Design and Implementation of a Multi-use Attack-Defend Computer Security Lab

Sergio Caltagirone, Paul Ortman, Sean Melton, David Manz, Kyle King, and Paul Oman*
University of Idaho
Department of Computer Science
Moscow, ID, 84844-1010
*(contact oman@cs.uidaho.edu; 208-885-6899) *

Abstract

This paper describes the rationale, design and implementation of a Reconfigurable Attack-Defend Instructional Computing Laboratory (RADICL). Topics covered include the ethical and pedagogical considerations for creating and using a lab that instructs and develops not only student's abilities to secure a system, but also their ability to attack a system. An in-depth description of the design and implementation and future work of the lab is also included.

1. Introduction

Analysis of pedagogical methods for teaching information security systems professionals has shown that a blend of theory and applied technique is needed to prepare students [5]. One of the ways to enhance the traditional lecture based teaching method at the undergraduate and graduate levels is to use isolated, experimental computer laboratories to focus students on practical security roles [7, 12]. Building on the work of Hill et al. [7], the additional feature of *high reconfigurability* is desired to give multiple courses access to the lab. A *reconfigurable* computer security lab has increased value to the academic department and encourages inter-departmental collaboration on security and survivability issues.

2. Academic Attack/Defend Exercises

2.1. The Need for Hands-On Laboratory Exercises

In most computer security classes defense is a central topic, either from the standpoint of confidentiality (e.g. cryptography) or availability and integrity (e.g. firewalls, intrusion detection systems) [3]. Many industrial recruiters assume that students in computer security acquire the requisite knowledge with respect to systems defense principles [2, 9]. However, many researchers have identified the need for additional instruction on the tools and techniques of systems attack [4, 6–8].

A Reconfigurable Attack-Defend Instructional Computing Laboratory (RADICL) is the next evolutionary step in computer security education. RADICL provides a safe venue for students to perform lab exercises such as, Capture the Flag, Worm/ Virus analysis, Denial of Service attacks,and network discovery to name a few. An in-depth discussion of laboratory exercises conducted within RADICL can be found in [10].

2.2. Ethical and Legal Concerns

When creating a lab of this nature, it is necessary to answer the relevant ethical and legal questions. In this case, there are two questions that need to be addressed; the first is whether it is ethical (and responsible) to instruct students in the tools and techniques of computer attack, the second question is what ethical and legal dangers are there in mistakes occurring with a lab that encourages dangerous tool use?

There are three positions with regard to systems attack instruction: first, that any attack, regardless of motive and purpose is unethical [11]. Second, there is a potential for some students to use the tools and techniques in an irresponsible manner — and therefore instructors should not take on the responsibly of teaching new hackers. The last position, and that of several authors, is that teaching attacks is a necessary component of a computer security education [4, 6–8].

The ethical obligation of an educational instution is to train students to the best of its ability. Students in computer security today should be trained to protect the nation's infrastructure of tomorrow. Denying computer secu-

*The authors gratefully acknowledge funding support through NSF SFS grants #DUE–0114016 and #DUE-0416757.

rity students practical knowledge of computer security, and "hacking" poorly prepares them for the work they will do in industry, government, and business. RADICL is a necessary part of preparing students at the University of Idaho for their careers. One cannot build or architect defenses for attacks that one does not truly understand.

The second question to address is of the legal responsibility if the lab were intentionally or mistakenly misused. The liability can be mitigated by educating the lab users on their ethical responsibilities. Pursuant to the US v Morris decision, an individual is liable for the accidental release of malicious software under the Computer Fraud and Misuse Act (18 USC 1030) [1]. There are basic precautions in place to protect networks outside of the RADICL lab. First, RADICL is an isolated, air-gaped network. Second, information does not leave the lab in any form other than hand written notes and printed paper. This prevents malicious code from affecting other machines. Our first line of defense is the moral and ethical education of our students. Ultimately any lab or network must rely on the ethical conduct of its participants. Ethical behavior is a mandatory part of our computer science curriculum.

3. A Survey of Attack-Defend Labs

Looking at Universities from across the nation, a few labs stand out:

- **Polytechnic University** — *Information Systems and Internet Security Laboratory (ISIS) used to research computer and network security, digital watermarking, and steganography. Reconfigurable through secure swappable hard drives and cloning them, or installing them directly into other machines. This design requires a physical reconfiguration of the laboratory.*

- **University of California, Berkeley** — *The DETER testbed allows the analysis and research of attacks and countermeasures.*

- **University of Pittsburgh** — *A conventional computer lab with 12 workstations, but the network can be reorganized into distinct testbeds to allow segregated research. They are also developing a curriculum that will educate the students with practical attack and defend exercises.*

- **Iowa State University** — *Working with the Department of Justice, created a large scale laboratory that simulates critical public and private infrastructures.*

- **University of California, Davis** — *A research lab primarily focused on intrusion detection systems, distributed denial of service attacks, and other network security concepts.*

- **Columbia University** — *Focuses on several areas of computer and network security research including practical cryptography, peer-to-peer intrusion detection, and automatic software patching, among others.*

There are numerous computer security related laboratories all over North America. However, labs that focus on in-depth education are rare; labs that are rapidly reconfigurable for multiple uses and distinct needs are rarer still.

4. RADICL Goals

4.1. A Student Project

The team that developed RADICL consisted of senior and graduate students in Computer Science, Computer Engineering, and Mathematics at the University of Idaho. All students were enrolled in a directed study class in the Fall semester of 2004.

RADICL was designed by students in small teams that met with the whole group to present their research. This method of organization prevented class discussions from being dominated by a small number of individuals, increasing the diversity of ideas. In this way, students were allowed to design a custom solution to an extremely open-ended problem: the need for a reconfigurable and multi-use attack and defend computer laboratory.

Beyond the design, students were also charged with procuring lab equipment, researching vendors, securing prices quotes, and meeting a specified budget. Students also physically assembled the lab and configured the network; this type of experience is normally only available in the professional realm.

Over the course of a semester, students designed and built RADICL. Students working on the lab gained a variety of practical experiences lacking in a typical CS curriculum. During the design phase of RADICL, students had the opportunity to analyze requirements and implement the laboratory.

4.2. Enabling Future Research

The final RADICL configuration meets the design criteria of easy reconfigurability and multiple use. The reconfigurability inherent in the lab's design comes from two major factors: the OS/image management and the keyboard video mouse (KVM) switch solution.

The image management of RADICL is flexible enough that every computer in the lab can have a new OS installed and running in less than ten minutes. The software configuration can be saved and redeployed at any time. The KVM solution allows a user to control multiple computers from a single user station. Consequently, a veritable cornucopia of simultanious research projects and classroom exercises can be facilitated by RADICL in a semester.

5. Designing and Implementing RADICL

5.1. Design Requirements

The following are the design requirements given to the students at the start of the RADICL project:

- Reconfigurability — The ability to change network topology and operating systems
- Multi-Use — Serving the needs of multiple classes and experiments
- Rapid Transition — Entire lab reconfiguration in less than ten minutes
- Modular Network — Partition lab into separate subnetworks for team exercises
- Isolated network — Prevent leaks of dangerous network traffic and software

In the following sections we describe the design and implementation of the lab in detail and provide justification as to how design decisions meet the above requirements.

5.2. Physical Configuration

For the uses envisioned by the class [10] two rows of eight desks facing each other is the optimal solution. Each desk has a monitor, keyboard, mouse, and KVM user station. The actual computers are rack mounted in one corner of the room. Additionally, the KVM switch is mounted and connected using CAT5e cable to each user station. A projector and pull-down screen are available for presentations.

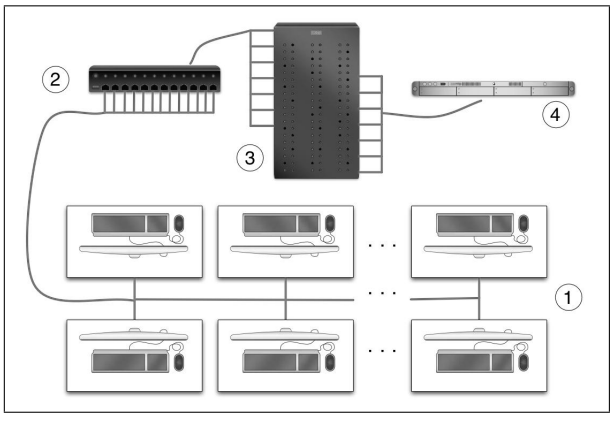

Figure 1: Overview of the Classroom Setup. User-stations (1) are connected to the machines (3) through a KVM switch (2) using CAT5e cable. Network connections between the machines are handled by a layer II/III network switch (4).

5.3. Hardware

The lab contains sixteen 2.4GHZ / 256MB / 80GB machines, four with dual network interface cards (NICs). There is also one 3GHZ Xeon server with 4×200GB drives in a RAID5 configuration to act as the image server — this is connected to an un-interruptible power supply (UPS). These components are divided onto two racks that are situated under an ventilation duct for cooling.

5.4. Network Configuration

The lab network satisfies the rapid transition requirements by enabling instructors and lab administrators to quickly change the lab's topology.

The lab uses a Cisco Catalyst 3550 Layer 3 switch. This switch provides us with 48 10/100 ports for future expandability and a 1000Base-T port directly connected to the backplane for a faster connection to the image server. The 3550 allows us the ability to partition the network into virtual local area networks (VLAN), configure spanning ports (to sniff traffic from multiple interfaces), set access control lists (ACL), and switch layer 2 and route layer 3 network traffic.

Aside from the routing switch, the lab also has a Cisco PIX 501 firewall to adapt the lab's network topology through placement of a firewall or VPN tunnel between subnets.

5.5. Disk Imaging

Disk imaging allows a "snapshot" of a partition to be made and reloaded on demand for any number of machines very rapidly. Because the grant did not allow for software purchases, students created custom disk imaging software using standard linux tools. Details on how our software operates can be found in Section 6.1..

5.6. Enterprise KVM

To provide the reconfigurability that is not available in most labs, an enterprise KVM solution from Raritan was employed. Using the KVM allows 16 user stations to access up to 128 machines using CAT5e cable. This allows each user to control as many of the lab machines as necessary for an experiment from one location — for example, two people could be playing "war-games" with several different machines without moving.

5.7. File Backup and Recovery

In any situation, especially one that facilitates experiments and classroom exercises, a method of backup and recovery must be created. In this design, there are two methods for file backup and recovery. The first is the the RAID image

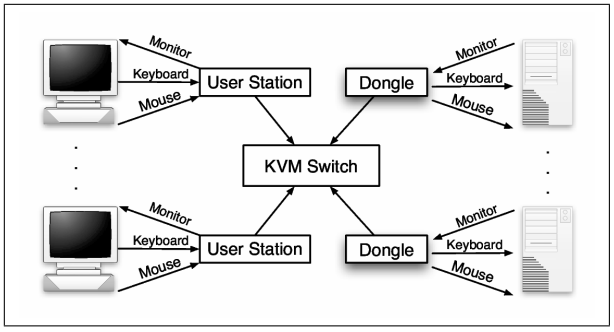

Figure 2: KVM Arrangement.

server which allows for limited drive failure without data loss. Secondly, a DVD+R drive allows the laboratory to create 4.7GB DVD disks for backup purposes.

6. Using and Administering RADICL

One of the biggest design decisions, reconfigurability, also created the biggest challenge from a technical design and implementation perspective. Because it is necessary to support many different network configurations, and operating systems, none of the open source software would allow the flexibility of image and computer management that was needed.

For this reason, we decided to write our own management software for the laboratory. We developed the management software using GNU/Linux tools for their well-known interoperability. The end product was a series of shell scripts on the back end and a dynamic web administration front end. These programs control the creation and deployment of software images for client machines and network reconfiguration. Without any previous technical knowledge of RADICL, faculty and students can use the graphical front end to easily reconfigure the room for custom experiments. There are two primary funtions performed by users of the system: Creating images/ network configurations and deploying these configurations. Examples of these are described in the next sections.

6.1. Initializing a Computer

Any "RADICL compliant" computer must be initialized. Initialization configures the machine to work with the custom written RADICL software. Re-initialization of a computer can be done at any time and simply requires the user to enter a few commands from the bootable OS on CD (LiveCD) command prompt.

The RADICL script first partitions the hard disk into 4GB segments. 4GB is a compromise between size and usability. Almost any OS can install to a partition of 4GB and the size isn't such that unzipping and downloading the image would take more than a few minutes.

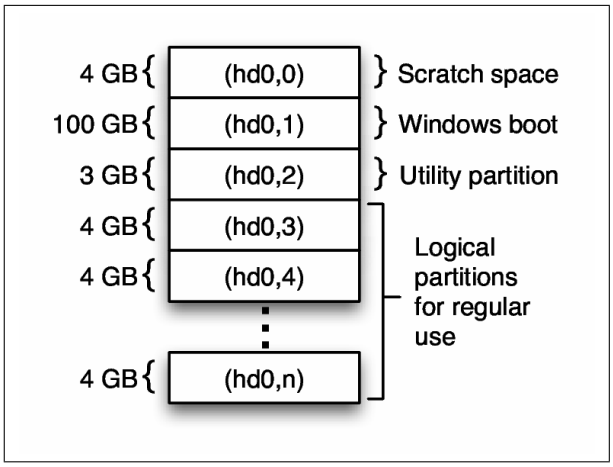

Figure 3: The Partition Table Layout of a Lab Machine After Initialization.

Second, the initialization script installs a utility partition. This partition can be booted anytime to allow the image server to synchronize the operating systems on the machine with the web interface. The utility partition also contains the scripts needed to patch and upload new operating systems.

Third, because of a need to multi-boot the machine into any of the OSes installed on any of the logical partitions, there existed a need for an interactive boot loader. The GNU Grand Unified Boot Loader (GRUB) was used to accomplish this requirement. GRUB allows the operator to select which partition to boot from during the computer startup sequence. However, in order to maintain knowledge of what OSes are installed on each partition, GRUB must have access to boot configuration information stored on disk. This boot configuration is stored on the utility partition.

To initialize a lab machine, the user first starts the computer from a bootable CD. The user then initializes the network interface and uses FTP to connect to the image server and download the "master" script. Running this master script with the single parameter of `default` starts a process that connects to the server, downloads all necessary tools and then partitions the hard drive. It also pulls down images for the Windows boot and utility partitions and applies them to partitions (hd0,1) and (hd0,2) respectively. Finally, the boot configuration file is created and the MBR is initialized. After this process is complete ($\sim$5 minutes) the machine is completely initialized.

6.2. Creating an Image

The process of creating an image is one of installing an OS (or using an existing OS image) and customizing it to

fit the needs of the experiment. After installation and configuration is complete, the task of preparing the image for deployment takes several steps.

The user first reboots the machine into the utility partition where the rest of the process will take place. From within the utility partition the newly configured partition is modified to make it partition and machine independent. At the same time, a configuration script is written to apply machine and partition specific information back on to the image when it is deployed in the future. The final step in image preparation is to create a small snippet of GRUB configuration for the configured image.

When all this configuration is done, the master script is once more invoked, this time with the `push` parameter. This process then extracts the contents of the partition that was configured, bundles up the special configuration the user just created and pushes to the image server.

One final step requires the administrator of the image server to transfer the image from the "incoming" to the "published" directories on the server. This step is required as a security and policy measure to ensure the integrity of those images that are deployed.

6.3. Deploying an Image

The process of deploying an image for a session of experimentation is anticipated to be "the most common user" interaction, and thus it has received the most automation and interface emphasis.

From a web page interface on the imaging server, a user is shown the current configuration of every machine in the lab. This information includes which OS are on each partition, who installed them there, why, and if they are considered to be a dangerous install. Dangerous installs are installs where there was a chance of worm propagation, virus distribution, or any such nefarious process that might pose a danger to other machines or experiments.

After viewing the configuration, the user can chose to change the configuration of any machine or machines. All machines that were changed, are then physically rebooted into the utility OS in a special "auto-configure" mode. Auto-configure mode polls the imaging server for any changes the users may have indicated. If changes are found, the system automatically downloads the necessary images and applies them to the correct partition without any user interaction. After all changes are applied, the machine automatically reboots.

Performance testing has shown that a typical OS install, without many additional utilities, compresses roughly to a 500–1000MB file stored on the image server. The process of pulling this image to the client takes approximately four minutes. This process is IO bound, specifically it will go as fast the client hard drive can write data to disk.

6.4. Administration

The imaging server is the foundation for the lab, and as such it deserves some description and explanation. The imaging server runs the Linux kernel and supporting utilities as packaged by Gentoo. Three network services run on the imaging server: an HTTP server (Apache) with PHP scripting capabilities, a relational database (MySQL), and a high-performance FTP server (vsftpd).

The only service that is network accessible is the FTP server. Both the HTTP server and the relational database are used for the user interface for reconfiguring the lab as well as hosting lab documentation in a wiki format (MediaWiki). Communication between the image server and the lab machines are all handled through file transfer, this makes the amount of server administration very minimal for the class.

This administrative configuration attempts to maximize academic flexibility, while also providing adequate assurance against network or physical abuse. Physical access to the room is limited by card swipe and keyed entrance. Machine access is controlled by the Raritan KVM which has its own access control lists allowing us to grant certain classes access to only certain user stations and machines. Furthermore, administrative access to the image server functionality is protected through traditional username and password management

6.5. Classes and Projects

RADICL is in high demand by teachers and project teams wanting to utilize the unique abilities of the lab. Basic computer science courses have used the lab for its ability to demonstrate multiple operating systems with quick system switching. Advanced computer security courses find the lab invaluable for its complete administrator access to the machines used. There is no other venue at the University of Idaho were upper division computer security classes are allowed to perform potentially dangerous security lab experiments [10].

The University of Idaho's network security class has utilized the lab's reconfigurability to allow students to execute a SAMBA exploit, install a backdoor, deface a website and then analyze the traffic they generated. These and other lab exercises are documented in [10]. The KVM and Cisco switch enabled that lab to be run isolated from and in unison with a Department of Defense sponsored senior design project being developed on machines not used in the simulation.

The senior design project uses RADICL computers to cut down on set-up time and to back up work done that isn't easily recreated such as creation of domain controllers and certificate authorities. Th Design process benefits from RADICL because it provides complete control over machines and re-imaging after major problems.

7. Challenges and Future Work

Within a modest budget (<$50,000), acquiring enterprise level KVM and network equipment and other hardware flexible enough for the design goals required numerous design iterations and price/performance tradeoffs.

An ongoing research and implementation task is determining operating system support for reconfigurability. Each operating system has its own idiosyncrasies with respect to partition independence, network setup, and unique host identification.

There are a number of opportunities for future work in RADICL. The first is to acquire additional operating systems and platforms (i.e. Apple, Sun, etc.). The central control system and graphical interface can be further extended. Additional work can also be done with regard to the saving of lab state. Specifically, the students would like to see the ability to save the entire state of the lab for future redeployment and have the ability to replay certain activities for future analysis.

8. Summary and Conclusions

Computer security issues are becoming the forefront of a computer science education. Even so, actual hands on experience in computer security remains virtually nonexistent in academia.

The ethical concerns of teaching students "hacking" are dwarfed by the need for knowledgeable, competent, and, above all, experienced computer security professionals in industry and government. RADICL was created to remove this gap between theory and practice for students at the University of Idaho.

RADICL was designed to allow students to do attack/defend exercises that exercises require a variety of operating systems, network topologies and the ability to control many computers from one user station. The lab also allows students to have full administrative access to the machines.

The lab has to remain highly reconfigurable. An exercise might require every operating system in the lab to be changed or re-initialized to some preset configuration. Toward that goal, a sophisticated but simple imaging scheme was designed by which users can upload an image of any partition to a central image server where an administrator approves it for safekeeping.

These images can be downloaded and deployed on any computer in the local network in a matter of minutes. In this way, the lab will be used as a dedicated classroom for a small number of network and computer security courses while remaining open for research groups and attack/defend exercises.

A lab like RADICL, dedicated to giving computer science students the resources they need for hands on computer security, should be considered by every academic institution that commits itself to producing computer security professionals.

References

[1] *United States v. Robert Morris*, ser. Federal Reporter. United States Court of Appeals for the Second Circuit, 1991, vol. 928.

[2] S. F. Barnett, "Computer security training and education: A needs analysis," in *1996 IEEE Symposium on Security and Privacy*, Oakland, CA, 1996, pp. 26–29.

[3] M. Bishop, "Teaching computer security," in *Ninth IFIP International Symposium on Computer Security*, 1993, pp. 43–52.

[4] ——, "The state of infosec education in academia: Present and future directions," in *National Colloquium on Information System Security Education*, 1997, pp. 19–33.

[5] ——, "Computer security education: Training, scholarship, and research," *IEEE Computer*, vol. 35, no. 4, pp. 30–32, April 2002.

[6] M. Bishop and D. Frincke, "Who watches the security educators?" *IEEE Security and Privacy*, vol. 1, no. 3, pp. 56–58, 2003.

[7] J. M. Hill, C. A. Carver Jr., J. W. Humphries, and U. W. Pooch, "Using an isolated network laboratory to teach advanced networks and security," in *32nd SIGCSE Technical Symposium on Computer Science Education*. Charlotte, North Carolina, United States: ACM Press, 2001, pp. 36–40.

[8] P. Mullins, J. Wolfe, M. Fry, E. Wynters, W. Calhoun, R. Montante, and W. Oblitey, "Panel on integrating security concepts into existing computer courses," *SIGCSE Bulletin*, vol. 34, no. 1, pp. 365–366, 2002.

[9] NSTISSI-4011, "National training standard for information systems security professionals," June 20, 1994.

[10] P. Ortman, "Designing and building a rapidly reconfigurable attack-defend instructional computing laboratory," Masters Thesis, Dept. of Computer Science, University of Idaho, Moscow, ID, 83844-1010, 2005.

[11] E. Spafford, "Are computer hacker break-ins ethical?" in *Computers and Ethics in the Cyberage*, D. M. Hester and P. J. Ford, Eds. Upper Saddle River, New Jersey: Prentice Hall, 2001, pp. 332–344.

[12] W. Yurcik and D. Doss, “Different approaches in the teaching of information systems security,” in *Information Systems Education Conference (ISECON)*, Cincinnati OH. USA, November 2001.