published as: Ji Hu, Christoph Meinel: A Virtual Laboratory for IT Security Education;
Proceedings of EMISA International Conference on Informationssysteme im E-Business und E-Government (EMISA 2004);
Luxembourg, 2004, pp. 60-71.

A Virtual Laboratory for IT Security Education

Ji Hu, Dirk Cordel, Christoph Meinel

FB IV – Informatik
Universitaet Trier
D-54286 Trier, Germany
{hu, cordel, meinel}@ti.uni-trier.de

Abstract: Success of IT security not only depends on the evolution of technologies, but also relies on the knowledge of IT-Personnel and the level of their IT-security education. In order to deliver IT students and professionals hands-on experience of security technologies and tools by electronic means, this paper proposes the concept of a virtual laboratory and discusses its application in practical IT security education. Providing security experience in traditional ways is difficult because it needs dedicated test-bed networks and a lot of administrative efforts, which makes on-site training expensive. The virtual laboratory is an online security laboratory which is built with virtual machines. Virtual machines are software applications which simulate real machines on a host. Equipped with rich security tools and network interfaces, those virtual machines can be assigned to users as laboratory platforms. For that virtual machines are under monitoring and administration, the laboratory can be managed in a reliable way. This makes the running of a virtual laboratory on the Internet possible. Experience within the TeleLab "IT-Security" project successfully proves that the concept of virtual laboratories effectively eliminates geographical and financial limitations in traditional IT Security education.

1 Introduction

Success of IT security not only depends on the evolution of technologies, but also relies on the knowledge of IT-Personnel and the level of their IT-security education. Unfortunately, many attacks succeed because people have no knowledge about vulnerabilities of their systems and do not know how to defend them against attacks. Therefore, more and more universities provide IT security courses and develop dedicated security laboratories for passing students hands-on security experience. In this connection, besides learning theoretical facts, students must exercise security on real-world systems, use real-world tools and resolve practical problems.

In Proceedings of the Conference on Information Systems in E-Business and E-Government (EMISA), Luxembourg, 6-8 Oct 2004, pp. 60-71

However, compared with other topics in software teaching, providing security experience has been found particularly difficult by conventional means. Firstly, dedicated laboratories are needed. This introduces big administrative problems, e.g. preparation for exercises needs many efforts to install systems and to prepare security tools. Secondly, students might frequently cause system errors because super-user rights have to be given to them in some security tasks. Then, recovery from failures is needed. It is difficult to maintain such an unstable system in practical use. Moreover, due to financial reasons, not many institutions can afford a dedicated test-bed network. Considering security, students might misuse their super-user rights. This leads to serious security risks. The laboratory network has to be physically separated from production networks, which limits its application in a local area and makes on-site training expensive.

In order to facilitate tele-teaching or e-learning activities, we have been researching and developing new methods to integrate security laboratory into the electronic tutoring system. The first experimental system named "e-Learning Platform IT Security" (LPF) [SHM03] has been developed in 2002. It is a computer-based training system which teaches practical security technologies and skills. LPF provides practical exercises which help students be familiar with the application of technologies and tools. It runs on a standalone Linux machine which is equipped with various open-source security tools. We prefer these tools because they are open and free and easier for students to understand underlying technique. LPF offers a small laboratory platform to a user. He or she completes exercises (e.g. cracking passwords) using security tools in Linux. After submission, user's results are evaluated by the tutor server. LPF partly replaces the work of an instructor because those exercises are automatically prepared and evaluated. One problem is that it runs in an unreliable mode, because user errors might frequently corrupt machines. Recovery from failures increases its administrative expense since all installation and configuration must be restored.

In 2003, we developed the Tele-Lab CD [HM04] which integrates the entire LPF system into a portable live CD which can detect common hardware and run a Linux completely without hard-disk installation. By this means, a user can use it on any decent PC. Any errors would not affect hardware and software. Rebooting can restore the system from the CD. Although the Tele-Lab CD is more reliable, it is difficult to integrate big tasks into its exercises because of the space limitation and the nature of local usability.

Our final objective is to support distance education. Therefore, we developed the concept of the *Virtual Laboratory* on IT-Security. The virtual laboratory is an online security laboratory built with virtual machines. Virtual machines are software applications which simulate real machines on a host. They are equipped with rich security tools and network interfaces and therefore can be assigned to users as laboratory platforms over the Internet. Since virtual machines can be easily monitored and managed, the laboratory operates in a reliable way, which makes providing security experience over the Internet possible. The Tele-Lab "IT-Security" is a novel virtual laboratory implemented on the Internet. Its experience indicates the application of the virtual laboratories is helpful to eliminate geographical and financial limitations existing in traditional IT Security education.

The rest of this paper is organized as follows. Section 2 reviews previous developments including the LPF and Tele-Lab CD. The concept and architecture of the virtual laboratory are presented in Section 3. Section 4 describes its implementation in the Tele-Lab "IT-Security" project. Its application and teaching experience are discussed in Section 5. Finally, Section 6 concludes the paper.

2 Previous Developments

The LPF is designed following a web structure [SHM03]. Its architecture and components are illustrated in Fig. 2.1.

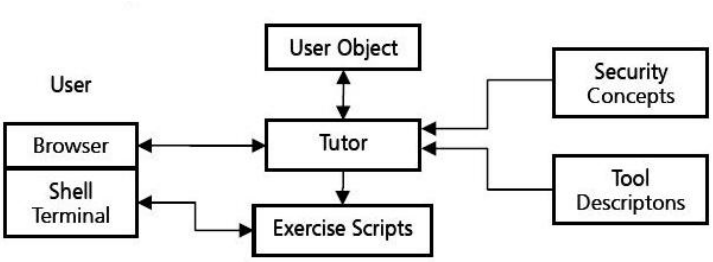

Fig. 2.1 The LPF architecture

The training content includes concepts of a specific IT security topic, descriptions of relevant security tools and exercise scripts written in the PHP (Hypertext Preprocessor) and Perl languages. The user object contains student's personal information and learning records that can be used to analyze a user's performance and to present relevant statistics. The tutor is a web server which presents teaching materials in the form of web pages to the student. It is responsible for execution of scripts to prepare laboratory environments, generate exercise data and evaluate user's results after he/she completes a task. The user interface includes a standard web browser and a Linux shell terminal. User needs to log into the Linux system and to apply security tools using either a text or an X-window based terminal. There are two interactions between the LPF and a user: (1) he/she sends result data to the web server; (2) the user makes some changes in the Linux system, the tutor directly tracks those changes in the local system.

LPF has to assign a user a super-user right in some tasks (e.g. security scanning). This introduces a risk, i.e. a user might misuse his/her right and corrupt the system. In this case, its administrator has to reboot the machine using a backup partition. Recovery interrupts learning processes and frustrates the user's enthusiasm.

A new feature of the Tele-Lab CD is that we re-master a live Linux, the DSL CD [Ds03] with the LPF contents. Thus, system installation and configuration are protected and easy to be restored. The structure of the Tele-Lab CD is illustrated in Fig. 3.2. After the kernel image is loaded from the bootable CD to the memory, the live Linux kernel creates its root file system in the memory. Those directories which are frequently updated (e.g. /tmp, /home, /proc and /etc) are moved to a RAM-disk. Their size is usually small and they can stay in RAM. The read-only directories contain big applications which have a lot of system files, tools, libraries and data (e.g. /bin, /sbin, /lib, /usr, etc.). These directories are imported from the CD via a "cloop" device. The cloop device transparently decompresses the compressed CD file-system on the CD. A large file system (about 1.6 GB) with rich applications can be conveniently provided from a normal CD (700MB). Thus, more space can be reserved for teaching materials. In addition, a live Linux sets up special configurations for detecting devices, loading proper drivers and configuring its X-window server.

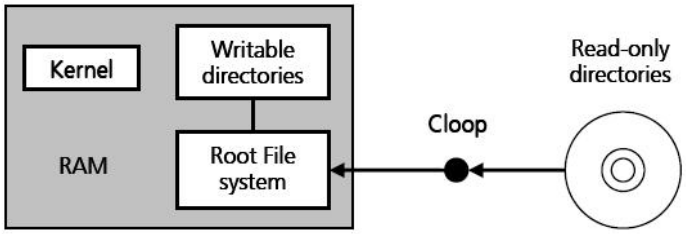

Fig. 2.2 The Tele-Lab CD structure

3. The Virtual Laboratory

The new Tele-Lab "IT-Security" introduces the virtual laboratory concept. The virtual laboratory replaces physical machines with virtual machines on one host server and eliminates the limitation that users have to complete security exercises on the local operating system. Thus, the entire security laboratory and its tutoring server can be moved to the Internet and shared by remote users.

3.1 Virtual Machines

A virtual machine (VM) is a fully protected and isolated copy of the underlying physical machine's hardware. Thus, each user is given the illusion of having a dedicated physical machine [Go74]. Since a virtual machine is a standard software application, multiple virtual machines can be held on a host PC. Moreover, a distinct IP address can be assigned to a virtual machine. Therefore, they can be connected to the network and the Internet. From the user's point of view, a virtual machine can be treated as any normal machine on the network. The destruction of a virtual machine does not result in adverse effects on the underlying host system. Therefore, it is possible to grant super-user rights to an ordinary user.

According to the platform upon which virtual machines are built, virtual machines are classified into two types. The VMs such as IBM's VM/370 [Go74] and VMware [Vm02] are implemented directly on the physical hardware. Then various operating systems can be installed on the virtual machine. The others are built completely on the top of a host operating system. For instance, User-Mode Linux (UML) [Di00] is the open source software that runs a virtual Linux operating system on a host PC. A UML virtual machine is running as processes on the host PC. Its image is contained in an ordinary file. They can be managed like normal applications and can be started, closed or recovered easily.

3.2 Architecture

The system architecture of the virtual laboratory can be divided into three main units according to theirs functionalities, which include a user machine pool, target servers and a control centre. These units are shown in Fig 3.2.1.

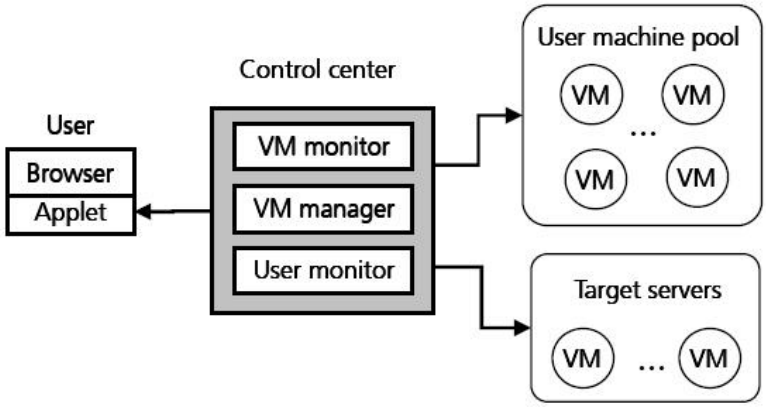

Fig. 3.2.1 Structure of the virtual laboratory

User Machine Pool: The user machine pool runs many VMs as user stations on the host. These VMs are assigned to users when they log in. A VM is installed with a Linux system and open-source security tools. Thus, a user gets an effective working environment in which he/she can complete exercises. In order to improve the performance of the pool, we configure a VM in such a way that its filesystem becomes as small as possible and therefore a VM consumes less computing resources. The filesystem is divided into two parts, a local part and an external part. The local part includes an operating system kernel and a few necessary programs. Large applications and security tools remain in the external filesystem which is imported from the host machine. The external file system is mounted each time a VM is started. Therefore, the size of a VM which we actually manage is decreased, which helps to start, close or recover VMs fast.

For the user interface to the user machine, we integrated a desktop service on individual virtual machines. This service uses a dedicated protocol which collects input from a user, encodes desktop displays on the server side, and sends them to the browser. E.g. the VNC (Virtual Network Computing) [Ri98] applet is a desktop viewer running in a browser. It is able to interact with a VNC server on the virtual machine via the remote frame buffer protocol (RFB) [Ri98]. The advantage of this method is it has a good performance because it compresses desktop data before they are sent and saves a lot bandwidth.

In addition, a remote execution interface, a Secure SHell (SSH) server, is installed on each VM. SSH is a UNIX command interface for securely getting access to a remote host. In this way, the tutor can evaluate a student's results on his/her VM and the control centre can administrate virtual machines in the pool.

Target Servers: Most security exercises can be completed on a virtual machine in the pool. For a few special security exercises, a network environment or target servers are needed. For example, security scanning or attack simulations require a dedicated server which is running with vulnerabilities, otherwise the user can not get an effective result. These target servers are "virtual servers" which are also prepared with virtual machines. They are separated from the user machine pool, because they are subject to corruption in the exercises. In case of a crash detected, the VM manager will recover them in time.

Control Centre: The control centre is a management system located on the host machine. Its purpose is to monitor the run-time status of the laboratory and make it reliable by recovering failures in time. It comprises three functional modules: a VM manager, a VM monitor, a user monitor.

The VM manager is responsible for assigning virtual machines to users, starting, closing or recovering VMs. It maintains a VM assignment table of the user machine pool. This table records the information indicating which VMs have been assigned and for whom, which are free and which are corrupted and need recovery. It reflects the latest status of the virtual laboratory. The VM manager is responsible for updating this table based on the running status of the virtual laboratory. Each active VM has an entry in the table, which stores its IP address, owner and current mode. There are three possible modes: free, assigned and recovered. The free mode means this machine is available for being assigned to a user. If a VM enters the assigned mode, it becomes a dedicated workstation for a user. Its owner item will be filled with the name of its user. The recovered mode represents a transitional mode of a VM. If the defects of a virtual machine are found, this machine is marked as "recovered" and a recovery procedure is called in the background. The processes of the VM are killed and its filesystem is restored. After that, the VM is restarted again and its mode is back to the free mode.

The VM monitor tracks the mode of each active VM and tries to find key errors and report them to the VM manager in time. It periodically scans some services on each VM. These essential services are necessary for a user to work on the VM (e.g. the VNC server), for the tutor to evaluate results (e.g. SSH and Email services). Both connectivity and services are checked, if any does not pass, this VM proves to be failed, and will be recovered in the background.

The user monitor helps prevent unnecessary occupation of a VM. User activities are monitored by detecting user's keyboard and mouse inputs. A detection agent can be inserted into the VNC desktop viewer (a java applet). In this way, the user's idle time on the VM can be measured. If the idle time value is beyond a threshold (e.g. half hour), it seems this user is inactive on the VM, the manager will reclaim his/her VM and let him/her log out.

4. Integrate the Virtual Laboratory into the Tele-Lab "IT-Security"

The Tele-Lab "IT-Security" adopts a similar web structure to the LPF and Tele-Lab CD. The difference is that a virtual laboratory is integrated and some changes are done for transforming Tele-Lab on the Internet.

First, we make changes on the user interface so that a student can access laboratory resource using a standard browser. Tele-Lab "IT-Security" uses a remote desktop access solution, the TightVNC [Ti04], which provides a special Java applet viewer embedded in the browser. It connects to the VNC server on the virtual machine via the RFB protocol. In this way, no additional software is required at the userend as long as the browser supports a Java virtual machine.

The host runs a Linux operating system. The tutor is an Apache web server which is equipped with Perl and PHP interpreters. The teaching contents are organized in a similar way to the LPF except that security exercises are not executed on the local system any longer. The exercise scripts are stripped from the host and moved into the virtual machine. The tutor keeps only an index of those scripts. It will contact the desired virtual machine via the SSH protocol and execute the scripts to evaluate the user's results.

Compared with other VM solutions, the User-Mode Linux is resource-friendly virtual machine software. Therefore, we chose User-Mode Linux to build virtual machines. Its root file system is not bigger than 300 MB. A very compact Linux is installed on the virtual machine. Other applications including X-window based programs and security tools are installed on its external file system which is imported from the host. The original root file-system is protected as ready-only on the host. Each VM is started using a small copy of the root file-system, called the COW (copy-on-write) file. COW files in the UML only record changes to an original image, therefore they are very small and easy to be restored.

VMs are configured as follows. Software needed in exercises is installed and necessary services are configured. For instance, local email services including a SMTP (Simple Mail Transfer Protocol) program (*Exim*) and a POP3 (Post Office Protocol 3) server or an IMAP (Internet Message Access Protocol) server are set up for the Secure E-mail exercise. User accounts on the VM are prepared in advance, his/her email account (e.g. the *Mozilla* email client) data, such as email address, servers and ports, are fixed. A VM is assigned with a private IP address (e.g. 192.168.0.100) which can not be directly accessed from the Internet. A firewall is installed on the host, which only allows an authorized user to access his/her own virtual machine without offending other machines.

The management scripts mentioned in Section 3.2 are Perl scripts on the host system. Some daemons are running in order to monitor services on the VM and user activities periodically. A VM control interface from the UML, called *uml_mconsole,* is integrated in order to manipulate VMs from the host. It provides some useful commands, e.g. "reboot" to restart a VM, "halt" to kill a VM. For recovery, the VM manager immediately kills the failed VM with this console, replaces its COW-file with a new copy and starts it again.

5. Applications and Experience

The Tele-Lab "IT-Security" system can be applied to practical IT security education because its virtual laboratory provides an infrastructure in which security tasks can be designed and tested. Moreover, chapters of the Tele-Lab "IT-Security" are designed as modules which can be flexibly inserted to the system. The contents can be tailored depending on specific teaching purposes and objects, e.g. teaching materials can be customized for administrator users and for common users. Therefore, the Tele-Lab "IT-Security" can be a complete e-learning system or used to support a security course.

The learning process of a security chapter has three phases: first, concepts and principles of a security topic are introduced to a student. Next, we provide the information about typical security tools relevant to the topic. Finally, in the third phase, the student performs exercises with these tools. The completion of each step is recorded into his/her user object so that if a student logs on at next time, he or she is able to continue from the break point.

Essential activities in the learning include user sign-on, user sign-out and exercises. They are described as follows.

User Sign-on: the steps that a user logs into the Tele-Lab include:

- User authenticates to the system.
- The user object is loaded from the user database.
- The tutor server requests a free virtual machine from the control centre.
- The VM manager finds an available VM from the user machine pool and updates its assignment table entry. Related firewall configuration is changed to allow the user to access his/her VM.
- The tutor web server sends a VNC java applet to the user's browser. The desktop on the VM is shown in the user's browser window.

User Sign-out: after a user chooses to log out the Tele-Lab,

- The user object is updated to the database.
- The VNC applet window is closed.
- The web notifies the control centre to reclaim the VM.
- The VM manager recovers the used VM with a fresh copy and starts it again, then, the assignment table is updated.
- The firewall configuration is updated to reclaim the user's access permission to the VM.

Exercises: typically, an exercise task is finished through three phases:

- The working environment on the VM is configured, e.g. for the secure email task, an account of a virtual partner is created so that email interaction is possible between two sides.

- Scripts for preparation of tasks are activated and questions are shown on web pages. If possible, these questions are dynamically generated. E.g. in the password cracking exercise, a UNIX *passwd* file, is generated at run-time. The interface for performing exercises is shown in Figure 5.1.

- After the user has completed the tasks and submitted his results, the tutor evaluates those results. In case of failures, help pages provide information about possible solutions to the problem. In this way, a student can repeat an exercise until a correct solution is finally found.

Fig. 5.1 User interface of the Tele-Lab "IT-Security"

Beside the processes mentioned above, the control centre is responsible for managing unusual events. For example, user might close a web session without sign-out, user has been idle for a long period, or a virtual machine crashes. Since we have the VM monitor and user monitor which observe the running status of the virtual laboratory, failed virtual machines can be detected and be recovered in time.

After we finished the demo of the Tele-Lab "IT-Security", tests were made by the students from the junior grade in our department. Those students did not take security courses before. The feedback from them shows they can understand theoretical parts and complete exercises without help of a human instructor. If students are unfamiliar with Linux, the Tele-Lab will provide necessary help information in exercises. However, it is easier for those who know Linux. The results indicate that the use of Linux as a laboratory platform might bring some difficulties but it is not a barrier in learning. We have to point out that virtual security laboratories can not completely replace traditional dedicated security laboratories. Exercises which need a complicated network environment or long run-time are difficult to implement. Nevertheless, virtual laboratories are an effective alternative to run practical security exercises on the Internet in an economic and automated way.

6. Conclusion

In this paper, we propose a virtual laboratory concept and discuss its application in practical IT security education. This work improves the existing security education activities in the following aspects.

- It offers a virtual security laboratory to users over the Internet, instead of a limited simulation or an expensive dedicated laboratory. The virtual laboratory is built with virtual machines which are equipped with real security tools. A user can perform security exercises remotely without managing any software installation and configuration.

- It is a reliable tutoring system. In the virtual laboratory, privilege rights can be safely assigned to students for security tasks. Failures on a virtual machine would not affect the host system and can be detected and recovered in time.

- A thin user interface is implemented. At the client side, a student only needs a standard web browser. An embedded JAVA applet desktop viewer in the browser allows the student to access the virtual laboratory.

- It builds an online security laboratory with low software and hardware costs. Also, administrative work for preparation, configuration and recovery is carried out automatically, which degrades its management expense.

Experience of the Tele-Lab "IT-Security" project proves that virtual laboratories are very helpful to eliminate geographical and financial limitations in traditional IT security education.

Acknowledgements

We would like to acknowledge Dr. Michael Schmitt for his contribution to the Tele-Lab "IT-Security" concept. And, also Gennadi Umanski for his helpful discussions. We would like to acknowledge our student members for their contribution to the implementation.

Reference

[Di00] Dike, J.: A User-Mode Port of the Linux Kernel. In: Proc Conf. of Annual Linux Showcase, Atlanta, GA, 2000.

[Ds03] DSL Linux: Available from http://damnsmalllinux.org/, 2004.

[Go74] Goldberg, R. P.: Survey of Virtual Machine Research. In: IEEE Computer, June 1974; pp 34-45.

[HM04] Hu, J.; Meinel, Ch.: Tele-Lab "IT-Security" on CD: Portable, Reliable and Safe IT Security Training, In: Computers & Security, Elsevier, Volume 23, Issue 4, Elsevier, June 2004, pp. 282-289.

[SHM03] Schmitt, M.; Hu, J.; Meinel, Ch.: A Tutoring System for IT Security Education. In: Journal of Information Warfare, Vol.2, Issue 3, 2003, pp. 79-85.

[Ri98] Richardson, T. et. al.: Virtual Network Computing. In: IEEE Internet Computing, Jan/Feb 1998; pp. 33-38.

[Ti04] TightVNC: Available from http://www.tightvnc.com/, 2004.

[Vm02] VMware Inc.: VMware GSX Server. Available from http://www.vmware.com/, 2002.