Creating and Using a Computer Networking and Systems Administration Laboratory Built Under Relaxed Financial Constraints

Michael P. Conlon
michael.conlon@sru.edu

Paul Mullins
paul.mullins@sru.edu

Slippery Rock University of Pennsylvania
Slippery Rock, PA USA

Abstract

The Computer Science Department at Slippery Rock University created a laboratory for its Computer Networks and System Administration and Security courses under relaxed financial constraints. This paper describes the department's experience designing and using this laboratory, including lessons learned and descriptions of some student projects performed in the lab.

Keywords: computer laboratory, hardware, laboratory, network, security, systems administration

1. INTRODUCTION

Designing a laboratory for teaching about networking and security is difficult. Many papers have been written about experiences designing and using computer network labs, often in cases of significant or even severe constraints on the cost of the space and equipment for the lab:

Riabov (2006) describes the use of a network simulator called *Virtual Opnet* for a small college without a networking lab. Crowley (2006) describes a network project performed with "live" CD's and open-source tools, obviating the need for installed software in a dedicated lab. Kretzer and Frank (2005) and Yuan and Zhong (2008) suggest using labs of surplused computers and open-source software for security lab activities. Other authors suggest a laboratory using equipment emulation, which they state will save more than ninety percent of the lab equipment cost (Li, Pickard, Li, Mohammed, Yang, and Augustus, 2008). Krap (2004) suggests the use of *User Mode Linux*, which enables the creation of multiple Linux instances on virtual machines on a Linux host. He points out that networks of virtual machines can be readily re-configured, but that many virtualization technologies suffer from poor performance.

It may be instructive to compare these experiences with that of a lab created without such constraints, and using real hardware. This paper addresses one computer science department's experience with a laboratory for computer networking and system administration developed at a time when there were few significant financial constraints in designing and equipping the lab.

In 2002, Slippery Rock University began the process of designing a new science and technology building. At that time, the Computer Science Department's network and security lab consisted of a collection of surplused computers,

in an inappropriate facility with very poor reliability. Because of this problem, and the generally poor facilities of the department at that time, we embarked on a serious effort to specify appropriate computer laboratories for future needs. We were not given significant budget constraints, so we asked for all the facilities that we believed we could reasonably use, anticipating that some of our requests would subsequently be denied. Surprisingly, virtually all of the department's requests for the new building were granted. The most elaborate of the requested facilities was the Network and Systems Administration Laboratory.

2. NETWORK LAB REQUIREMENTS

The lab was designed for use by Computer Science, Information Systems, and Information Technology majors in the *Computer Networks* course and by students in the *Systems Administration and Security* course, who are almost exclusively Information Technology majors. While the lab facility requirements of the two courses are not identical, the requirements had significant overlap and no conflict, especially since these two courses run in different semesters. Students in both courses need experience setting up servers, workstations, routers, switches, firewalls, etc. Computer Science majors in the Networks course need experience writing network software. Both courses, but especially the SysAdmin and Security course, should provide their students with experience using cracking and intrusion-detection software.

It was decided that the optimal lab for department needs should serve thirty students. Each student should have a desktop workstation and exclusive administrative access to several servers. Each student workstation should have access to its associated servers at a hardware level, so that servers and workstations can be administered before, during, and after operating system installation.

As emphasized by Kretzer and Frank (2005), the lab network must be disconnected from the campus network so that lab experiments will not interfere with ordinary campus communication. Other researchers agree: Hill et al. warn that a non-isolated security lab could be used by external crackers to attack systems in the campus network (Hill, Carver, Humphries, and Pooch, 2001). Bullers et al. report a campus-wide *Code Red* worm infestation originating in a lab that was not isolated (Bullers, Burd, and Seazzu, 2006).

Since it will be necessary at times to obtain software from the Internet, a firewalled-connection to the department's Unix server would allow a two-step software download: first to the Unix server, then to the student's workstation or server. The firewall would prevent student access to the rest of the campus network from the lab, since lab operations might interfere with ordinary campus network operation.

The lab was to have a flexible network configuration in order to make it easy to modify the topology of the lab network. The purpose of this was to allow students to partition the network into subnets to permit installation of routers and firewalls, and to permit students to study the kinds of problems that arise on LAN's of multiple subnets.

Refining the Lab's Requirements

As is often the case, some of the specifications turned out to be impractical. For example, it was thought that “blades” would make the best servers for the lab, because of their compact size and popularity in commercial server farms.

Blades consist of a set of book-sized computers installed in a case approximately 6 rack-units (“6U”) high, with about 12 servers per case. However, all of the blades in the case share a single optical drive. This meant that when one student was using the optical drive, such as for operating system installation, other students needing the optical drive for their blades would be forced to wait, perhaps for many hours. 1U servers, by contrast, are complete PC's in a one-rack-unit-high package, including optical drive, and USB, video, and network ports, so there are no delays from hardware sharing.

With 1U servers, thirty servers with network switches and switches for keyboard, video, and mouse, fit in a single rack. The entire lab could be outfitted with three such racks. Had blades been appropriate for a student network lab, two sparsely-filled racks would have sufficed.

The university's networking staff ruled that Internet access through the Unix server was unsafe for the campus network. This led to the use of “sneakernet”: students would download software in a nearby lab and carry it into the Network/Admin Lab on flash drive. This was unsatisfactory, but we had to live with it for some time.

Illustration 1: View of the Network/Systems Administration Laboratory. Two of the three racks of servers can be seen in the background.

Eventually, during a campus-wide network upgrade, it became possible to route all outgoing packets from the lab directly to the university's Internet gateway. This worked out better than the original proposal, allowing software downloads in a one-step process, yet protecting the campus network from lab activity. It would be better still if faculty could turn this connection on and off, preventing Internet access except when needed for a particular lab project.

It was thought that the desktop computers could be configured with several operating systems in a multi-boot configuration, enabling additional courses to make use of the lab. The idea was that the first disk partition would contain a copy of Windows to be used for courses such as *Productivity Software*, so students could learn computer procedures that are disallowed in other labs, such as installation of software. However, since SysAdmin and Networking students need to install operating systems on the remaining partitions, the original partition turned out to be too vulnerable to be relied upon.

The lab has occasionally been used for activities other than its main networking-and-sysadmin purpose. In particular, operating systems students have used the lab for kernel-level programming projects, but this has depended on the enrollment in Networks or Sysadmin being small enough to leave several workstations and associated servers unassigned. It may yet be possible to use the desktops for additional purposes, with the rise of "live" Linux distributions. This is the only lab where live distributions would be usable, since computers in other labs will not boot from removable media. Because of the cost of maintaining the lab, efforts are continuing to find ways to increase lab utilization without compromising its original purpose.

3. THE ACTUAL LAB

The Computer Network and Administration Laboratory at Slippery Rock University contains thirty desktop computers and ninety servers. The lab is networked with gigabit Ethernet, and connected to the rest of the Internet through a firewall that prevents computers in the lab from communicating with other computers on campus. Each functional unit (composed of keyboard, monitor, and mouse) is connected to a local/remote KVM (keyboard-video-mouse) switch, enabling a student to switch between the local desktop computer and a remote switch, which, in turn, lets the user select which of three servers to address (Illustration 2).

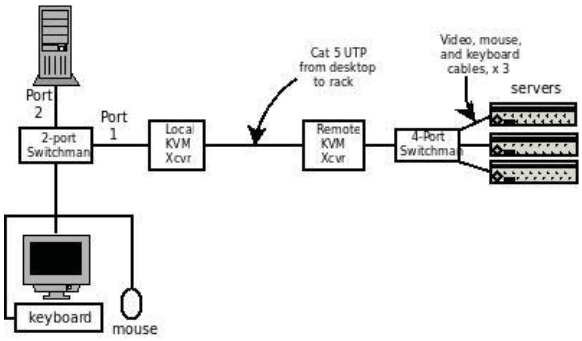

Illustration 2: KVM Switch Wiring. The local switch selects the local workstation (port 2) or a KVM transceiver (Port 1). Special key-strokes enable the user to select any one of three servers attached to the remote 4 port KVM.

There is a considerable collection of cables running between the racks and the workstations, or part-way along that path, with the potential for a considerable tangle of cabling. Rather than just tie them together with nylon tie-wraps, which would make re-cabling difficult, a cable trough was installed to hold the cables (Illustration 3). The trough runs along the back of each row of desks. Because the troughs are made of heavy-gauge steel wiring with large gaps between the wire, cables can enter or exit the trough at almost any point, in any direction.

At the end of each row, network cables follow a riser to the ceiling, where they enter a horizontal ladder-like device (Illustration 4). This ladder runs perpendicular to the rows of desks, carrying network cables from row to row. Such cable-carrying ladders are common in newer

construction; here, the difference is that the ladder is mounted below the ceiling so that cabling can be accessed without disturbing the ceiling tiles. The arrangement of troughs and the ladder permits re-routing and addition of network cables quite easily.

Illustration 3: View of a row of workstations, showing the wire trough for laying network cabling.

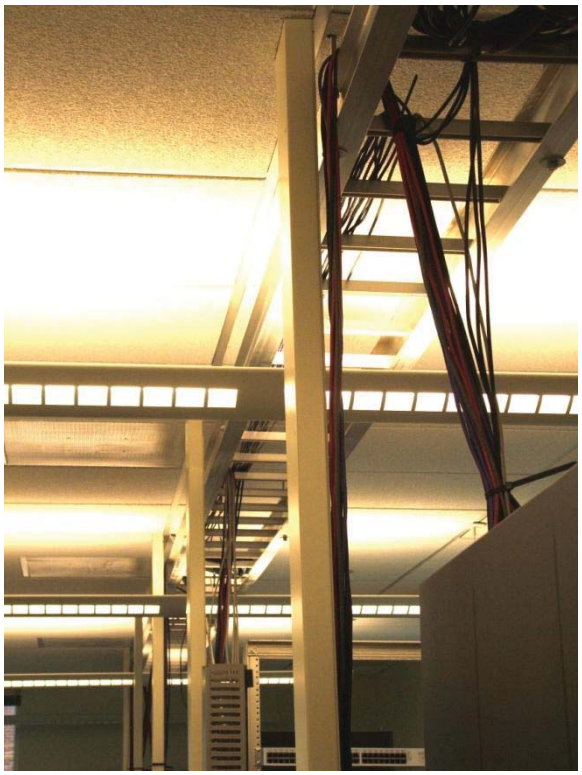

Illustration 4: The "ladder" carrying network cabling between rows of workstations. Note the bundle of cables rising obliquely from the server rack (right foreground) to the ladder, and vertical bundles of cables running from the ladder down to the trough below (not shown).

At the end of each row, network cables follow a riser to the ceiling, where they enter a horizontal ladder-like device (Illustration 4). This ladder runs perpendicular to the rows of desks, carrying network cables from row to row. Such cable-carrying ladders are common in newer construction; here, the difference is that the ladder is mounted below the ceiling so that cabling can be accessed without disturbing the ceiling tiles. The arrangement of troughs and the ladder permits re-routing and addition of network cables quite easily.

Despite all this, however, the goal of having an easily-reconfigurable network was not reached, because cabling to the network switches inside the racks was so congested physically that rewiring of the servers' network cables is very difficult. Each server has two network interfaces, but only one from each server has ever been used because of the difficulty of access. Experience has shown that it would have been better to put the network switches into a separate rack with patch panels, allowing rerouting without needing to work inside a congested rack. Fortunately, the flexible layout of the lab will allow this change to be implemented without difficulty.

4. LAB PROJECTS

In the Systems Administration and Security course, 50% of class time is spent in lab, with an additional three hours per week of required lab time. In the Computer Networks course, all lab time is outside scheduled class time.

As with much of the previously cited research, most of the lab projects utilize open-source software, which is free to download, sophisticated, and abundant. However, some of it, notably *OpenLDAP*, can be quite difficult to configure.

Systems Administration Course Projects

Because so many of today's students have never partitioned a disk or installed an operating system, partitioning and OS installation are among the first projects assigned. An alternate approach could be to have a technician perform the partitioning and installation, but, in the opinion of the authors, operating-system installation is an essential experience for future systems administrators.

A typical set of project for Systems Administration and Security in approximately sequential order is:

- Determine hardware configuration (CPU type, types and quantities of ports, monitor resolution, available I/O devices, installed RAM, etc.)
- Partition the hard drive and install Windows and Linux side-by-side (dual-boot) on the desktop computer.
- Install Windows server on two servers and Linux on the other server.
- Install all available updates.
- Install any necessary software and configure the software so that each server can be managed, using a graphical user interface, from either desktop OS, without use of the KVM switches.
- Use Bastille Linux to harden the student's Linux server.
- Set up the Linux server to be a DHCP server.
- Set up the Windows server to be a Windows domain controller.
- Set up the Linux systems to use the Windows domain controller for authentication.
- Set up the Linux server to be a Domain Name Server.
- Set up the Windows server as a file server.
- Write a bash script to extract student information from a mainframe report to create Unix accounts on the server for each student in the class.

It was hoped that students could go well beyond these basic projects. Additional projects considered included such things as testing password complexity, monitoring network communication, attempting intrusions, or configuring firewalls. However, our experience is that very few students finish all of the basic projects listed above.

Computer Networks Course Projects

Designing appropriate projects for a course that contains Computer Science, Information Systems, and Information Technology majors requires considerable thought. Computer Science majors ought to be writing network programs. Information Systems majors are generally not strong enough as programmers to write sophisticated software, and instead need experience with selecting and installing networking software. Information Technology majors have had considerable script-programming experience but have limited experience with general-purpose languages such as C++ or Java. All majors should get some experience working with existing network software.

Illustration 5: Rack of 30 1-U servers. Note the boxes of KVM switches above each cluster of six servers. The bottom three servers in the middle cluster have their escutcheons removed to allow access the optical drives and USB ports.

A possible solution to this problem is to assign alternate projects by major. There was hesitancy to do this, because some students might perceive that other students could get the same grade by doing “easier” work. Eventually, a solution was found to this quandary.

For part of their lab grade, students are given the choice of writing a server or installing and configuring several servers. To date, all CS majors have voluntarily chosen the software project, because writing one program is easier for them than doing several server projects. IS majors have all chosen the server-installation projects, and IT majors have split between the two.

IS and IT students cannot completely avoid programming, however. All students are expected to write a time server and a time client to coordinate the clocks on their desktop and servers. This software is relatively easy to write, and students of all majors have completed this project successfully.

To make the software easier to write and debug, the projects are to be written in Python. Experience has shown that network projects written in Python are much easier to implement than equivalent projects in C++ or Java, even though students in the course normally have no prior Python experience. This also gives students expertise in an additional language, thus helping to fulfill an ABET accreditation requirement.

The result is that everyone does (an) appropriate project(s) without being assigned their projects by major, and the difficulty of the programming projects are not so great as to preclude additional lab projects.

Typical lab projects for Computer Networks include:

- Install client and server operating systems (just as is done in Systems Administration and Security).
- Write a time server and time clients to coordinate the clocks on your servers and client.
- Set up an Apache Web server with a MySQL server feeding it data.
- Write a simple multi-threaded Web server, serving only GET and HEAD methods.
- Set up an anonymous FTP server.
- Set up a Jabber instant messaging system with both Linux and Windows clients.
- Set up Linux Samba as a file server for Windows clients.
- Use Linux OpenLDAP to authenticate Windows and Linux clients. (Tough!)

Other Projects

There are many other projects that have been considered for students who use this lab. The world of open source software includes many programs around which valuable projects can be built. While our students have not attempted many of these projects, some readers of this paper may find such projects worth considering:

- Set up printing for both Linux and Windows, using the same printer.
- Back up the servers using Bacula.
- Set up an iptables firewall on the Linux server. This can be simplified through the use of arlo-iptables-firewall, ferm, fiaif, guarddog, gnome-lokkit, kmyfirewall, knetfilter, or lokkit.
- Set up chillispot to operate a wireless hotspot.
- Use the following programs to attack machines and detect security problems:
  - Nessus
  - Snort
  - Crack and/or John
  - aide
  - rkhunter
  - tiger
  - tinyhoneypot, honeyd, or labrea
  - ettercap
  - fragroute, fragrouter
  - idswakeup and hping2
  - portsentry

5. HARDWARE PROBLEMS IMPEDE PROGRESS

As mentioned previously, students in both courses complete a surprisingly small number of lab projects. One reason for this is that sophisticated projects generally depend on more-basic ones, and cannot be attempted until the basic ones are completed.

Another problem has been a succession of hardware failures. Particularly surprising was the rate of critical hardware problems in this lab; this rate seems greater than in our professionally configured labs, which are used by large numbers of students daily. The technician expressed little surprise at this, stating that booting problems are common in computers that are not up and running continuously.

Hardware problems have increased over the four years that the lab has been in use, so that reliability during the fourth year was becoming a serious problem. Normally, the university replaces computers every three years, but upgrading this lab was delayed a year to put all servers in this lab into the same year of the replacement cycle. Campuses with cycles longer

than three years can expect diminished usefulness in such a lab beyond the third year.

Typical hardware problems included:

- Computer won't boot from the optical drive, preventing OS installation.
- Operating system installation fails for no apparent reason.
- Computer dead.
- KVM switch will not switch between desktop computer and server rack.
- KVM switch will not switch among servers.
- Hard drive not recognized.
- Network interface not detected during OS installation, and no network driver installed.
- Linux server software cannot negotiate video parameters with monitor over the KVM switching system.

Any one of these problems will delay a student considerably. While the technician has provided excellent support, the delays do compound and students fall behind. While it should be possible to install operating systems on three servers and a desktop dual boot Windows/Linux configuration within a few hours, or a few days for inexperienced students, our experience has been that often a significant minority of students does not have all of their computers configured one-third of the way into the semester, and much of the delay is caused by hardware problems.

That being said, it is also important to note that many students do rise to the occasion when problems develop. All students need to learn that hardware problems will occur, even in production facilities, and it is the job of the network or system administrator to find solutions in the face of such difficulties.

6. CONCLUSIONS

Adequate funding and proper planning are necessary but not always sufficient to guarantee a successful computer networking laboratory. Sufficient technical support is needed to keep facilities working, else students' experiences will fall short of expectations. Facilities must be upgraded regularly to assure lab success. However, good students will often rise to the occasion when hardware fails them, and they will find ways around their problems; this develops the kind of confidence that computing graduates ought to have. Overall, this lab has worked very well for the department and students, and it serves as a showpiece for visitors.

7. REFERENCES

Bullers, W., Burd, S., and Seazzu, A. (2006) Virtual Machines – An Idea Whose Time Has Returned: Application to Network, Security, and Database Courses. *Proceedings of the 37th SIGSCE Technical Symposium on Computer Science Education*, 102-106

Crowley, E. (2006). Developing "Hands-On" Security Activities with Open Source Software and Live CDs. *The Journal of Computing Sciences in Colleges*, 21(4), 139-145

Hill, J., Carver, C., Humphries, J., and Pooch, U. (2001) Using an Isolated Network Laboratory to Teach Advanced Networks and Security. *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, 36-40

Krap, A. (2004) Setting up a Virtual Network Laboratory with User-Mode Linux. Tech. rep., 2004. Masters programme on System and Network Administration, University of Amsterdam. Retrieved June 14, 2010 from https://www.os3.nl/~arjen/snb/asp/asp-report.pdf.

Kretzer, J. and Frank, C. (2005). Network Security Laboratories using Smoothwall. *The Journal of Computing Sciences in Colleges*, 21(1), 41-49.

Li, Pickard, Li, Mohammed, Yang, and Augustus (2008). A Practical Study on Networking Equipment Emulation. *The Journal of Computing Sciences in Colleges*, 24(2), 137-143.

Riabov, V. (2006). Challenging Projects and Virtual Labs in Web-Enhanced Networking Technology Classes. *The Journal of Computing Sciences in Colleges*, 21(6), 88-99.

Yuan, D, and Zhong, J, (2008). Designing a Comprehensive Open Network Laboratory Courseware. *The Journal of Computing Sciences in Colleges*, 24(1), 174-181.