CEPIS UPGRADE is the European Journal for the Informatics Professional, published bi-monthly at <http://cepis.org/upgrade>

Publisher
CEPIS UPGRADE is published by CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>), in cooperation with the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>) and its journal Novática

CEPIS UPGRADE monographs are published jointly with Novática, that publishes them in Spanish (full version printed; summary, abstracts and some articles online)

CEPIS UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies)

CEPIS UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- inforewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <info@cepis.org>

Subscriptions
If you wish to subscribe to CEPIS UPGRADE please send an email to info@cepis.org with 'Subscribe to UPGRADE' as the subject of the email or follow the link 'Subscribe to UPGRADE' at <http://www.cepis.org/upgrade>

ISSN 1684-5285

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade

Vol. XII, issue No. 5, December 2011

Farewell Edition

./..

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade
Vol. XII, issue No. 5, December 2011

Farewell Edition

Cont.

UPENET (UPGRADE European NETwork)

CEPIS NEWS

IT Security

Practical IT Security Education with Tele-Lab

Christian Willems, Orestis Tringides, and Christoph Meine

This paper was first published, in English, by ***Pliroforiki*** (issue no. 21, July 2011, pp. 30-38). ***Pliroforiki***, ("Informatics" in Greek), a founding member of **UP**ENET, is a journal published, in Greek or English, by the Cyprus CEPIS society CCS (Cyprus Computer Society, <http://www.ccs.org.cy/about/>) . The July 2011 issue is available at <http://www.pliroforiki.org/>.

The rapid burst of Internet usage and the corresponding growth of security risks and online attacks for the everyday user or the enterprise employee have emerged the terms Awareness Creation and Information Security Culture. Nevertheless, security education has remained widely an academic issue. Teaching system security or network security on the basis of practical experience inherits a great challenge for the teaching environment, which is traditionally solved using a computer laboratory at a university campus. The Tele-Lab project offers a system for hands-on IT security training in a remote virtual lab environment – on the web, accessible at any time.

Keywords: Information Security, IT Security Training, Online Attacks, Security Risks, Tele-Lab Project, Virtual Lab Environment.

Introduction

Increasing propagation of complex IT systems and rapid growth of the Internet more and more draws attention to the importance of IT security issues. Technical security solutions cannot completely overcome the lacking awareness of computer users, caused by indifference or laziness, inattentiveness, and lack of knowledge and education. In the context of awareness creation, IT security training has become a topic of strong interest – as well as for companies as for individuals.

Traditional techniques of teaching (i.e. lectures or literature) have turned out to be not suitable for IT security training, because the trainee cannot apply the principles from the academic approach to a realistic environment within the class. In IT security training, gaining practical experience through exercises is indispensable for consolidating the knowledge. Precisely the allocation of an environment for these practical exercises poses a challenge for research and development. That is, because students need privileged access rights (root/administrator account) on the training system to perform most of the perceivable security exercises. With these privileges, students could easily destroy a training system or even use it for unintended, illegal attacks on other hosts within the campus network or on the Internet.

Authors

Christian Willems studied computer science at the University of Trier, Germany, and received his diploma degree in 2006. Currently he is research assistant at the Hasso-Plattner-Institute for IT Systems Engineering, giving courses on internet technologies and security. Besides that he is working on his PhD thesis at the chair of Prof. Dr. Christoph Meinel. His special research interests focus on Awareness Creation, IT security teaching and virtualization technology. <christian.willems@hpi.uni-potsdam.de>

Orestis Tringides is the Managing Director of Amalgama Information Management and has participated in research projects since 2003 in the areas of e-learning, e-business, ICT Security and the Right of Access to Information. He holds a B.Sc. degree in Computer Science, an M.Sc. degree in Information Systems and is currently pursuing an MBA degree. He also participates in Civil Society projects and is interested in elderly care, historical remembrance, soft tourism and social inclusion. <orestis@tringides.com>

Christoph Meinel is scientific director and CEO of the Hasso-Plattner-Institute for IT Systems Engineering and professor for computer science at the University of Potsdam, Germany. His research field is Internet and Web Technologies and Systems. Prof. Dr. Meinel is author or co-author of 10 text books and monographs and of various conference proceedings. He has published more than 350 per-reviewed scientific papers in highly recognised international scientific journals and conferences. <meinel@hpi.uni-potsdam.de>

> “Security education has remained widely an academic issue”

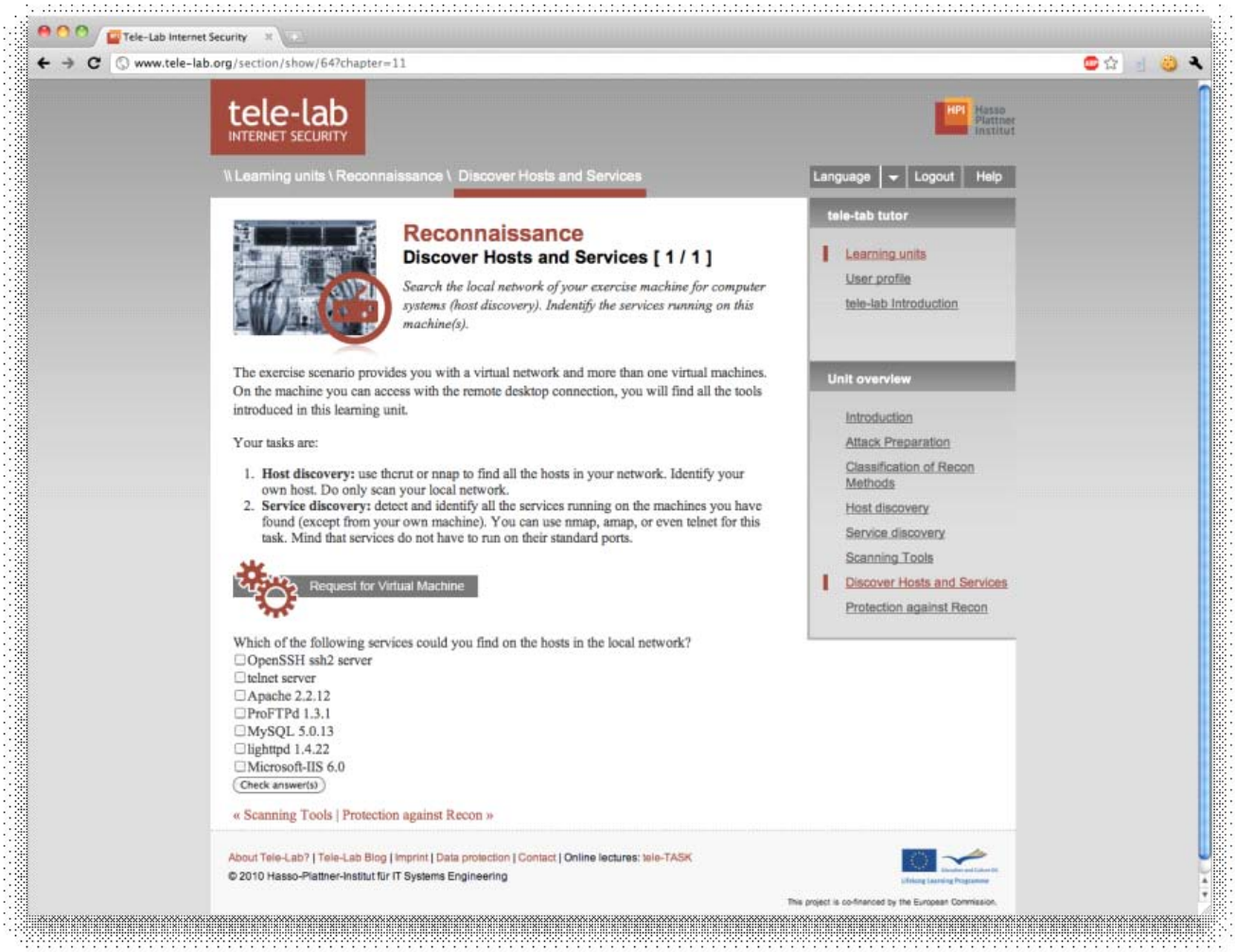

Figure 1: Screenshot of the Tele-Lab Tutoring Interface

The classical approach requires a dedicated computer lab for IT security training. Such labs are exposed to a number of drawbacks: they are immobile, expensive to purchase and maintain and must be isolated from all other networks on the site. Of course, students are not allowed to have Internet access on the lab computers. Hands-on exercises on network security topics even demand to provide more than one machine to each student, which have to be interconnected (i.e. a Man-in-the-Middle attack needs three computers: one for the attacker and two other machines as victims).

Teleteaching for security education mostly consists of multimedia courseware or demonstration software, which do not offer real practical exercises. In simulation systems users do have a kind of hands-on experience, but a simulator doesn't behave like a realistic environment and the simulation of complex systems is very difficult – especially when it comes to interacting hosts on a network. The Tele-Lab project builds on a different approach for a Web-based teleteaching system (explained in detail in section 2).

Furthermore, we will describe a set of exercise scenarios to illustrate the capabilities of the Tele-Lab training environment: a simple learning unit on password security, an exercise on eavesdropping, and the practical application of a Man-in-the-Middle attack.

Tele-Lab: A Remote Virtual Security Laboratory

Tele-Lab, accessible at <http://www.tele-lab.org>, was first proposed as a standalone system [4], later enhanced to a live DVD system introducing virtual machines for the hands-on training [3], and then emerged to the Tele-Lab server [2, 6]. The Tele-Lab server provides a novel e-learning sys-

> "In IT security training, gaining practical experience through exercises is indispensable for consolidating the knowledge"

> “The classical approach requires a dedicated computer lab for IT security training”

tem for practical security training in the WWW and inherits all positive characteristics from offline security labs. It basically consists of a web-based system (see Fig. 1) and a training environment built of virtual machines. The tutoring system provides learning units with three types of content: information chapters, introductions to security- and hacker tools and finally practical exercises. Students perform those exercises on virtual machines (VM) on the server, which they operate via remote desktop access. A virtual machine is a software system that provides a runtime environment for operating systems. Such software-emulated computer systems allow easy deployment and recovery in case of failure. Tele-Lab uses this feature to revert the virtual machines to the original state after each usage. This is a significant advantage over the traditional setting of a physical dedicated lab, since the recovery to the original state can be performed quicker, more often and without any manual maintenance efforts.

With the release of the current Tele-Lab 2.0, the platform introduced the dynamic assignment of several virtual machines to a single user at the same time. Those machines are connected within a virtual network (known as team, see also in [1]) providing the possibility to perform complex network attacks such as Man-in-the-Middle or interaction with a virtual (scripted) victim (see exemplary description of a learning unit below).

A short overview of the Tele-Lab architecture is given later in this section.

A Learning Unit in Tele-Lab

An exemplary Tele-Lab learning unit on malware (described in more detail in [5]) starts off with academic knowledge such as definition, classification, and history of malware (worms, viruses, and Trojan horses). Methods to avoid becoming a victim and relevant software solutions against malware (e.g. scanners, firewalls) are also presented. Afterwards, various existing malware kits and ways for distribution are described in order to prepare the hands-on exercise. Following an offensive teaching approach[1], the user is asked to take the attacker’s perspective – and hence is able to lively experience possible threats to his/her personal security objectives, as if physical live systems were used. The closing exercise for this learning unit on malware is to plant a Trojan horse on a scripted victim called Alice – in particular, the Trojan horse is the out-

[1] See [9] for different teaching approaches.

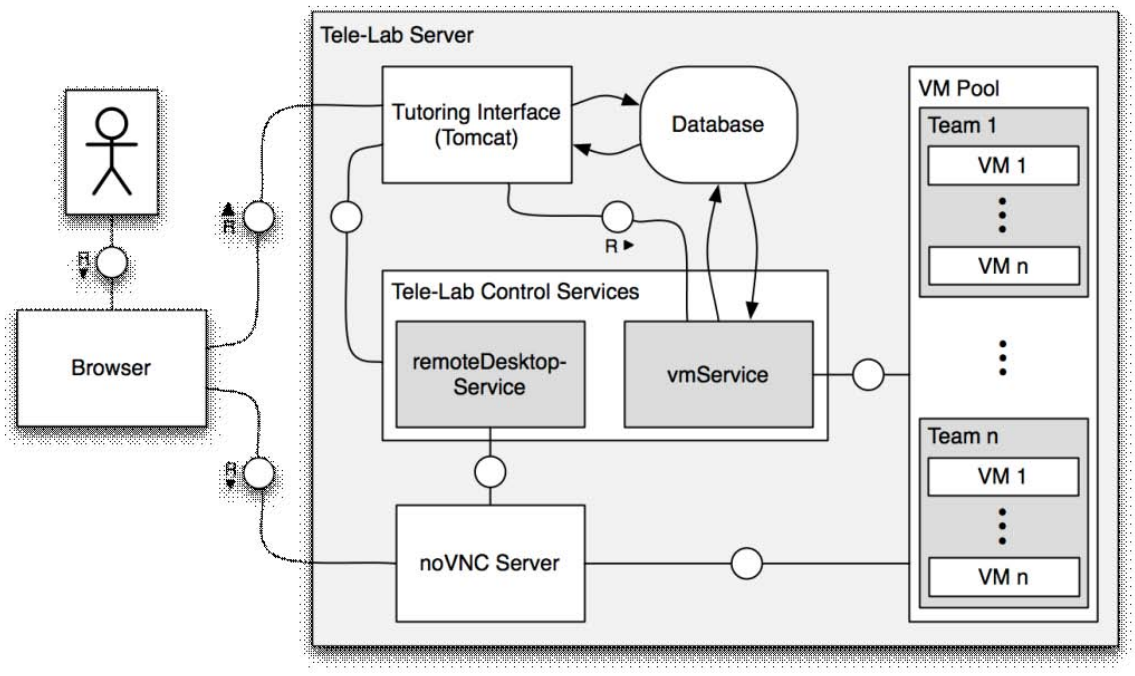

Figure 2: Architecture of the Tele-Lab Platform.

> “The Tele-Lab project builds on a different approach for a Web-based teleteaching system”

dated Back Orifice[2]. In order to achieve that, the student has to prepare a carrier for the BO server component and send it to Alice via e-mail. The script on the victim VM will reply by sending back an e-mail,indicating that the Trojan horse server has been installed (that the e-mail attachment has been opened by the victim). The student can now use the BO client to take control of the victim's system and spy out some private information. The knowledge of that information is the user's proof to the Tele-Lab tutoring environment, that the exercise has been successfully solved.

Such an exercise implies the need for the Tele-Lab user to be provided with a team of interconnected virtual machines: one for attacking (with all necessary tools pre-installed), a mail server for e-mail exchange with the victim and a vulnerable victim system (in this particular case, an unpatched Windows 95/98). Remote Desktop Access is only possible to the attacker's VM.

Learning units are also available on e.g. authentication, wireless networks, secure e-mail, etc. The system can easily be enhanced with new content. For example, in a project participating the Hasso-Plattner-Institute, the Vilnius Gediminas Technical University (VGTU), nSoft and Amalgama Information Management Ltd., new learning units were easily added to the VGTU implementation of Tele-Lab, <http://telelab.vgtu.lt>, and have been shared among partners. The content was translated for Lithuanian language localization. For the future, the project consortium plans to add more learning units and expand localization for the Greek language.

Architecture of the Tele-Lab Server

The current architecture of the Tele-Lab 2.0 server is a refactored enhancement to the infrastructure presented in [6]. Basically, it consists of the following components (illustrated in Fig. 2).

Portal and Tutoring Environment: The Web-based training system of Tele-Lab is a custom Grails[3] application running on a Tomcat application server. This web application handles user authentication, allows navigation through learning units, delivers their content and keeps track of the students' progress. It also provides controls to request a team of virtual machines for performing an exercise. The *Portal and Tutoring Environment* (along with the *Database* and *Administration Interface* components described later on) offer tutors and students facilities of a Learning Management System, such as centralized and automated administration, assembly and delivery of learning content, reuse of the learning units, etc. [11]

Virtual Machine Pool: The server is loaded with a set of different virtual machines needed for the exercise scenarios – the pool. The resources of the physical server limit the maximum total number of VMs in the pool. In practice, a few (3-5) machines of every kind are started up. Those machines are dynamically connected to teams and bound to a user on request. The current hypervisor solution used to provide the virtual machines is KVM/Qemu[4]. The way virtual machines are used in Tele-Lab's architecture, allow for further creative ways to allocate resources in an optimized and collaborative manner, by setting collaboration among different instances of the Tele-Lab system that are installed on different sites: in the example of the abovementioned consortium, HPI's and VGTU's Tele-Lab servers share resources in order to dynamically provide virtual machines to each other, when needed. For example: if a student from VGTU requests to conduct a laboratory exercise, but the VGTU's Tele-Lab server has already reached the maximum limit of VMs that can be allocated, it automatically requests HPI's Tele-Lab server to allocate a VM from its own resources (and vice versa). This automatic process occurs seamlessly, so the user does not experience any disruptions. In the future, this collaboration arrangement can easily be expanded into a grid of different institutions, sharing their Tele-Lab server's resources to each other, thus evenly distributing the whole process workload, when e.g. there is a peak in VM demand at one of the partners' site.

> “Students perform those exercises on virtual machines (VM) on the server, which they operate via remote desktop access”

[2] BackOrifice (BO) is a Remote Access Trojan Horse developed by the hacker group „Cult of the Dead Cow", see <http://www.cultdeadcow.com/tools/bo.php>.

[3] Grails is an open-source frame work for web application development, see <http://www.grails.org/>.

[4] See <http://www.linux-kvm.org/ and http://www.qemu.org>/.

For the *network connections within the teams*, Tele-Lab uses the Virtual Distributed Ethernet (VDE)[5] package. VDE emulates all physical aspects of Ethernet LANs, in software. The *Tele-Lab Control Services* launch virtual switches or hubs for each virtual network defined for a team of VMs and connect the machines to the appropriate network infrastructure. For the distribution of IP addresses in the virtual networks, a DHCP server is attached to every network. After sending out all leases, the DHCP server is killed due to security constraints. [7]

Database: The Tele-Lab database holds all user information, the content for web-based training and learning unit structure as well as the information on virtual machine and team templates. A VM template is the description of a VM disk image that can be cloned in order to get more VMs of that type. Team templates are models for connected VMs that are used to perform certain exercises. The database also persists current virtual machine states.

Remote Desktop Access Proxy: The Tele-Lab server must handle concurrent remote desktop connections for users performing exercises. This is realized using the open-source project *noVNC*[6], a client for the Virtual Network Computing Protocol based on *HTML5 Canvas* and *WebSockets*. The noVNC package comes with the HTML5 client and a WebSockets proxy which connects the clients to the VNC servers provided by QEMU. Ensuring a protected environment for both the Tele-Lab users and system is a challenge that is important to thoroughly implement at all levels, as the issue of network security for virtual machines in a Cloud Computing setting (such as the case of Tele-Lab) poses special requirements. [8] The system uses a token-based authentication system: an access token for a remote desktop connection is generated, whenever a user requests a virtual machine team for performing an exercise. Using TLS ensures the confidentiality of the token.

Administration Interface: The Tele-Lab server comes with a sophisticated web-based administration interface that is also implemented as a Grails application (not depicted in Fig. 2). On the one hand, this interface is made for content management in the web-based training environment and on the other, for user management. Additionally, the admin interface can be used for manual virtual machine control, monitoring and for registering a new virtual machine or team templates.

Tele-Lab Control Services: The purpose of the central Tele-Lab control services is bringing all the above components together. To realize an abstraction layer for encapsulation of the virtual machine monitor (or hypervisor) and the remote desktop proxy, the system implements a number of lightweight XML-RPC web services. The *vmService* is for controlling virtual machines – start, stop or recover them, grouping teams or assigning machines or teams to a user. The *remoteDesktopService* is used to initialize, start, control and terminate remote desktop connections to assigned machines. The above-mentioned Grails applications (portal, tutoring environment, and web admin) allow the user to control the whole system using the web services.

On the client side, the user only needs a web browser supporting SSL/TLS. The current implementation of the noVNC client does not even need an HTML5-capable browser: for older browsers, HTML5 Canvas and/or the WebSockets are emulated using Adobe Flash.

IT Security Exercises

As stated before, one of the strengths of Tele-Lab (and other isolated laboratories) is the ability to provide secure training environments for exercises, where the student takes the perspective of an attacker. Next to the learning unit on Trojan horses presented in chapter 2, we introduce a set of additional exercise scenarios to illustrate this approach: Attacks on Accounts and Passwords, Eavesdropping of Network Traffic, and a Man-in-the-Middle Attack.

Exercise Scenario A: Attacks on Accounts and Passwords

Gaining valid user credentials for a computer system is obviously major objective for any attacker. Hackers can get access to personal and confidential data or use a valid login as a starting point for numerous further attacks, such as gaining privileged access to their target system.

It is well known that one should set a password consisting of letters (upper and lower case), numbers and special characters. Moreover, the longer a password is, the harder it is to crack. Thus, it is inherently important for a user to choose strong credentials – even though passwords of high complexity are harder to memorize.

Studies[7] show, that users still choose very weak passwords, if allowed so. In December 2009, a hacker stole passwords from the popular online platform rockyou.com and released a dataset of 32 million passwords to the Internet[8]. An analysis of those passwords revealed several interesting findings:

- 30% of the users chose passwords with a length of 6 characters or less, 50% had a password not longer than 7 characters
- Almost 60% of the users chose their password from a limited set of alphanumeric characters
- Nearly 50% used names, slang words, dictionary words or trivial passwords (consecutive digits, adjacent keyboard keys, and so on)

The learning unit on Password Security explains how passwords are stored within computer systems (i.e. password hashes in Linux), and how tools like *Password Sniffers, Dumpers* and *Crackers* work.

[5] See <http://vde.sourceforge.net/>.
[6] See <http://kanaka.github.com/noVNC/>.
[7] See i.e. <http://www.rsa.com/solutions/consumer_authentication/reports/9381_Aberdeen_Strong_User_Authentication.pdf>.
[8] See <http://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/>.

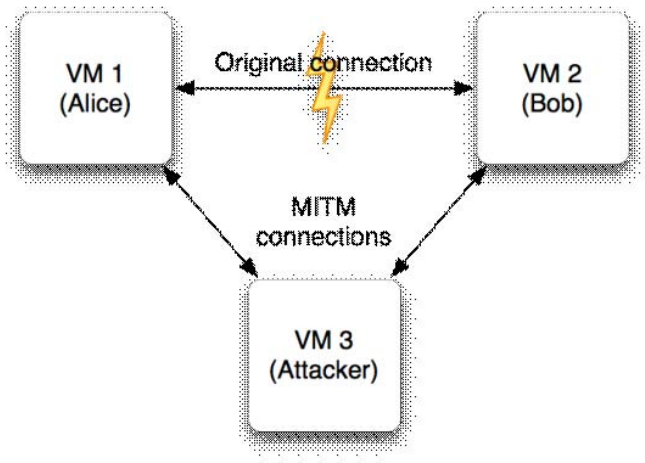

Figure 3: Man-in-the-Middle Attacks.

In the exercise section, the user is asked to experience how fast weak passwords can be cracked. On the training machine (Windows XP) the user must dump the passwords to a file using *PwDump*, and crack the hashes with the well-known *John-the-Ripper*[9] password recovery tool. It gets obvious, that passwords like the username or words from dictionaries usually can be cracked within a few seconds.

The learning unit concludes with hints, how to choose a strong password that can be memorized easily.

Exercise Scenario B: Eavesdropping of Network Traffic

The general idea of eavesdropping is to secretly listen to the private communication of two (or more) communication partners without their consent. In the domain of computer networks, the common technique for eavesdropping is *packet sniffing*. There are a number of tools for packet sniffing – *packet analyzers* – freely available on the Internet, such as the well-known *tcpdump* or *Wireshark*[10] (used in this learning unit).

A learning unit on packet sniffing in a local network starts with an introduction to communication on the data-link layer (Ethernet) and explains the difference between a network with a hub and a network in a switched environment.

This is important for eavesdropping, because this kind of attack is much easier when connected to a hub. The hub will forward every packet coming in to all its ports and hence to all connected computers. These hosts decide if they accept and further compute the incoming data based on the MAC address in the destination field of the Ethernet frame header: if the destination MAC is their own MAC address, the Ethernet frame is accepted, or dropped otherwise. If there is a packet analyzer running, also frames not intended for the respective host can be captured, stored and analyzed. This situation is different in a switched network: the switch does not broadcast incoming data to all ports but interprets the MAC destination to "switch" a dedicated line between source and destination ports. In consequence, the Ethernet frame is only delivered to the actual receiver.

After this general information on Ethernet-based networking, the learning unit introduces the idea of packet sniffing and describes capabilities and usage of the packet analyzer *Wireshark*, especially on how to capture data from the Ethernet device and how to filter and read the captured data.

The practical exercise presents the following task to the learner: *"Sniff and analyze network traffic on the local network. Identify login credentials and use them to obtain a private document."* The student is challenged to enter the content of this private document to proof, that he/she has solved the task.

When requesting access to a training environment, the user is assigned to a team of three virtual machines: the attacker machine that is equipped with the Wireshark tool, and two machines of (scripted) communication partners: Alice and Bob. In this scenario, Bob's machine hosts an FTP server and a Web server, while Alice's VM runs a script that generates traffic by initiating arbitrary connections to the services on Bob's host. Among those client/server connections are successful logins to Bob's FTP server. As this learning unit focuses on sniffing and the interpretation of the captured traffic of the machines are connected with a hub. There is no need for the attacker to get into a Man-in-the-Middle position in order to capture the traffic between Alice and Bob.

[9] See <http://www.openwall.com/john> for information on John-the-Ripper, <http://www.foofus.net/~fizzgig/pwdump/> for PwDump6.
[10] See <http://www.wireshark.org/>.

> "With Tele-Lab 2.0, the platform introduced the dynamic assignment of several virtual machines to a single user at the same time"

> “The continuous evolvement of the issue of IT security demands for a constant updating the curriculum with new learning units”

Since FTP does not encrypt credentials, the student can obtain username and password to log in to that service. On the server, the student finds a file called *private.txt* that contains the response to the challenge mentioned above.

The section concludes with hints on preventing eavesdropping attacks, such as the usage of services with secure authentication methods (i.e. SFTP or ftps instead of plain FTP) and data encryption.

Exercise Scenario C: Man-in-the-Middle Attack with ARP Spoofing

The general idea of a Man-in-the-Middle attack (MITM) is to intercept communication between two communi-cation partners (Alice and Bob) by initiating connections between the attacker and both victims and spoofing the identity of the respective communication partner (Fig. 3). More specifically, the attacker pretends to be Bob and opens a connection to Alice (and vice versa). All traffic between Alice and Bob is being relayed via the attacker's computer. While relaying, the messages can be captured and/or manipulated.

MITM attacks can be implemented on different layers of the TCP/IP network stack, i.e. *DNS cache poisoning* on the application layer, *ICMP redirecting* on the internet layer or *ARP spoofing* in the data-link layer. This learning unit focuses on the last-mentioned attack, which is also called *ARP cache poisoning*.

The Address Resolution Protocol (ARP) is responsible for resolving IP addresses to MAC addresses in a local network. When Alice's computer opens an IP-based connection to Bob's computer in the local network, it has to determine Bob's MAC address at first, since all messages in the LAN are transmitted via the Ethernet protocol (which is only aware about the MAC addresses). If Alice only knows the IP address of Bob's host, (i.e. 192.168.0.10) she performs an *ARP request*: Alice sends a broadcast message to the local network and asks, *"Who has the IP address 192.168.0.10?"* Bob's computer answers with an *ARP reply* that contains its IP address and the corresponding MAC address. Alice stores that address mapping in her *ARP cache* for further communication.

ARP spoofing [10] is basically about sending forged ARP replies: referring to the above example, the attacker repeatedly sends ARP replies to Alice with Bob's IP address and MAC address – the attacker pretends to be Bob. When Alice starts to communicate with Bob, she sends the ARP request and instantly receives one of the forged ARP replies from the attacker. She then mistakenly thinks that the attacker's MAC address belongs to Bob and stores the faked mapping in her ARP cache. Since the attacker performs the same operation for Alice's MAC address, he/she can also manage to trick Bob, that his/her MAC address is the one of Alice. In consequence, Alice sends all messages to Bob to the MAC address of the attacker (and the same applies for Bob's messages to Alice). The attacker just has to store the original MAC addresses of Alice and Bob to be able to relay to the original receiver.

A learning unit on ARP spoofing begins with general information on communication in a local network. It explains the Internet Protocol (IP), ARP and Ethernet including the relationship between the two addressing schemes (IP and MAC addresses).

Subsequently, the above attack is described in detail and a tool, that implements ARP spoofing and a number of additional MITM attacks is presented: *Ettercap*[11]. At this point, the learning unit also explains what the attacker can do, if he/she becomes Man-in-the-Middle successfully, such as specifying *Ettercap filters* to manipulate the message stream.

The hands-on exercise of this chapter asks the student to perform two different tasks. The first one is the same as described in the exercise on packet sniffing above: to monitor the network traffic, gain FTP credentials and steal a private file from Bob's FTP server. The training environment is also set up similarly to the prior scenario. The difference is that this time the team of three virtual machines is connected through a virtual switch (instead of a hub), so that capturing the traffic with Wireshark would not reveal the messages between Alice and Bob. Again, the student has to proof the successful attack by putting in the content of the secret file in the tutoring interface.

The second (optional) task is to apply a filter on the traffic and replace all images in the transmitted HTML content by an image from the attacker's host (which would be displayed in Alice's browser).

This kind of attack is still working and dangerous in many currently deployed local network installations. The only way to protect oneself against ARP spoofing would be the usage of SSL with a careful verification of the host's certificate, which is explained in the conclusion of the learning unit.

A future enhancement of the practical exercise on ARP spoofing would be the interception of an SSL secured channel: Ettercap also allows a more sophisticated MITM attack including the on-the-fly generation of faked SSL certificates, which are presented to the victims instead of the original ones. The Man-in-the-Middle can then decrypt and re-encrypt the SSL traffic when relaying the messages.

[11] See <http://ettercap.sourceforge.net/>.

> “The Tele-Lab consortium partners share knowledge, development tasks, functionalities, new curriculum content and resources”

Outlook and Conclusion

The Tele-lab system has been developed in order to attend to the particular challenges and needs posed in IT security training and IT security laboratory settings. First of all, it is essential for an IT security course to be able to provide real hands-on experience to the learners, by using the necessary systems and contemporary IT security tools. For this, the use of virtual machines is an obvious approach in order to, on one hand, deliver realistic hands-on exercises to the learners and on the other hand, to isolate such exercises from the "real" network infrastructure of the training provider.

The continuously increasing importance of the issue of IT security, as it is presented everyday in the mass media, and the very serious negative repercussions it can bring nowadays, pushes for more awareness and a more imperative need for IT security knowledge and practical skills. Academic institutions and training providers need to provide such training that is in the state of the art, however, constructing an IT Security training environment (i.e., a computer laboratory devoted to IT security training) requires knowledge, a considerable upfront investment for acquisition, costs for administration and maintenance) and poses risks when there are omissions in properly insulating such physical laboratories from the rest of the network infrastructure. Tele-Lab mitigates those difficulties by providing a fairly cheaper solution, that adds up to nearly no effort at all for maintenance and administration.

More important, the continuous evolvement of the issue of IT security (that evolves in parallel to, and perplexes with, all innovations in ICT) demands for a constant updating the curriculum with new learning units, or update existing learning units with new perplexing factors. Although Tele-Lab provides the facilities for easy addition of new learning units and exercises, the big feat of updating the knowledge base can be achieved by collaboration of different institutions that are using Tele-Lab, and sharing amongst them the new learning units and newly constructed system functionalities. Also sharing resources (e.g. Virtual Machines) in order to even the systems workload is a valuable outcome of such cooperations. In the example of the project consortium mentioned in section 2 and 3, such arrangements have already been put in place, and the consortium partners share knowledge, development tasks, functionalities, new curriculum content and resources.

It is a challenge to prove that Tele-Lab, in combination with such a collaborative and evolving model of cooperation among networks of institutions, can achieve delivering an innovative and always updated course of high standards, that can address the difficulties faced in modern IT security training.

References

[1] C. Border. The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes. SIGCSE Bulletin, 39(1): 576–580, 2007.

[2] J. Hu, D. Cordel, and C. Meinel. A Virtual Machine Architecture for Creating IT-Security Laboratories. Technical report, Hasso-Plattner-Insitut, 2006.

[3] J. Hu and C. Meinel. Tele-Lab IT-Security on CD: Portable, reliable and safe IT security training. Computers & Security, 23:282–289, 2004.

[4] J. Hu, M. Schmitt, C. Willems, and C. Meinel. A tutoring system for IT-Security. In Proceedings of the 3rd World Conference in Information Security Education, pages 51–60, Monterey, USA, 2003.

[5] C. Willems and C. Meinel. Awareness Creation mit Tele-Lab IT-Security: Praktisches Sicherheit straining im virtuellen Labor am Beispiel Trojanischer Pferde. In Proceedings of Sicherheit 2008, pages 513–532, Saarbrücken, Germany, 2008.

[6] C. Willems and C. Meinel. Tele-Lab IT-Security: an Architecture for an online virtual IT Security Lab. International Journal of Online Engineering (iJOE), X, 2008.

[7] C. Willems and C. Meinel, Practical Network Security Teaching in a Virtual Laboratory. In Proceedings of Security and Management 2011, Las Vegas, USA, 2011 (to appear).

[8] C. Willems, W. Dawoud, T. Klingbeil, and C. Meinel. Security in Tele-Lab – Protecting an Online Virtual Lab for IT Security Training, In Proceedings of ELS’09 (in conjunction with 4th ICITST), IEEE Press, London, UK, 2009.

[9] W. Yurcik and D. Doss. Different approaches in the teaching of information systems security. In Security, Proceedings of the Information Systems Education Conference, pages 32–33, 2001.

[10] S. Whalen. An Introduction to ARP Spoofing. Online: <http://www.rootsecure.net/content/downloads/pdf/arp_spoofing_intro.pdf>.

[11] J. Bersin, C. Howard, K. O’Leonard, and D. Mallon. Learning Management Systems 2009, Technical Report, Bersin & Associates, 2009. Online: <http://www.bersin.com/Lib/Rs/Details.aspx?docid=10339576>.

