTML.

For practical exercises, a host is assigned to the user that can be accessed via a Java applet. In the past, a stand-alone prototype has been developed where the practice environment is identical to the host of the user. However, efforts are made to realize a secured remote practice environment by means of virtual machines (VMs). Virtual machines allow to simulate several hosts with varying operating systems on a single physical machine. For the Tele-Lab IT Security, User Mode Linux (UML; Dike, 2000) was chosen as technical solution. Open issues concern the efficient administration

of the virtual machines. In particular, techniques have to be developed that reduce the memory requirements of VMs and the size of their file systems. These techniques are vital to achieve acceptable startup times (which might be necessary with each new lecture) even under heavy load.

In principle, the communication between applet and VM can take place via different protocols. Experiments have shown that the *Remote Framebuffer Protocol* (*RFB*) of the *Virtual Network Computing* (*VNC*; RealVNC Ltd., 2003) is well-suited for this purposes as it works reliably even with low bandwidths.

The tutoring system is not a closed system. The possibility to gain practical experience in a user-owned practice environment with standard applications makes it necessary to develop complex kinds of interaction among tutor, practice environment, and user.

A crucial task of the tutoring system is to prepare the practice environment in such a way that the user is able to perform his exercise in a controlled manner. For instance, if the user wants to practice the sending of confidential emails by means of certificates, the tutoring system has to set up a local mail server in the practice environment first. Then, it has to create a virtual user with which the user can communicate. Finally, certificates for both the user and the virtual partner must be issued by a certificate authority (CA). During the exercise, the tutoring system is responsible for controlling the communication of the virtual user at different points in time (by reading the mail box and by encrypting, decrypting, signing, and sending mails). After task completion, an automatic assessment is made without any further interaction with the user.

3 User Profiles

For each user, the Tele-Lab IT Security maintains a user profile which comprises static and dynamic data. These data are stored in a PHP object. The corresponding PHP class is defined in file `./Include/user.php` on the web server.

Each user profile contains the following information:

- General user information
  - name – The full name of the user.
  - account – The account name under which the user logs in.
  - password – The password with which the user authenticates.
- Tele-Lab settings
  - language – The preferred language. At present, the user may choose between *English* (`en`) and *German* (`de`).
  - profile – The user group to which the user belongs. It determines the selection, the technical level, and the order of lectures. Currently, the user may choose between *System administrator*, *User*, and *Student* (see also section 4.1).

- Practice environment
  - virtual machine – The name or IP address of the machine on which the user performs his/her exercises. In the standalone CD version of the Tele-Lab IT Security, this variable is set to `localhost`.
- Current page
  - bookmark – The address of the previously visited page. The bookmark allows the user to continue her/his courses seamlessly even if (s)he log outs and starts a new session later.
- History – Success tracking
  - visits – The number of times each section has been visited. The counter is increased every time the first page of a specific section is invoked.
  - completions – The number of times each section was finished completely. This information is mainly relevant for sections that comprise exercises.
  - duration – The time spent on the pages of a specific section. The duration is measured as the time between the invocation of the web page itself and the invocation of the next web page.

  All three information are stored in a multi-dimensional map (associative array) that is constructed at run-time. In this way, the user profile management does not have to know about the precise teaching content structuring.

The general user information and the Tele-Lab settings are determined during the user registration. Figure 2 shows the corresponding registration dialog.

All other data are updated dynamically during the course. Based on the history information, the Tele-Lab IT Security is able to provide statistics that are presented on a special statistics web page (see figure 3).

Depending on the mode of operation, the user profile is stored at different locations. If the Tele-Lab IT Security is booted from a CD-ROM and executed locally, the user data are saved on floppy disk. In this way, the data are preserved even in case of a system crash.

If, on the other hand, the Tele-Lab IT Security is placed on a remote web server, the user profile is stored in a special `./Users` directory on the web server. In this scenario, the web server is considered to be secure and stable as the exercises are made on a virtual machine.

In principle, the user profile can be serialized and saved after each user request to the web server. However, this is too costly in practice, because save operations on a slow medium like a floppy disk cause unwanted delays. Therefore, the storage of the user profile is only triggered if (a) the user registers, (b) a new section is visited, (c) a section has been completed, (d) the user spent more than 5 minutes on the previous page, or (e) the user logs out.

Figure 2: User registration

Figure 3: Statistics page

```
$profiles['administrator'] =
  array( 'name' => array( 'de' => 'System-Administrator',
                          'en' => 'System Administrator' ),
         'description' => array( 'de' => '...',
                                 'en' => '...' ),
         'chapters' => array( 'ServicesPortScanning', 'PacketSniffing',
                              'SecurityScanner', 'Firewalls',
                              'Auditing', 'IDS' ) );
```

Figure 4: User group specification

4 Content Structuring

In principle, the teaching contents should have as few dependencies with the Tele-Lab web framework – which is responsible for their presentation – as possible. Nevertheless, both sides have to provide some services and data to each other. In the following, those information are described that the web framework needs to present the teaching contents in a structured manner.

4.1 User Groups

Depending on the type of user, the requirements with regard to IT security may vary strongly. For instance, a typical end user needs information on the secure exchange of emails whereas a system administrator is more concerned with aspects of intrusion detection. Similarly, the depth in which a topic should be discussed may vary. The Tele-Lab IT Security takes this into account by compiling information and exercises individually for each user group.

Every user is assigned to one of (currently) three categories: administrators, end users, and students. Each user group is described by three attributes:

- name – The user group name with translations into various languages.
- description – A short textual description of the special properties of the given user group.
- chapters – An ordered list of chapters. Each chapter is denoted by a symbolic name such as `ServicePortScanning`. For each symbolic name, there must be an entry in array `$chapters` (see next section).

Technically, the information about the different user groups is stored in an associative array, called `$profiles`. The web framework reads the user group definitions from file `./Structure/profiles.php`. A sample entry is shown in figure 4.

Administrators of the Tele-Lab IT Security are free to modify or add new entries according to their specific needs without interfering with other functionalities of the web framework. In particular, chapters may be re-used for several user groups.

```
$chapters['Cryptography'] =
  array( 'name' => array( 'de' => 'Kryptographie',
                          'en' => 'Cryptography' ),
         'description' => array(
            'de' => 'Lernen Sie die Grundlagen von asymmetrischer
                     Verschlüsselung und Zertifikaten und tauschen Sie
                     verschlüsselte und signierte E-Mails mit Ihrer
                     virtuellen Praktikums-Betreuerin Alice aus.',
            'en' => '...' ),
         'image' => 'box_vis6',
         'sections' => array( 'Info-CryptIntro', 'Tool-SecureEmail',
                              'Exer-SecureEmail', 'Tool-Openssl',
                              'Exer-OperateCA' ) );
```

Figure 5: Chapter specification

4.2 Chapters

A chapter treats one security topic where the chapter again is split into several sections: starting with a concrete problem (e.g., sending confidential emails), the user is introduced into the theoretical concepts (e.g., certificates) and the required software tools (e.g., Mozilla Messenger). Then, the user is requested to perform some exercises within his/her practice environment.

Each chapter is characterized by four attributes:

- name – The chapter name, translated into various languages.
- description – A short textual description of the objective of the chapter.
- image – The name of an image. The actual image is loaded from file `./Image/`*name*`.jpg`.
- sections – An ordered list of sections. Each section is referred to by a symbolic name such as `Info-CryptIntro`. For each symbolic name, a corresponding section must be defined. The Tele-Lab IT Security expects the section content and the section meta data in directory `./Sections/`*name*`/` (see also next section).

In analogy to user groups, the chapter meta data are stored in an associative array, called `$chapters`. The web framework reads the chapter definitions from file `./Structure/chapters.php`. A sample entry is shown in figure 5. Once again, the configuration can be adapted without causing problems such as broken hyperlinks. In addition, sections may reoccur in several chapters.

Based on the chapter meta information, the web framework is able to create an overview page dynamically. An excerpt of such an overview page is shown in figure 6.

Moreover, each chapter has its own entry page (which is not identical to the first page of its first section). The web framework expects this page as `./Chapters/`*chaptername*`.php`. If no such file has been specified by the content provider, the Tele-Lab IT Security creates a generic web page. It shows a table of contents with all sections listed.

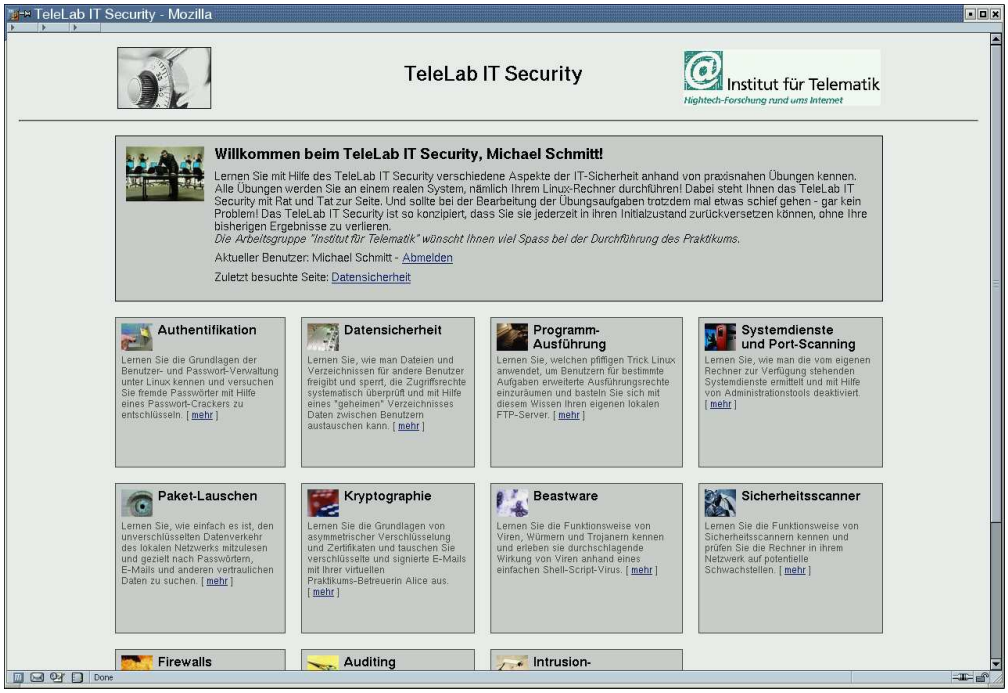

Figure 6: Overview page

4.3 Sections

As mentioned above, by convention, a section with symbolic name *secname* is stored in directory `./Sections/`*secname* on the web server. This allow the web framework to identify and load the teaching contents on demand.

For each section, the web framework has to know some meta data. They are needed for constructing the table of contents and the navigation bar. In detail, the following information must be given:

- name – The section name, translated into various languages.
- type – The type of teaching content; either `information`, `tool`, or `exercise`.
- pages – The number of section pages.

By convention, the meta data are stored in the section directory in file `section.php`. Technically, they are passed to the framework as elements of the associative array `$sections`, where the symbolic section name is used as index. (Note: The symbolic name must be identical to the section directory.) In figure 7, a sample configuration is given for section `Exer-OperateCA`.

Among others, the section meta data are used to set up the navigation bar. As illustrated in figure 8, suitable icons are displayed depending on the type of each individual section.

```
<?php
  $sections['Exer-OperateCA'] =
    array( 'name'
             => array( 'de' => 'Verwaltung einer Certificate Authority',
                       'en' => 'Operating a Certificate Authority' ),
           'type' => 'exercise',
           'pages' => 4 );
?>
```

Figure 7: Meta information for section `Exer-OperateCA`

5 Web Framework Functions

In the preceding section, the meta data have been described that allow the web framework to present the teaching contents in a structured manner. Now, the opposite direction is investigated. In the following subsections, functions of the web framework are described that can be used inside PHP scripts that describe the teaching contents.

5.1 Hyperlinks

Although the navigation bar maintained by the web framework allows the user to browse through the (already activated) pages of sections, the content provider may also want to use hyperlinks in the content area, i.e., the right bottom area in figure 8. However, most requests to the web server are handled by one central PHP script (`main.php`). Therefore, the URLs are rather complex. The generic form for pages inside a section is

`http://`*`server`*`/`*`directory`*`/main.php?pos=section`
`&chapter=`*`chapter`*`&section=`*`section`*`&page=`*`page`*

In order to avoid complex, hard-coded links inside the PHP scripts, the web framework provides a set functions that insert anchor tags of the format

`<A HREF='...'>...</a>`

into the generated HTML pages. In detail, the following functions are given:

- startLink() – Creates a link to the start (i.e., overview) page.
- statsLink() – Creates a link to the statistics page.
- logoutLink() – Creates a link to the logout page.
- chapterLink($chapter, $label = null) – Creates a link to the title page of a chapter. If no label is passed, the name of the chapter is displayed inside the anchor.
- sectionLink($chapter, $section, $label = null) – Creates a link to the first page of a section. A call to this function is equivalent to pageLink($chapter,$section,1,$label).
- pagelink($chapter, $section, $page, $label = null) – Creates a link to some page inside a section. If no label is passed, the name of the section is displayed inside the anchor.

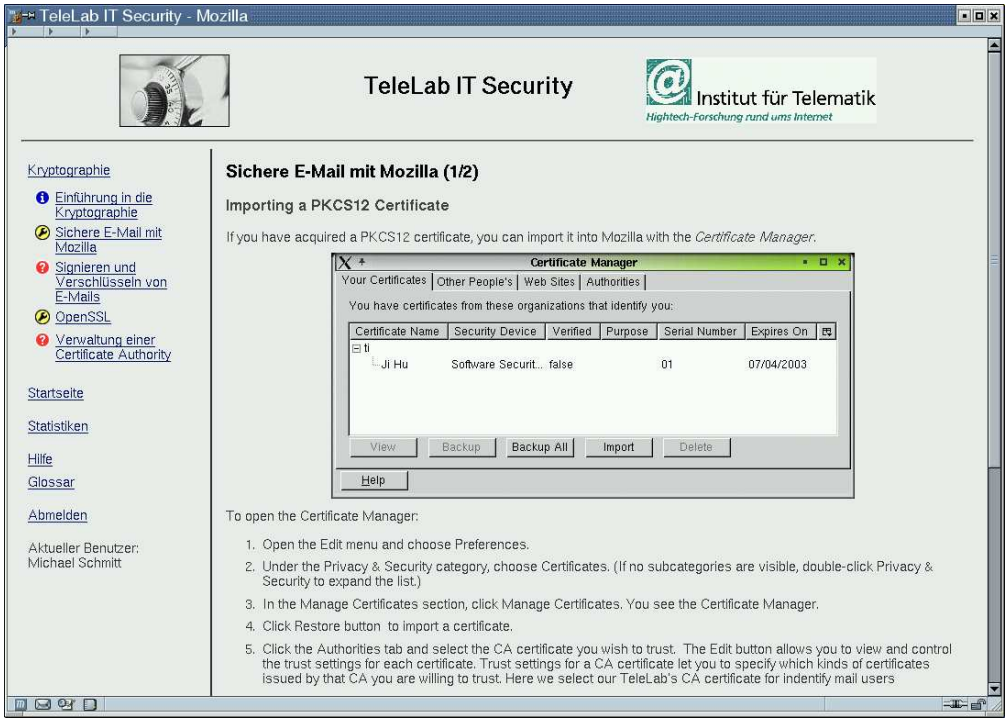

Figure 8: Navigation bar and teaching content

Unless specified manually, the labels are always translated automatically into the preferred user language. This relieves the content provider from writing even more code.

In many cases, a link to the logically next page is desirable. The hierarchically structured sections and pages should be linearized so that the user can follow the pages in a book-like way. For instance, at the last page of some section, the user should be directed to the first page of the succeeding section. Since the exact order of sections depends on the user group, the links must be determined dynamically. For that purpose, the web framework provides function nextPageLink() that computes the correct link at run-time.

All link-related functions are defined in `./Include/links.php`. They are included automatically into the main PHP script.

5.2 VNC Applet

With the Tele-Lab IT Security, the user is able to perform exercises within a practice environment. In a simple approach, the user performs them on his/her local computer. However, the Tele-Lab system architecture also considers a secured virtual system on a remote server. As mentioned in section 1, the *Virtual Network Computing* (*VNC*) software is used to transmit keystrokes and mouse events from the user to the server and to transmit changes on screen in the opposite direction.

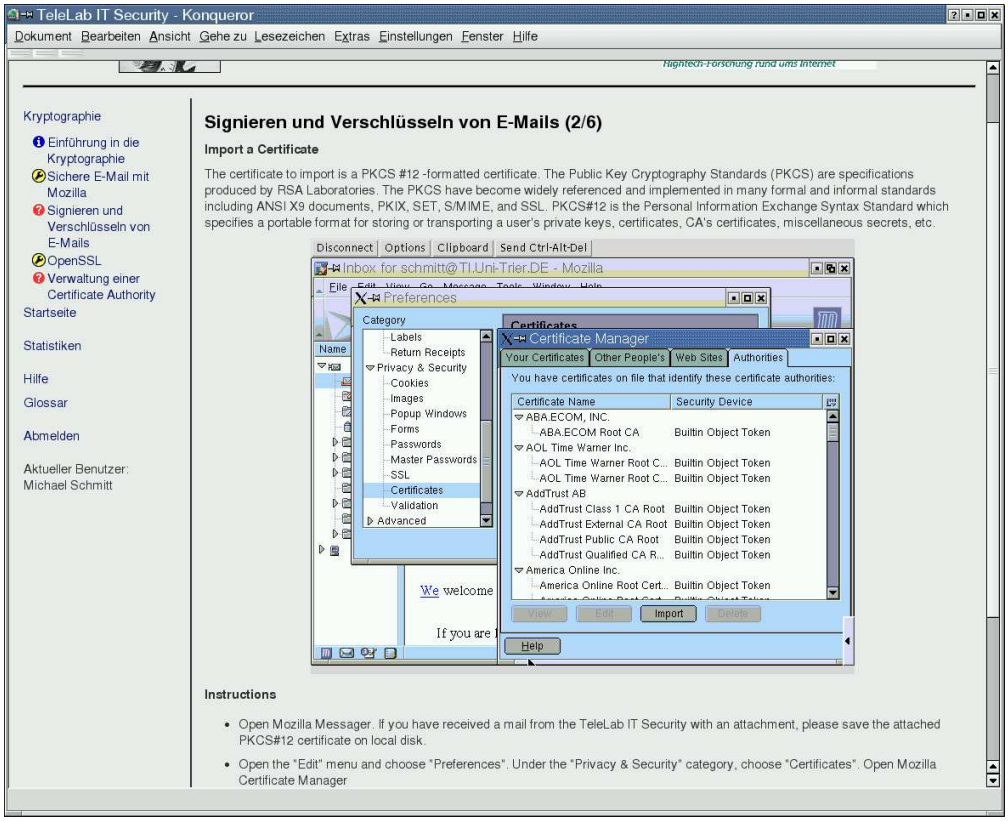

Figure 9: Embedded VNC applet

Ideally, the user should be provided with a coherent, purely browser-based interface. For that purpose, a VNC applet can be embedded inside the web page by means of an HTML `iframe` that downloads the applet from the target machine. The required HTML code is generated when calling PHP function openVNCFrame(). It is defined in `./Include/vnc.php`. A screenshot that demonstrates the inclusion of the VNC applet is given in figure 9.

5.3 Successful Completion of Sections

If a user has finished a section successfully, this fact must be registered in the user profile. Typically, a section is completed if the last page of a section is sent to the user. But in case of web pages that handle exercises, one PHP script may handle both the successful completion of an exercise and faulty results. To maintain maximum flexibility, the content provider must explicitly state the completion of a section. This is accomplished by invoking global function `setSectionCompletion()`. No information has to be provided about the current section as this information is already known to the web framework.

6 HTML Document Creation and Validation

Initially, the HTML pages of the web framework have been developed with Microsoft Frontpage. Compatibility tests with various Linux web browsers — Mozilla (Mozilla Organisation, 2003), KDE Konqueror (KDE (various authors), 2003), and Opera (Opera Software ASA, 2003) — have shown that the output of the HTML pages varied significantly among the browsers and between different browser versions. Moreover, it turned out to be a difficult undertaking to manually post-process the generated HTML documents as they were cluttered with a lot of formatting attributes.

It soon became clear that the only way to create lean, maintainable HTML pages was to directly edit the HTML documents. A very powerful HTML editor is Quanta (Laffon and various contributors, 2003). It supports syntax highlighting, creates HTML templates, and allows to check the output of HTML documents in a preview mode.

A precise formatting of HTML elements can be achieved with *Cascading Style Sheets* (*CSS*). Experience has shown that all modern web browsers support CSS satisfyingly and that the differences with regard to the formatting are only marginal. One conclusion that can be drawn is that very often less is more when it comes to formatting directives. Since CSS supports concepts like defaults and inheritance, it is not necessary to associate CSS styles or classes with each and every HTML tag. In fact, the spare use of CSS elements leads to better and more reliable results.

File `./Styles/common.css` defines a basic set of classes. A reference to this file is automatically contained in each HTML page generated by the web framework. Thus, these definitions also apply to the formatting of teaching contents. Inline *style* attributes within HTML tags should be avoided in the teaching contents.

To ensure maximum compatibility, web documents should conform to the standards of the WWW Consortium (W3C). There are mainly two relevant standards: HTML 4.01 (W3C, 1999) represents the latest version of the original standard. A reformulation of HTML by means of XML is specified as XHTML (W3C, 2002b). The Tele-Lab IT Security aims at conforming to XHTML 1.0 *Strict*, a cleaned-up XHTML version where all formatting must be expressed in terms of CSS.

Validation

To ensure that a web page indeed conforms to the standard referred in the document header, a validation tool should be applied. The W3C offers a web-based validation service (W3C Validator Team, 2002). It can be used by either specifying the URI of the document to be checked or by uploading the document.

The first approach is not applicable, since the Tele-Lab IT Security is run on a web server that is located within a protected network. As a consequence, the W3C Validator is not able to download the file from the server.

In contrary, the upload mechanism allows to pass a firewall. The Opera web browser supports (X)HTML validation by upload to the W3C service directly. Therefore, it is

not necessary to save a dynamically generated web page to disk and upload it afterwards with a command line tool or with the web form of the W3C Validator.

A similar service for the validation of cascading style sheets is available at (W3C, 2002a).

7 Summary

In this technical report, the design and implementation of the web server part of the Tele-Lab IT Security have been discussed. Despite its limitations in the area of object-orientation and type-safetyness, PHP 4 has turned out to be a solid basis for the rapid development of web applications. The web framework and its technical solutions can also be adopted to other kinds of e-learning systems that deal with structured contents. The future development of the Tele-Lab IT Security will concentrate on the integration of virtual machine technologies. In this context, questions concerning security and scalability will have to be investigated.

References

Apache Software Foundation (2003). PHP Hypertext Preprocessor. `http://www.php.net`.

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

Esslinger, B. and Eckert, C. (2002). CrypTool – Demonstrations- und Referenzprogramm für Kryptographie. `www.cryptool.de`.

Fraunhofer-Institut für sichere Telekooperation SIT, Bereich Innovationsberatung und Entwicklung (2002). Sicherheit im elektronischen Geschäftsverkehr. Ein Web-basiertes Training für Anwender. `http://www.sit.fraunhofer.de/german/hps1/sit_news/webtraining.html`.

Hu, J., Meinel, C., and Schmitt, M. (2004). Tele-Lab IT Security: An Architecture for Interactive Lessons for Security Education. In *Proceedings of the Technical Symposium on Computer Science Education (SIGCSE 2004)*, Norfolk, Virginia USA. ACM. Accepted paper.

Hu, J., Schmitt, M., Willems, C., and Meinel, C. (2003). A Tutoring System for IT Security. In Irvine, C. and Armstrong, H., editors, *Security Education and Critical Infrastructures — Proceedings of the Third Annual World Conference on Information Security Education (WISE-3)*, pages 51–60, Monterey, California. IFIP Working Group 11.8 and the Center for INFOSEC Studies and Research (CISR) at the Naval Postgraduate School, Kluwer Academic Publishers.

KDE (various authors) (2003). Konqueror. `http://www.konqueror.org`.

Laffon, E. and various contributors (2003). Quanta Plus Web Development Tool. `http://quanta.sourceforge.net`.

Mozilla Organisation (2003). Mozilla Web Browser. `http://www.mozilla.org`.

Opera Software ASA (2003). Opera Web Browser. `http://www.opera.com`.

RealVNC Ltd. (2003). Virtual Network Computing Remote Control Software. http://www.realvnc.com.

Schmitt, M., Hu, J., and Meinel, C. (2003). A Tutoring System for IT Security Education. *Journal of Information Warfare*, 2(3):79–85.

W3C Validator Team (2002). W3C MarkUp Validation Service. http://validator.w3.org.

Woo, C., Choi, J., and Evens, M. (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.

World Wide Web Consortium (1999). HTML 4.01 Specification. W3C Recommendation, http://www.w3.org/TR/html401.

World Wide Web Consortium (2002a). W3C CSS Validation Service. http://jigsaw.w3.org/css-validator.

World Wide Web Consortium (2002b). XHTML 1.0 – The Extensible HyperText Markup Language (Second Edition). W3C Recommendation, http://www.w3.org/TR/xhtml1.