A SWOT Analysis of Virtual Laboratories for Security Education

Alan Davidson, Javier de La Puente Martinez, Markus Huber

A SWOT Analysis of Virtual Laboratories for Security Education

Alan Davidson, Javier de La Puente Martinez, and Markus Hüber

Department of Computer and Systems Sciences,
Stockholm University/Royal Institute of Technology, Sweden
{alan, jdlpm, mhuber}@dsv.su.se

Abstract. Work is active in many institutes of higher education on utilising virtual computer environments for creating laboratories for practical course-work. Computer Security education is one area where virtual environments are proving to be useful, and where several schools have reported their own schemes for implementing environments for practical exercises. In this study we attempt to take a somewhat broader look at what the use of virtualisation technology can imply terms of a number of factors, i.e. the pedagogy, security, licensing, administration and cost. A simple analysis of the general strengths, weaknesses, opportunities and threats of virtual security laboratories allows us to motivate design choices when implementing yet another of these experimental environments.

1 Introduction

Latest developments in virtual computer environments are encouraging a spate of experiments with building virtual laboratories for use in education. Many of the advantages are clear and oft cited, such as how virtual machines allow for efficient use of hardware, and how they can ease system administration.

Courses in ITC security often have special requirements when it comes to their practical laboratory environments. The tools and environments used many times require a student to have administrator privileges. This can create problems if students, either maliciously or by accident, can cause damage to their own environment or others'. Some educational tools, such as virulent malware, or techniques, such as sniffing and password cracking, can be dangerous things to have in a normal working environment. The kind of virtualisation that has been termed *platform virtualisation* [1] therefore seems especially applicable in solving several problematic issues that are typical for ICT security courses.

An existing sandbox laboratory run by the SecLab group at the Department of Computer and Systems Sciences at Stockholm University has previously been used to allow students a practical test environment. Computers within the sandbox were are fitted with removable hard drive cassettes for the computers' operating systems which allowed students to have administrator privileges for their system without affecting the computers. They simply kept the removable cassettes for the duration of their experiments. The sandbox is segregated from

the rest of the department's computer networks with a firewall that allows only outgoing HTTP requests. The firewall machine also implements an operative image server which allows students to recreate their environments from scratch in case they should suffer irreversible problems. Various physical devices such as extra network interfaces, cables and hubs are kept in the laboratory allowing students to build sub-nets and connections as any exercise might require.

A number of standard practical exercises for differing operating systems have been devised for execution in the sandbox laboratory. Apart from these we have had an initiative to allow students themselves to create their own experiments and pass them on to others [2]. Though this laboratory has been successful the possible advantages of running an equivalent virtual laboratory have prompted the authors to investigate further.

Several similar projects to utilise virtualisation for diverse ICT security learning environments have been documented [3–10]. Such experiments are encouraging, yet when it comes to implementing our own virtual version of a laboratory we find that there are several general questions that we can identify and that we hope to address:

- Pedagogy. What kind of pedagogical issues are there when moving from a physical environment to a virtual one? Can students learn the same things in either kind of environment?
- Security. Since we are working with potentially dangerous tools and methods we need to be sure that they are efficiently contained. What additional security advantages or problems can one expect with virtual environments?
- Cost. Virtual environments are often touted as being particularly economical. They allow for maximum use of hardware capacity. But moving to virtualisation will have its own costs, as will running the virtual environment. How can these costs be kept to a minimum?
- Administration. As with cost, ease of administration is often cited as one of the major advantages of virtual environments. How true is this in teaching environments?
- Licensing. Software licenses for physical machines are relatively easy to handle since the number of licenses relates to the number of machines. How does this situation relate to the licensing of innumerable virtual machines? Are the opportunities for pirate copying of university licensed software any different when using virtual machines.

These were the main questions that we bore in mind when studying previous virtualisation projects and when designing our own.

2 A SWOT Analysis

In order to gain some perspective over the issues we attempt to structure them according to a simple SWOT analysis, i.e., the strengths, weaknesses, opportunities and threats of virtual environments. We include in the concept of virtual environments both platform virtualisation and virtual networking. We avoid

general virtual machine issues in order to concentrate on the implications for a security laboratory.

2.1 Strengths

Virtual environments allow us the possibility to give students administrator privileges. This is also possible with the physical laboratory described above. With hard drive cassettes however, one does have to be physically present in the lab to run an individual student machine. One could imagine a system of allowing students to connect to physical machines through VPN connections, thereby freeing them from working in the physical environment. However, the combination of allowing students to control their own image of a system and to be able to work from a distance is only achievable through virtual machines.

Virtual machine implementations invariably have simple means to save the state of the system, and allow roll-back to such states. When students work with administrator privileges and with security tools that manipulate complex systems the opportunities for misconfiguration and accidental damage are many. The safety and ease of administration that virtual machines offer is one of their primary strengths. [8]

It is generally true that virtual machines are scalable for the task at hand. In a security laboratory there can be several situations where this is particularly relevant. Take for example a firewalled network experiment with three machines, a machine representing a network to protect, an attacker machine, and a dedicated firewall machine between the two. Though the protected machine may have a fully fledged operating system, the attacker would only need a minimal system to run attack tools. The firewall would be a run on an absolute minimal operating system. Running such an environment on physical machines would be less than effective use of hardware.

Versatility in networking scenarios and services is a central requirement in networking security education, and can be supported by virtual environments [11]. In a physical environment students can reconfigure networks by moving cables, switches, hubs etc. But this can also be achieved in virtual networks, and what is more, system images with varieties of services can be kept on cheap secondary memory ready to be placed in a network at a moment's notice.

Cost savings are possibly the most commonly cited reason for employing virtual architectures in general. This is a noticeable issue with our physical laboratory where security exercises are not run all year round, and because of sensitive nature of much of the equipment and software that is used in this laboratory, it is unsuitable to run other courses in parts of this laboratory in the interim. Machines lie dormant, whereas if the environments were virtual they could be more easily swapped out to accommodate other laboratory situations. Furthermore, it has been proposed that server based virtual machines can also give a new lease of life to an old and tired machine park as it can effectively be utilised as smart terminals to interface with the server [12].

Cost savings with virtual environments are most often associated with efficient use of hardware. There are however issues of software costs to consider as

well. Cost could well be counted as a weakness in that commercial virtualisation solutions come with a commercial price tag. Given that there are a number of fully workable open source and freeware solutions available such as VirtualBox, KVM, Qemu, etc. the cost of software is well within the security laboratory's typically shoestring budget. What the true cost of free software is may be the subject of constant debate, but in practise we have found that these "free" solutions are practically feasible without prohibitive administrative overheads.

2.2 Weaknesses

Virtual machines are good versatile tools in as much as they implement a level of abstraction above specific hardware. This level of abstraction in turn comes with the cost of loss of computing power. A number of typical security experiments such as encryption, key and password cracking, vulnerability scanning, etc. are demanding on computational resources. We could expect such heavy operations to be slower. The problem is mitigated by continuing developments in virtualisation technology, and modern processors may be expected to have hardware acceleration for virtual machines.

Computing environments are general, integrated tools where all manner of uses are concentrated into a single versatile machine. With virtual machines we are creating segregation within that environment. Where communication between tasks is simple on an integrated environment, it can be painful across segregated virtual environments. For the security laboratory this problem is noticeable in the way that results from an exercise should be included in the students hand-in documentation. If the documentation is done within the secure environment then there may be extra effort involved in extracting that documentation to a normal study environment where hand-ins can be accepted. If the documentation is done in a study environment then there will be the problem of extracting result from the secure environment. One might also expect computer screens to become easily cluttered with windows to virtual machines overlapping with text editors, email agents, and other such tools necessary for normal student activities.

Perhaps the trickiest issue is how we with virtual environments are attempting to imitate real environments, and we are attempting to teach lessons about real environments. In some cases the difference will no doubt be negligible, such as if we are only experimenting with the use of a software tool within a single machine environment. But if we replace building networks with cables, switches and hubs with building virtual networks through a software interface, how can we be sure that we are not missing out on vital parts of the learning experience? Furthermore, there will be an overhead involved in teaching the student how to cope with virtual environments that may not be as intuitive as interacting with a physical world.

2.3 Opportunities

Among the strengths that we noted above is the idea that one can both allow students to control their own environments, yet not restrict them to a physical locality. By connecting to a virtual machine server it can be possible to run exercises within a secure environment from a distance. It may also be possible to supply students with a secure environment that could be run virtually completely within their own computers. This would not only free up some of the universities own computers, but also potentially provide a means to hold distance education with practical exercises that would previously have been implausible without requiring the students to occasionally come to the physically secure laboratory.

Some security experiments have been difficult to implement in our physical laboratory. A case in point is a firewall exercise that was developed for a group of some 16 students simultaneously using eight computers. Two more computers in the laboratory were used by staff to act as on the one had the machine under attack and on the other hand the attacker. The students were in turn required to configure each of their machines to act as firewalls between the staff machines. Each attack required constant re-routing of network traffic through each of the firewalls. With virtual architectures, each students machine can be given their own internal network of three machines or more, completely and independently set up to run attacks from one machine to the others, while the student is required to properly configure the virtual firewall. Virtual environments can therefore be used to implement experiments that are difficult, if not unfeasible, in a physical network.

2.4 Threats

The above discussion on the weakness of virtual environments in terms of how they segregate working environments may have greater implications than just the practical difficulties. Some kind of connectivity will always be a practical advantage. In our own physical laboratory we choose to allow limited access to the outside world by opening port 80 for outgoing HTTP requests. This allows the student the ready source of information that they have come to expect in all walks of life, as well as access to security tools, malware and exploits. We find that this amount of access encourages students' individual initiative during exercises. Furthermore we allow students to use removable media to transfer materials to and from the secure environment, in particular to aid in their documentation of their work. There are methods to strongly segregate virtual networks [10] while virtual machines implemented with jails are useful when high level security is preferred, but it will of course be at the cost of connectivity. The ever present desire for connectivity is a potential security hazard as the tools and methods used within the secure environment may be difficult to contain.

We might have included the security of virtual environments as one of the strengths. It is true that in the laboratory environment the student will presumably be running the virtual machine within an unprivileged account, so even if

the virtual machine itself is running with administrator privileges, it can presumably do little damage to the host environment, beyond the possible containment problems discussed above. Nevertheless, the authors believe that the security of virtual machines is today overstated. It is not uncommon to equate the segregation of virtual environments with strong security but we have found no clear evidence that security is a goal of modern virtual machines. We must surely assume that the software that implements virtual machines is not intrinsically more secure than other general purpose software, which indicates that we should not assume that containment can be trusted. On the contrary we do see results that indicate that it is possible to programmatically verify that a process is being executed within a virtual machine [13]. This is surely a precursor to malware types that will be able to detect and to break their way out of virtual environments. We therefore prefer to take a conservative point of view on the security of virtual machines, and suggest that until the opposite can be shown, security laboratories run in virtual environments should be regarded with care and scepticism.

Problems with licensing might also be a reason to hold back on the implementation of virtual laboratories. The very portability of virtual machines means that it is simple to move a virtual image from one host to another. Insofar as licensing agreements that a school enters into require due diligence from the licensee to avoid spreading copies of the licensed software, this may be very difficult to uphold. A virtual image that contains licensed software can easily be copied and used on any number of other hosts. Perhaps the only means of limiting such use is though the enforcement of local security policies.

Another licensing problem is due to the way software companies specify the number of licenses. It is common to specify the number of machines that the software is to be installed upon. With easily copyable virtual environments it becomes far more difficult to calculate and control the number of machines that the software is installed upon, and it is a problem that current licenses specify limits on such numbers [4]. It would surely make better sense to have licenses that stipulate the number of copies that may be used concurrently [12].

3 Yet Another Virtual Security Laboratory

When implementing our own virtual version of our successful physical laboratory the above discussions have steered us to a number of design choices.

For reasons of security, administration and pedagogy, we have steered clear of solutions that involve installing virtual machines on the students own computers. It would be possible to implement administratively simple solutions such as providing so called live CDs i.e. bootable systems contained on a CD that could avoid starting the users own system and instead start a virtual environment. However, without considerable work on validating the security of such environments we regard the possibility of accidents that could violate the security of the students local environment as too high.

Our preferred solution is to have a central server for virtual machines. The virtual network is interfaced with that of our existing security laboratory, allowing the two environments to mix for seamless experimentation. A separate network interface allows the host server itself is to be accessed via department general purpose network, i.e., students easily access the server from the Internet, but the virtual machines that they start are automatically linked to the security laboratory network. Simple shell scripts allow students to copy, start and connect to their virtual machines via VPNs. There are two alternative ways to access the virtual machines: X-forwarding and VNC, both of them encapsulated in a SSH tunnel. If the user accesses the server from the university's facilities, X-forwarding is the preferred method. On the other hand, if the student accesses the server from a remote location where limited network capacity makes the X-forwarding impractical, VNC has proven to be a workable alternative. The server we are using is an IBM System x3450 running the x86-64 version of Debian GNU/Linux OS.

To avoid licensing difficulties as well as to keep down costs we have chosen virtual environment software that is open source, i.e. KVM (Kernel-based Virtual Machine) and Virtual Quare's VDE (Virtual Distributed Ethernet). KVM provides a core infrastructure for native virtualization and a modified version of QEMU is used to run the virtual machines themselves. VDE is compatible with QEMU and supports the creation of a virtual distributed Ethernet based on virtual switches and virtual crossed cables. The virtual machines themselves are to the larger part based on and configured to use open source software. The notable exception here is the use of Microsoft Windows XP, where the relatively liberal Microsoft educational licensing policy allows us to provide students with an environment that they generally have better experience in.

Our environment is still under development, and not yet been subject to full class deployment. Our ambitions for the near future are to implement simple web based control interfaces similar to those described in [14] to replace the control scripts that we have today. We are also developing a java based interface to the VDE virtual network software that will hopefully help to bridge the pedagogical gap between handling physical and virtual networks.

We have argued that costs can be kept to a minimum with the use of open source software for administering virtual environments, yet the work involved in designing and implementing this environment has been considerable. In our case we were provided a grant from the Royal Institute of Technology in Stockholm that was equivalent to some two person-months of paid time, as a project to further develop our teaching environment. We suspect that without such an injection of funds it would be very difficult to transition easily to such a solution.

References

1. Ramanathan, R., Bruening, F.: Virtualization: Bringing Flexibility and New Capabilities to Computing Platforms. Technical report, Intel Corporation (June 2004)

2. Davidson, A., Näckros, K.: Practical assignments in IT security for contemporary higher education. In Futcher, L., Dodge, R., eds.: World Conference on Information Security Education. Volume 237 of IFIP., Springer (2007) 25–32
3. Bullers Jr, W., Burd, S., Seazzu, A.: Virtual machines-an idea whose time has returned: application to network, security, and database courses. In: Proceedings of the 37th SIGCSE technical symposium on Computer science education, ACM New York, NY, USA (2006) 102–106
4. Hay, B., Dodge, R., Nance, K.L.: Using virtualization to create and deploy computer security lab exercises. In Jajodia, S., Samarati, P., Cimato, S., eds.: SEC. Volume 278 of IFIP., Springer (2008) 621–635
5. Hu, J., Cordel, D., Meinel, C.: A virtual laboratory for IT security education. In Feltz, F., Oberweis, A., Otjacques, B., eds.: EMISA. Volume 56 of LNI., GI (2004) 60–71
6. Hu, J., Meinel, C.: Tele-Lab IT security: A means to build security laboratories on the web. In: AINA (2), IEEE Computer Society (2004) 285–288
7. Kato, K.: Modeling and Virtualization for Secure Computing Environments (Invited Talk). Lecture Notes in Computer Science **4846** (2007) 196
8. Keller, J., Naues, R.: A collaborative virtual computer security lab. In: e-Science, IEEE Computer Society (2006) 126
9. Kuczborski, W.: A computer network laboratory based on the concept of virtual machines. In: Proc. 6th Baltic Region Seminar on Engng. Educ. 145–148
10. Sun, W., Katta, V., Krishna, K., Sekar, R.: V-netlab: An approach for realizing logically isolated networks for security experiments. In Benzel, T., ed.: CSET, USENIX Association (2008)
11. Alexander, M., Lee, J.A.: A scalable xen and web-based networking course delivery platform. In Parker, J., ed.: Proceedings of the Second IASTED International Conference on Education and Technology, Calgary, Alberta, Canada (July 2006) 131–134
12. Nastu, J.: Software virtualization: Virtual desktops offer ed-tech revolution. eSchool News (May 2008) 21–27
13. Raffetseder, T., Krgel, C., Kirda, E.: Detecting system emulators. In Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R., eds.: ISC. Volume 4779 of Lecture Notes in Computer Science., Springer (2007) 1–18
14. Kneale, B., De Horta, A., Box, I.: Velnet: virtual environment for learning networking. In: Proceedings of the sixth conference on Australasian computing education-Volume 30, Australian Computer Society, Inc. Darlinghurst, Australia (2004) 161–168