Virtualizing an IT Lab for Higher Education Teaching

Nils gentschen Felde, Tobias Lindinger
Munich Network Management Team
Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 Munich, Germany
{felde|lindinge}@mnm-team.org

Helmut Reiser
Munich Network Management Team
Leibniz Computing Centre
Boltzmannstr. 1, 85748 Garching, Germany
reiser@mnm-team.org

Abstract

In universities, a great amount of time is needed to manage and operate lab course IT infrastructures. Additionally, university's resources are occupied and teaching staff is needed to supervise the attending students.

In this paper, we present a concept for planning and deploying virtualized IT infrastructures (hosts and network) for higher education purposes and show an implementation including tool supported management of the virtual environment. The management platform facilitates the administration of virtual machines by students and thus frees the teaching staff from that duty. As a proof of concept, a number of different teaching environments used in a lab course on IT security have been virtualized. The course is intended for graduate students and poses high demands on the infrastructure, its availability and its performance, while security aspects have to be taken into account. Concluding the paper, experiences made during two years of productive use as well as updating the system to new releases of the virtualization software are pointed out.

1 Introduction

The Ludwig-Maximilians-Universität München and the Technische Universität München offer a practical course on IT security for graduate students. In this context, multiple workstations and servers are provided. Over time, defects of hardware components occur more often, which demand human interaction in order to ensure further operation of the lab. Moreover, the infrastructure is only accessible during certain days of the week and for a limited amount of time due to the institute's opening hours. As the course is attended by students of two different universities located at different places, the students' time of travel is considerably high as well. In order to improve the situation and save valuable time of the teaching staff, the virtualization of the whole lab course seems a suitable solution.

1.1 The Lab Course Use Case

The IT security lab course mainly deals with configuration aspects of network components and IT services. Security flaws are explained and the misuse of those illustrated in experimentals using sniffers, portscanners, several hacking tools and executing Denial of Service (DoS) attacks. Securing networks, their components and IT services are tasks students have to deal within the course. The course has a maximum capacity of 40 students working together in groups of two, each group having two computers at hand. During the course, several different network topologies are needed. Thus, a mechanism for simple and dynamic adaption of the infrastructure is necessary.

1.2 Requirements

In the context of the practical lab course, four major requirements have to be fulfilled while designing and implementing the lab course infrastructure:

1. Security.
   Due to the fact that the course deals with IT security, one important factor while designing the virtual lab is defined by IT security itself. Security aspects of the underlying host system are a primary issue in order to guarantee a highly available and secure course environment. As some critical experiments like DoS attacks and password cracking are carried out within the course, the protection of the outer world is an important fact as well, while access to the Internet is necessary to download software components.

2. Transparency.
   The virtualization must not be visible to the students. No student needs to have any access to or knowledge of the underlying physical hardware components. Students don't even have to know about the virtualization in order to work with the components provided.

3. Accessibility.
   Access to the machines should be possible from any workstation connected to the Internet, including both console access and the use of graphical user environments in a secure manner with adequate performance supporting small bandwidth Internet connections as for example ISDN or even analog dial-up connections. Besides, a large variety of operating systems used by the students has to be supported in order to connect to the lab. In particular, a minimum of Apple MAC OS, Microsoft Windows and Linux/UNIX on the client side should be usable, while the virtual machines themselves are based on Linux without exception.

4. Management.
   Management aspects have to be separated into two major dimensions:

   (a) Management of the virtual lab infrastructure.
   To ease the management of the virtual lab is a major requirement, meaning that it has to be comparatively easy to keep the lab up and running and to ensure a secure environment for the experiments. This discipline is left to the teaching staff and system administrators, as it only deals with the hosting system itself and not with the virtual workstations.

   (b) Management of the virtual machines.
   The management of the virtual workstations shall be left to the students, releasing the teaching staff and system administrators from that duty. It has to be possible for all the students participating in the course to manage their own virtual machines in a comfortable way. In particular, they have to be able to restart their machines if a problem occurs, create snapshots as backups or even reinstall a clean system image in case of a major misconfiguration. These operations should not be allowed to be executed on foreign virtual machines related to other students.

1.3 Contribution of this Paper

This paper describes how to migrate an existing lab infrastructure to a virtual lab infrastructure taking security, transparency, accessibility and management aspects into account. The implementation shown in section 4 contains more than 40 virtual machines (also referred to as *VM*s) including both workstations and servers, all of them having multiple network interface cards, more than 20 virtualized bridges, hubs or switches hosted by only one physical machine. Additionally, different network topologies are implemented, having the opportunity to switch between them dynamically using prebuilt scripts. The virtual machines are accessible using graphical desktop environments or secure shells 24 hours a day. The usage of virtual private network (VPN) technologies completes the implementation.

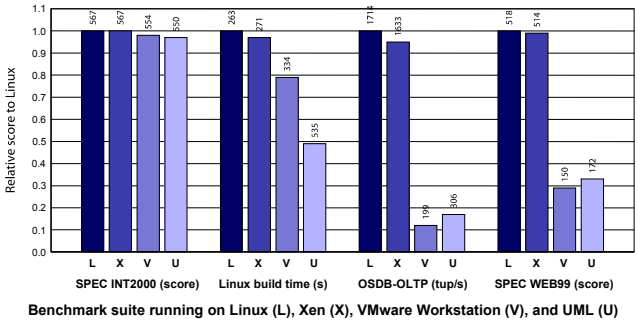

Figure 1. Virtualization benchmarks [7]

The remainder of the paper is structured as follows. First, Xen is introduced in section 2 as it will be the virtualization tool of choice for the implementation later on. Following, the concept, implementation and deployment of the virtual lab course is described and the fulfillment of the before mentioned requirements is shown. Concluding the paper, a short overview of the performance of the implementation experienced in real life usage is given in section 5 and some possible improvements and further work is pointed out in section 6.

2 State of the Art and Related Work

Beside virtual machines, network components like switches, hubs and firewalls as well as their connections have to be virtualized. These facts raise some additional requirements for the implementation of the virtual course infrastructure. Extensive tests [4] which virtualization technique is suitable for our usecase have been carried out and resulted in using Xen.

The next section introduces Xen as an example for host virtualization as it was the fastest platform (see figure 1) available when we started the project three years ago in 2005. Additionally, the network setups can be realized using the techniques and components provided by Xen, whereas User Mode Linux and VMware were too slow or not able to create virtual instances of our network setups due to the lack of several virtual components, in particular hubs. Details related to the implementation can be seen in section 4. Section 2.2 points out related work in the area of virtual lab courses and concludes this section.

2.1 Xen

Xen [12] is a hypervisor that uses the *para-virtualization* concept. Xen provides an interface which is very similar to

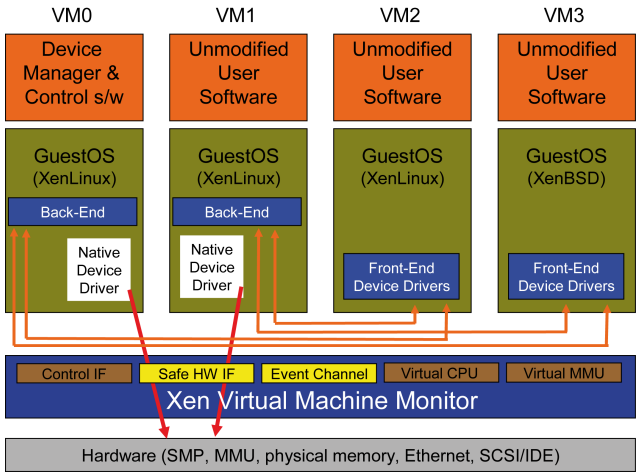

Figure 2. The Xen architecture [8]

the x86 architecture. In order to operate a guest system using Xen, some lines of code of the hosted operating system have to be adapted in order to run using the Xen interface instead of the underlying hardware (e.g. an x86 architecture). Therefore, Xen is only used in combination with open source operating systems as guests (in para-virtualization mode) or technologies like Intel VT-x and AMD-V formally known as Vanderpool and Pacifica.

Figure 2 depicts the layered model Xen implements. The *Xen Virtual Machine Monitor* (VMM, also called the *hypervisor*) [9] introduces an additional layer on top of the hardware of the host system. Via the *Safe Hardware Interface* access to the hardware is granted. The VMM is responsible for every component of the hosting system being accessed by just one system at a time. For example, in case of a CPU (multiprocessor systems are supported) every virtual machine is bound to a *Virtual CPU*. If the virtual machine becomes active that virtual processor is bound to a physical CPU core by the VMM and provides the compute power demanded by the VM.

On top of the VMM all the virtual machines – also called domains – are executed. All domains are treated equally, except the so-called *domain0*. This machine is privileged and its job is to control and manage all the others (the so-called *domU*s). Usually, but not necessarily, domain0 owns access to all physically available hardware components via the hypervisor. This is the reason why there are two different versions of kernels for Xen Linux: One kernel including drivers for the access to the physical hardware that is appointed in domain0 and another kernel without this functionality operated by the guest machines. Both versions can be configured and recompiled manually to add additional features. Frontend drivers for the access to virtual hardware served by the Xen backend system should be included in both versions.

To protect the system from illegal access, Xen makes use of the ring concept of the x86 architecture. Rings – there are four of them, but mostly only two of them are used – represent different access layers. Ring zero represents the kernel mode and ring three is known as the user mode. Xen modifies this mapping as follows: The hypervisor operates in ring zero and the operating system is shifted to ring one. Consequently, the operating systems can be controlled by the hypervisor. OS instances running in ring one are not allowed to execute any privileged instruction on the processor. This is why the operating system has been modified and runs some new functions called *hypercalls* instead of prohibited systemcalls. Trying to pass a systemcall anyhow results in an exception thrown by the processor and is handled by the hypervisor. This only holds true on 32 bit systems, 64 bit systems behave differently.

2.2 Related Projects

Research in the area of virtualizing IT environments used for educational purposes has already been carried out by other groups of researchers. Mostly, the work focuses on the simplification in creating lab infrastructures by booting a number of virtual machines and connecting them to special networks automatically. Usually, this is done according to configuration files built by administrators in advance. Examples include MLN (My Linux Network) [2], VNL (Virtual Networking Lab) [6] and VNUML (Virtual Network User Mode Linux) [11].

The tools developed in these projects ease the process of deploying virtual infrastructures. They also provide tool support for this task, but they are lacking a concept of how to transfer existing lab course infrastructures into virtual environments conveniently. Reconfiguration issues based on easy to use configuration files for whole network setups, as well as per user management interfaces for comfortable and secure remote access to the virtual machines are out of scope.

3 Basic Ideas & Concepts

The fulfillment of the requirements on the lab course infrastructure leads to some obvious attempts. This section presents some ideas on how to conform to these requirements. Afterwards, an implementation of the concept derived in this section is found in section 4.

1. Security.
   To protect the host from attacks originated in VMs, it is necessary to strictly separate the physical system from the VMs. Therefore, the only point of access to the virtual environment is delegated to a VM (the so-called

login server) directly bound to a physical network interface connected to the Internet. In our case, Xen offers a feature allowing the assignment of a physical NIC to a VM exclusively. This feature is granted by the hypervisor.

Using firewalls to prevent unauthorized access to the virtual networks or the management system is a further step towards securing the platform. As communication between the virtual machines via the pre-configured management network (see below) is unwanted, it is prevented by firewall rules. Firewalls implemented in the login server protect the login server from outer world attacks. Also, connections to the Internet can be filtered to prevent attacks from the lab harming foreign resources located outside the lab.

2. Transparency.
   Transparency is guaranteed by virtualizing every single component used for the course environment. Every workstation and every server (see figure 3) is virtualized. Thus, nobody has access to or knowledge of the underlying hardware which serves the infrastructure. This transparency adds additional security to the system. If one of the components is compromised successfully, only one virtual component could be intruded instead of the physical host. Besides, the host itself can be secured by several security means and be placed in a private network segment.

3. Accessibility.
   In order to access the virtual network, a dedicated virtual login server (see figure 3) is used. One of its virtual network interfaces is directly connected to the Internet, while another one connects to a management network. It is either possible to tunnel any traffic through the login server to the designated port on the target machine (e.g. port 22 for SSH) or to connect to the network using VPN technologies. In the latter case, the login server acts as the security gateway and the computer connecting to the VPN becomes part of the management network and thus can access any virtual machine.

4. Management.

   (a) Scripts for booting the scenarios.
   Scripts to start and stop virtualized scenarios are used. The scripts include virtual machine configurations, the creation of network resources e.g. hubs, switches and bridges and the correct wiring of the components. In our case, the creation of these scripts can be simplified using a feature provided by Xen: Parameters can be given and calculations can be performed in the configuration file, which enables the administrators to create virtual machines in a loop within a script. Individual configuration settings of the virtual machines can be calculated in the configuration file depending on the loop parameter.

   (b) Management platform for student use.
   A management platform is introduced in order to enable the participants of the course to manage their own virtual machines. The management includes rebooting, shutting down, backing up, recovering old snapshots and resetting a virtual machine to its initial state as a minimum subset of features. A management interface operated by the hosting system is mandatory for these tasks. A management proxy in the context of the login server grants remote access to the management interface. Making use of reliable authentication and authorization capabilities combined with encrypted data transfer ensures a secure operation of the management platform.

Figure 3 illustrates the basic ideas of the concept this work is based on. A main interest is to isolate the hosting system from the virtual infrastructure due to security aspects.

To ensure the accessibility of the virtual machines, a dedicated management network has to be set up, complementing the teaching network. This enables the users to connect to their virtual machines, regardless any misconfiguration of the interface cards connecting to the teaching network. In order to access the management network, the login server has to be used. A firewall running on this server allows remote access to the VMs, e.g. using SSH tunneling or VPN technologies. Remote logins on the gateway are not permitted for security reasons, of course. Additionally, outgoing traffic to the Internet can be masqueraded using NAT router capabilities, providing Internet access for the teaching network. Connections initiated by virtual machines to the Internet and communication among virtual machines using the management network is not desired and thus not permitted by restrictive firewall rule-sets.

Privileged access to the hosting system is needed in order to manage the virtual machines. For this reason, a management interface is introduced running on the host, which is able to control the VMs (e.g. (re-)booting, shutting down, backing up, recovering old snapshots, ...). To reduce management interactions performed by the teaching staff, a management proxy granting access to the management interface is deployed on the login server. This proxy passes connections originating from the Internet to the management tool transparently. This conserves transparency and enhances security aspects, as using a direct manage-

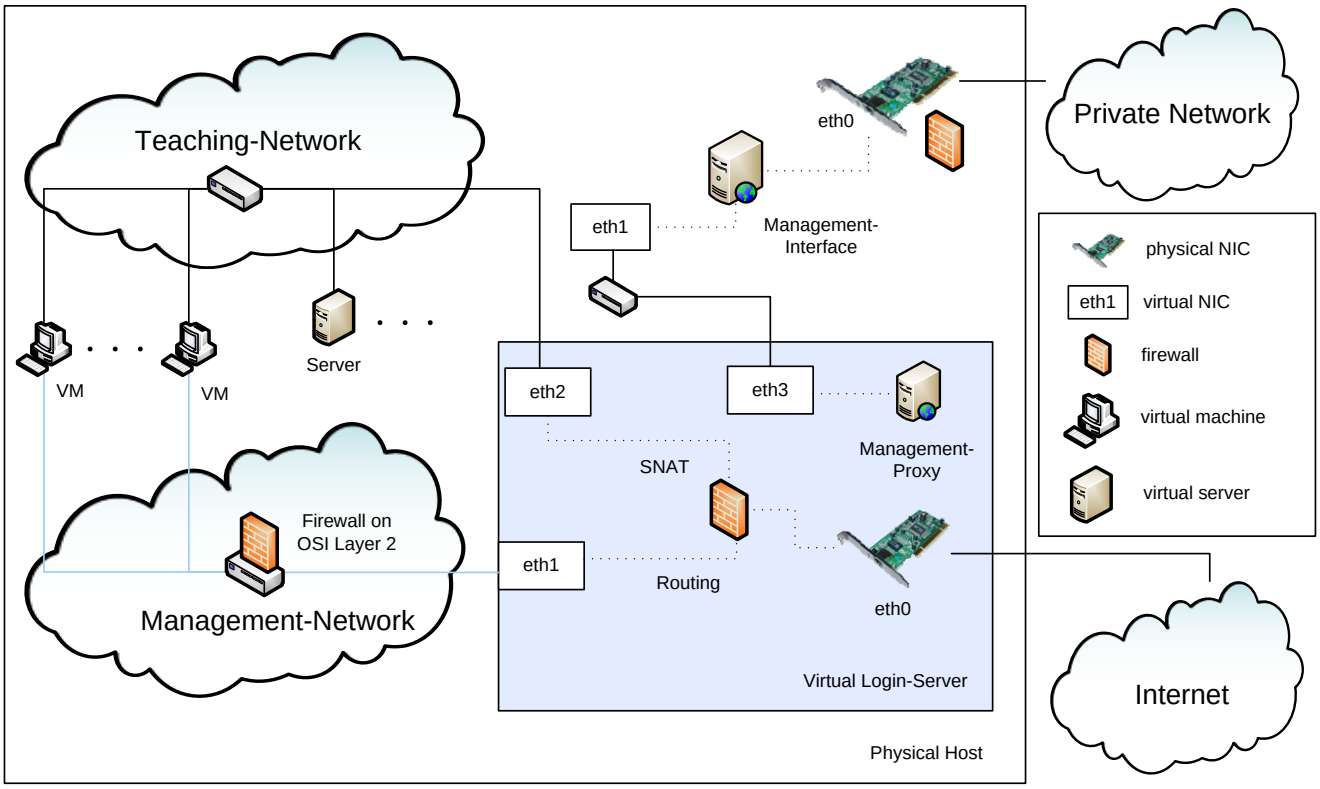

Figure 3. Conceptual view

ment connection to the host would result in granting students access to the host via HTTP and publishing the host's IP address. Of course, additional authentication and authorization processes have to be established. Therefore, a customized authentication handler on the web server that compares the password given to the root password located in the /etc/shadow file in a virtual machine is used. For this purpose the image of a VM is mounted read-only by the web server.

4 Deployment

Figure 4 illustrates the instantiation of the concept presented above for a lab course provided by the two universities. In this section, first the basis for the implementation is described briefly, including the hardware of the hosting server as well as the software chosen for the virtualization process. Section 4.3 gives a detailed overview of the implementation, before the upgrading process from Xen 2 to Xen 3 is described in section 4.4.

4.1 Hardware Basis

At the beginning of the project, various tests [4, 5] have been performed in order to figure out which kind of hardware is necessary to virtualize the lab course shown in figure 4. The results have proven that no CPU bound bottleneck is suspected, but RAM seems crucial as 40 machines for the student work and some additional servers should be operated on one single host.

SuSE Linux filesystem images created by the YaST installer including tools for development, the graphical desktop environment KDE and some free disk space for the students' work are about 3 GB of size. Those plus additional disk space for backups have to be hosted on the server. Therefore, a SATA RAID using RAID level 1 to ensure the integrity of data is used.

The productive server is a Fujitsu-Siemens server with two AMD Opteron processors (model number 246 at 2.0 GHz), 4 GB of RAM (as the initial setup is using Xen 2 and thus only supporting 32 bit environments) and about 400 GB of Soft-RAID storage (RAID level 1).

4.2 Software Basis

To implement the virtual lab, Xen was selected among other virtualization tools. Its performance surpasses all other tools that can be used to provide virtual machines and network components when we started implementing the project in 2005. Ian Pratt demonstrates the performance

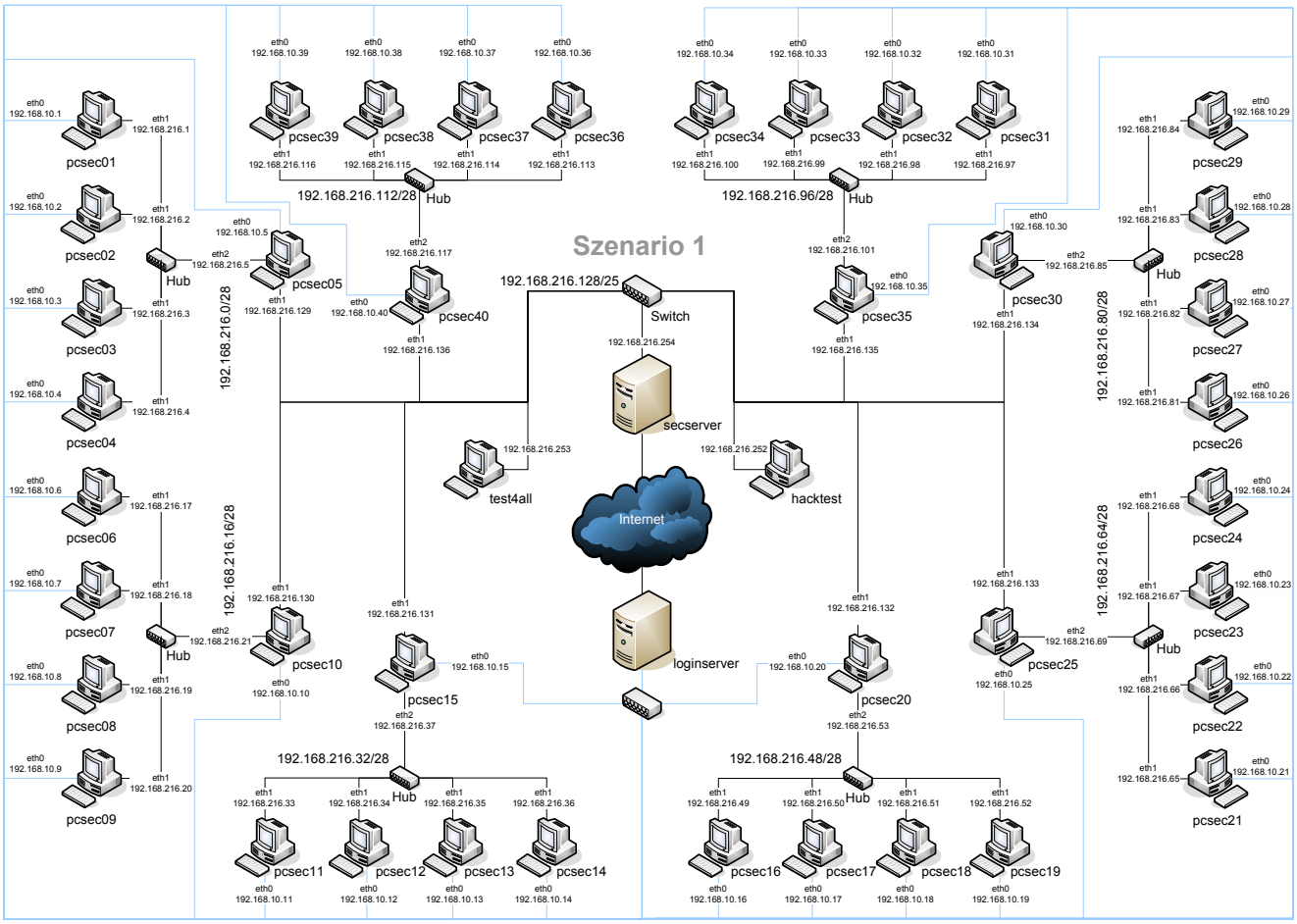

Figure 4. One of the virtual network environments

of Xen compared to VMware Workstation and User Mode Linux (UML) [7] in figure 1. The conclusion drawn in his work is that Xen performs best by far and in fact is close to a stand-alone Linux system.

Over and above this fact, Xen is very stable and implements a very powerful scheduling algorithm. As claimed by the Xen developers, it is possible to attack one virtual machine using DoS techniques, while other machines running on the same physical host are nearly not affected. As DoS attacks are executed during the course by students, this is an important fact which has been proven true.

4.3 Instantiation for the Lab Course

One of the scenarios used for the IT security lab course is shown in figure 4. The implementation of which is based on Xen version 2. This is due to the fact that Xen 3 did not perform well considering stability in our tests. It was still in beta stadium and thus the decision to deploy a version 2 system was made.

In this scenario, two switches, eight hubs, four servers, 40 student PCs and 94 network interface cards are needed. The darker marked interconnections between the servers and workstations depict the network topology used within the course (the "teaching network"), while the lighter connections represents the management network. The latter has to be deployed in order to grant access to the student machines as described in section 3.

To facilitate the use of graphical applications, X may be forwarded using SSH tunneling capabilities. This only proves suitable in case of the students working at machines with local area network connections to the system hosting the virtual lab. Besides, FreeNX [3], a remote desktop solution, is installed on every VM. FreeNX enables the export of the desktop environment in a very efficient manner. In our tests, even old-fashioned analog dial-up connections resulted in no overwhelming but acceptable performance. Both possibilities to use graphical interfaces can be used in combination with any of the two ways of connecting to the lab, either SSH tunneling or the usage of OpenVPN.

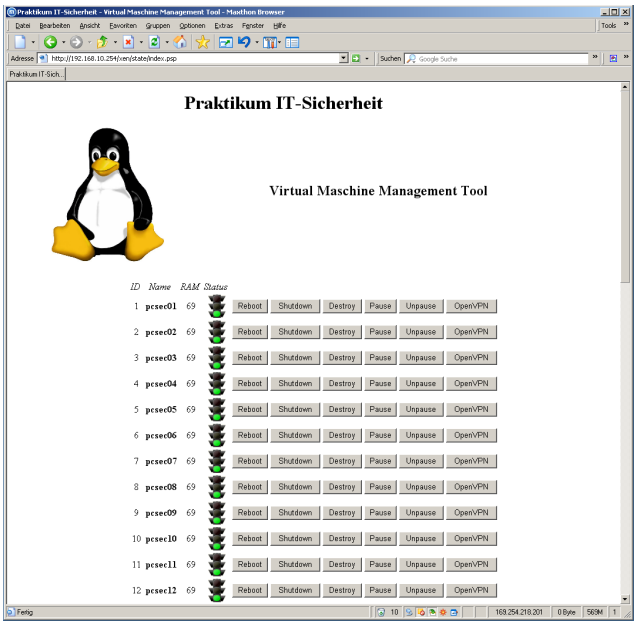

Figure 5. The management web interface

The possibility to access the interface of the Xen daemon *xend* using a web server via a python module enables the management of the virtual machines. Access control is realized using a custom-made authentication handler written in python. It compares any given password to the root password set in the virtual machine that shall be managed. Thus, access to both the virtual machine and the management interface is granted using just one single password. In this case, an apache web server in combination with the *auth* module is used and any traffic is encrypted using SSL. Figure 5 shows a screenshot of the home-grown simple management interface developed and used for the virtual lab course.

4.4 Upgrading from Xen 2 to Xen 3

New versions of SuSE Linux Enterprise Server do not support Xen 2 any longer and the demand to keep the software up to date is hard to satisfy. After operating the lab based on Xen 2 for two terms, a migration to Xen 3 was desired.

Some minor adoptions have to be made to the network configuration files and the configuration files used to create virtual machines. Both issues are more or less based on syntactical changes in Xen 3, resulting in minor problems.

In contrast, the port of the custom-made management tool to Xen 3 demands greater efforts because some interfaces of xend have changed. As a result, a part of the management tool has to be reimplemented using the new interface. Afterwards, the virtual infrastructure is working properly again.

Regarding the stability, both versions of Xen do not differ. Anyway, differences regarding the performance are obvious. While Xen 2 is a bit faster in general, Xen 3 is more powerful in accessing virtual disks using the new *xvd* (Xen virtual block device) driver. Both symptoms can be easily observed, e.g. by installing or booting new virtual machines.

One additional feature of Xen 3 is the possibility to virtualize Windows Workstation if suitable processors with the Intel-VT or AMD-V command sets are used. Our security course could thus be extended to Windows security issues as well. At the moment no suitable server hardware is available so that the course's focus lies on Linux. Anyway, the concept shown above still holds for Windows or mixed scenarios. Only some implementation issues would have to be adjusted as for example the current management platform just supports authentication and authorization methods for Linux VMs.

5 Experiences

Operating the virtual lab for four terms productively, no major problems occurred up to now. No problems related to stability are experienced, even critical actions like DoS attacks and malformed network packets sent during the lab course do not harm the infrastructure. Performance related problems are not noticeable, although 44 virtual machines are executed on one single physical host. Usually, load is distributed evenly over the week. Just in case of special events like tests or demonstrations that have to be passed as a milestone during the course, load is high, but the performance experienced by the end-user is still acceptable.

Compared to a native Linux machine a virtual Linux machine operated by Xen is insignificantly slower. Running more than one virtual machine at the same time is even more efficient. Due to intelligent scheduling algorithms, booting the virtual lab with all the virtual machines and services takes about 12 minutes. Hence, one single virtual machine needs about 16 seconds in average to start up into runlevel 5.

Network performance does not pose problems as well, even though many students are using graphical desktop sessions and all the traffic to the Internet is handled by only one physical network interface card. The network itself does not provide a bottleneck in the virtual lab. This also holds true for the virtualized network components interconnecting the machines among each other and providing the management network. Actually, the virtual network components in some cases show better performance than physical ones as they are simulated by kernel operations of the underlying host system.

The most important gain of the virtualization process is the reduction of administration costs to about a sixth part

compared to the initial course setup. Up to six advisors have been employed to manage and supervise the course and its attendees. Now, this task is accomplished by just one student advising the participants of the course regarding the content. As the hosting system now is a reliable server system, no disruptive incidents related to defects of hardware occurred yet. This was a frequent case before the virtualization of the lab and demanded a lot of in-time administrative work. Virtualizing the lab course, the hardware issues have been exchanged with the problem of managing VMs. Providing a management interface to students enabling reboots of hanging machines, etc. releases the teaching staff and shifts the efforts to the students while maintaining control. Additionally, access to the lab is possible from every computer connected to the Internet 24 hours a day. This leads to a maximum of flexibility in time and place for the students, especially as our lab course is offered at different universities.

6 Conclusion & Future Work

In this paper, a concept for a virtual IT infrastructure for higher education teaching is introduced. Security aspects, transparent usage of and convenient access to the infrastructure, as well as the comfortable management of the lab course are main requirements while designing the concept. The deployment as a proof of concept using Xen is a practical lab course dealing with IT security offered at the two Munich universities.

In everyday use, the experiences are predominantly good. The university's premises for the lab course could be released and valuable time of the teaching staff and administrators could be saved. This is mainly due to the fact that instead of managing several student PCs the management of one much more reliable server system has to be accomplished. The management of the virtual student PCs is performed by the students themselves, providing them with a web based management platform. In sum, this saved about two thirds of the costs to run the course.

In future work, the virtualization of other practical lab courses dealing with more technical content is considered. In this context, the question to which technical detail virtualization approaches seem applicable has to be answered. However, the concept presented in this paper has established a template for virtualizing other teaching environments.

Acknowledgment

The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful discussions and valuable comments on previous versions of this paper. The MNM Team founded by Prof. Dr. Heinz-Gerd Hegering is a group of researchers of the University of Munich, the Munich University of Technology, the University of Federal Armed Forces Munich and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences. Its web server is located at `http://www.mnm-team.org`.

This paper was supported in part by the EC IST-EMANICS Network of Excellence (#26854).

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM Press.

[2] K. Begnum, K. Koymans, A. Lrap, and J. Sechrest. Using Virtual Machines in System Administration Education. In *Proceedings of 4th International System Administration and Network Engineering Conference*. System and Network Engineering, 2004.

[3] FreeNX Project. FreeNX. `http://freenx.berlios.de/`.

[4] T. Lindinger. Machbarkeitsanalyse zur Virtualisierung des IT–Sicherheit Praktikums. Technical report, Ludwig-Maximilians-University of Munich, Oct. 2005.

[5] T. Lindinger. Virtualisierung einer Praktikumsinfrastruktur zur Ausbildung im Bereich Sicherheit vernetzter Systeme. Master's thesis, Ludwig-Maximilians-University of Munich, May 2006.

[6] S. Liu, W. Marti, and W. Zhao. Virtual Networking Lab (VNL): its concepts and implementation. In *Proceedings of the 2001 American Society for Engineering Education Annual Connference and Exposition*, Texas, USA, 2001. American Society for Engineering Education.

[7] I. Pratt. Performance of xen compared to native linux, vmware and user mode linux. `http://www.cl.cam.ac.uk/Research/SRG/netos/xen/performance.html`, Dec. 2004.

[8] I. Pratt. *Xen Status Report*. University of Cambridge, Dec. 2005.

[9] University of Cambridge. Computer Laboratory - Xen virtual machine monitor. `http://www.cl.cam.ac.uk/Research/SRG/netos/xen/`.

[10] University of Cambridge. XenoServers. `http://www.xenoservers.net/`.

[11] Universität Koblenz. Virtual Network User Mode Linux. `http://www.uni-koblenz.de/~vnuml`.

[12] XENSource. XenSource: Delivering the Power of Xen. `http://www.xensource.com/`.