

Design of A Virtual Laboratory for Information Assurance Education and Research

Vikram Padman and Nasir Memon

Abstract— One of the main impediments to establishing an IA program is the requirement of a laboratory facility that will reinforce concepts taught in class with hands-on experiences. This is due to the fact that an IA lab is difficult to build and maintain as it needs to be dedicated and isolated and cannot be part of a general purpose campus laboratory. Many schools cannot afford a separate laboratory just for an IS course. In this paper we present the design of a *virtual laboratory* that will allow multiple institutions to share one physical laboratory. This design was done as part of an NSF capacity building project to establish a centralized laboratory facility at Polytechnic that can be used by schools in the tri-state area surrounding NY City. By virtual laboratory, we mean a laboratory that can be accessed via the internet through a browser interface. In addition to being remotely accessible, the virtual laboratory is also remotely configurable, thereby allowing each individual member of the consortium to independently provision the required hosts and network components and configure them as needed for the specific hands-on assignment being performed by their students.

I. INTRODUCTION

The three most important resources that are needed for establishing an education program in Information Assurance (IA) are:

- Course material,
- Faculty expertise,
- Laboratory facilities.

There is plenty of courseware in computer and network security that is available today. For example, The National Colloquium for Information Systems Security Education (NCISSE) facilitates sharing of knowledge and resources through its web sites [8] which currently contain course materials on Ethics in Computing, Risk Management, and Malicious Logic. A more comprehensive resource is provided by the National Information Assurance Training and Education Center (NIATEC) [10] at the University of Idaho, in the form of teaching and curriculum materials available on Quarterly CD-ROMs to all participating institutions. University of Tulsa spearheads an effort to disseminate extensive course material in Digital Forensics. Polytechnic University itself makes available its lecture notes and lab assignments via the web and these have been used in some form or by various universities.

Computer Science Department, Polytechnic University, Brooklyn, NY 11201. Author email addresses are: vikram@isis.poly.edu and memon@poly.edu. This work was supported by an NSF capacity building grant NSF 0417048

Similarly, there are quite a few programs now for providing training to faculty in IA. Purdue, Tulsa and Idaho have well established high quality programs to address this need. Polytechnic University has hosted the Cisco Bootcamp in IA for faculty and plans to continue holding one every semester. In fact, Cisco has been holding these bootcamps across the country almost on a monthly basis [3].

However, motivated faculty armed with the requisite training and courseware still face the hurdle of establishing a laboratory to help deliver their IA courses. Often they are not able to provide a laboratory component in their courses due to the special needs of such a laboratory and/or due to budget and infrastructure constraints. The latter is especially true for smaller institutions like community colleges. As a result many schools either do not offer a course in IA or offer one without a laboratory component, thereby significantly reducing the quality of their course and diluting the learning experience they are able to impart.

In this paper we present the design of a *virtual laboratory* that will allow multiple institutions to share one physical laboratory. This design was done as part of an NSF capacity building project to establish a centralized laboratory facility at Polytechnic that can be used by schools in the tri-state area surrounding NY City. By virtual laboratory, we mean a laboratory that can be accessed via the internet through a browser interface. In addition to being remotely accessible, the virtual laboratory will also remotely configurable, thereby allowing each individual member of the consortium to independently provision the required hosts and network components and configure them as needed for the specific hands-on assignment being performed by their students. The idea of establishing such a laboratory was inspired by a similar facility used by Cisco systems in their IA Bootcamp for faculty they regularly conduct across the nation. However, to the best of our knowledge, the Cisco facility does not provide remote configurability to the extent that we envision and desire and hence is not suitable for hosting a variety of institutions teaching different courses and using the laboratory in very different ways.

II. A VIRTUAL LABORATORY FACILITY

It has been recognized for some time now that education in IA is better served by a laboratory component that reinforces principles and theoretical analysis learnt in the class

room with a follow-up hands-on component performed in an appropriate laboratory. However, a significant number of programs continue to teach IA in the decades old traditional framework, focusing solely on theoretical principles and their analysis. Although theoretical concepts are essential and need to be taught, it is very important to also show students how to apply the theory they have learnt in very different and important practical situations. Hence, a good part of an IA course should also focus on applications and operational concerns. In order to do this, a supporting laboratory becomes necessary.

Recent years have seen an increased awareness on the importance of a laboratory component in IA education [6], [2], [5], [7]. Irvine [6] points out that securing a system requires a “marriage” of good science and engineering and that engineering components are best taught by reinforcing concepts taught in the class by hands-on experiences in the laboratory. She further points out that just as it is unreasonable to expect a student to learn programming only by reading about it, it is also unreasonable to expect students to learn “security engineering” solely from discussions in the class room. Similarly, Hill [5] and Mateti [7] also make the case for laboratory based instruction in IA and in fact provide detailed examples of specific courses and lab projects that accomplish this goal. A laboratory for IA education can be designed in a different manner depending on the nature of the program and the course being serviced. However, there are certain general principles that guide the design of such a laboratory. Specifically, a well designed laboratory should possess the following characteristics, as first listed in our earlier paper [1]:

- *Reconfigurable*: The lab should be highly flexible and re-configurable. Different topics and assignments require different operating systems and/or network topologies and it should be possible to change the configuration of hosts and networks easily and efficiently.
- *Heterogeneous*: The lab should comprise of multiple platforms from multiple vendors. A lab with homogeneous environment does not effectively train students to cope with real world situations.
- *Scalable*: The lab should be scalable and should be able to sustain many students, and still have enough duties for each student to handle. Student groups should not get large due to lack of resources.
- *Cost Effective*: The cost of setup and maintenance of the lab must be far less than what’s being simulated by the lab. For example, the lab should effectively simulate a small to medium enterprise network but the cost for building and maintaining the lab should be far less than the cost of a moderate enterprise network.
- *Robust*: The lab should be able to sustain and handle inadvertent damage by the students. For example, it should be possible to quickly recover the set-up and configuration of a host node even after a student accidentally causes a

malicious program to erase the hard disk.

- *Maintainable*: The lab should be easy to maintain. Routine tasks like back-up and application of software patches should be easy to perform and automated to whatever degree possible.
- *Realistic*: The lab should provide practical and first hand experience to students in a network environment that is close, in terms of complexity, to a network that they might encounter in a real world enterprise.
- *Insulated*: Activities in the lab should not affect traffic on the campus network. There should be sufficient amount of separation and isolation enforced between the lab network and the external network. The presence of the lab should not be a cause of concern to campus network authorities.

A. THE VIRTUAL LAB CONCEPT

The Information Systems and Internet Security Laboratory (ISIS) at Polytechnic University is a facility that was designed with exactly the above criteria in mind. The design and implementation of ISIS is described in more detail in the next section. ISIS was established in 1999 based on an NSF CCLI grant. Subsequently it received additional funding for expansion from an NSF SFS capacity building grant in 2002 and a generous Cisco equipment donation in 2003. ISIS is now one of the premier laboratories of its kind in the nation.

Ideally, it would be desirable to have an ISIS like facility in every university teaching IA courses. However, this is clearly impractical. The substantial cost that was incurred and the time that was spent in building ISIS cannot be easily replicated in many institutions. This is especially true for institutions that do not have a full fledged IA program but teach only a course or two in IA.

An alternative then is to build a central facility hosted in one institution and allow multiple institutions to make use of the facilities remotely. This is what we call a *virtual laboratory*. Such a laboratory would enable sharing of resources and facilitate the inclusion of a lab component in universities that do not have a sufficiently comprehensive IA program. It can also be used as a “wetting ground” for universities that are starting an IA program and do not wish to make the upfront investment in a full fledged IA laboratory before their program takes root. They could instead start their program using a centralized and shared facility and slowly develop and move to their own facilities as their program grows in size and gains sufficient resources.

A virtual lab is a facility that provides a remotely accessible environment to conduct hands-on experimental work and research in information systems security. In its simplest form a virtual laboratory could simply be a network of hosts that are remotely accessible to students or member institutions. For example, one of the facilities provided by ISIS laboratory is convenient remote access to our entire test bed for students. However, this alone would not

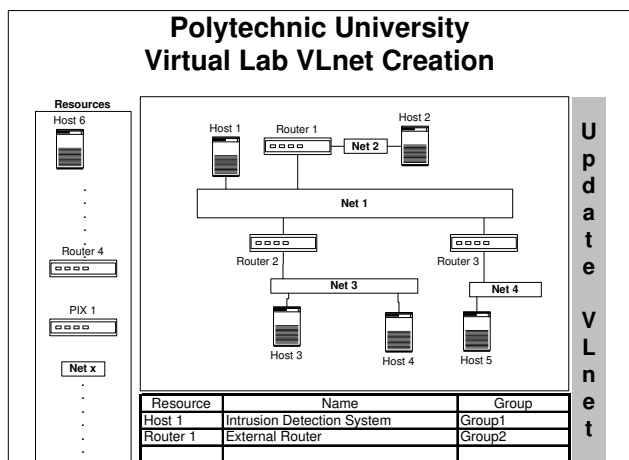


Fig. 1. View of VITAL to an Instructor

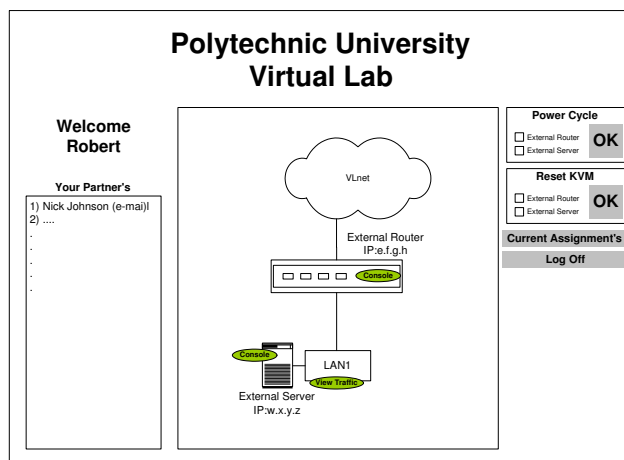


Fig. 2. View of VITAL to a Student

conform with many of the characteristics of an effective laboratory in IA that we listed in the previous subsection. For example, it would not be flexible enough to provide a realistic environment for hands-on experimentation in a variety of IA topics.

The virtual laboratory that we envision would allow an instructor to create virtual networks and allow students to access these networks through a secure connection (SSL or VPN for example) over the Internet. The virtual network and the hosts and network elements in it could themselves be configured according to the needs of an assignment. For example, an instructor should be able access the virtual lab resource repository, identify resources needed for setting up a specific network configuration and construct one remotely. So for example, suppose an instructor wants to assign an experiment that teaches students how to configure a firewall. To do this, she would access the virtual lab configuration page. Using the web interface she would be able to create a network consisting of many subnets, each subnet consisting say of a host and a firewall/router that protects the host in the subnet. She could then authorize a group of students to each subnet and they would access it through the same web interface. Figure 1 shows how the interface to the virtual lab could potentially appear to an instructor.

Continuing the example, the students would now be able to configure the firewall according to the instructions in their assignment. They would then be able to test the firewall by activating a node that is outside their subnet and send attack traffic to their subnet. The students would be able to get a log of the traffic coming into and out of their network by activating another sniffing component in their subnet or host. Figure 2 shows a possible student interface to the virtual laboratory.

B. DESIRABLE CHARACTERISTICS OF A VIRTUAL LAB

In addition to the characteristics of an effective IA lab described in the previous subsection, an effective virtual laboratory should also demonstrate the following characteristics:

- *Accessibility:* Students should be able to access a virtual lab in a manner similar to how they would in a real situation.
- *Observability:* Most laboratory exercises are in the form of experiments in which students need to observe facts or results of the experiment they have done. A virtual lab should not diminish the ability of a student to observe host and network events.
- *Ability to simulate realistic scenarios:* Virtual nature of the lab should not diminish the ability of the lab to simulate realistic security scenarios. Furthermore, virtual nature should be transparent to the students and provide a view as close as possible to a real network.
- *Realistic:* A virtual laboratory should have the capacity to host realistic networks and devices.
- *Separability of virtual networks:* Many participating institutions may be using the virtual lab simultaneously. However, each virtual network must be isolated such that the events on one virtual network has no effect on the others. This separability is necessary to maintain a reliable test environment.
- *Remote Configurability:* A virtual lab should provide flexible mechanisms to allow each instructor to configure their virtual networks remotely with very few, if any, interactions with the host institution.
- *Ability to share resources efficiently:* Many participating institutions may share the resources of a virtual lab. Resources in the lab must be provisioned such that the sharing of any resource does not affect the properties mentioned above adversely.

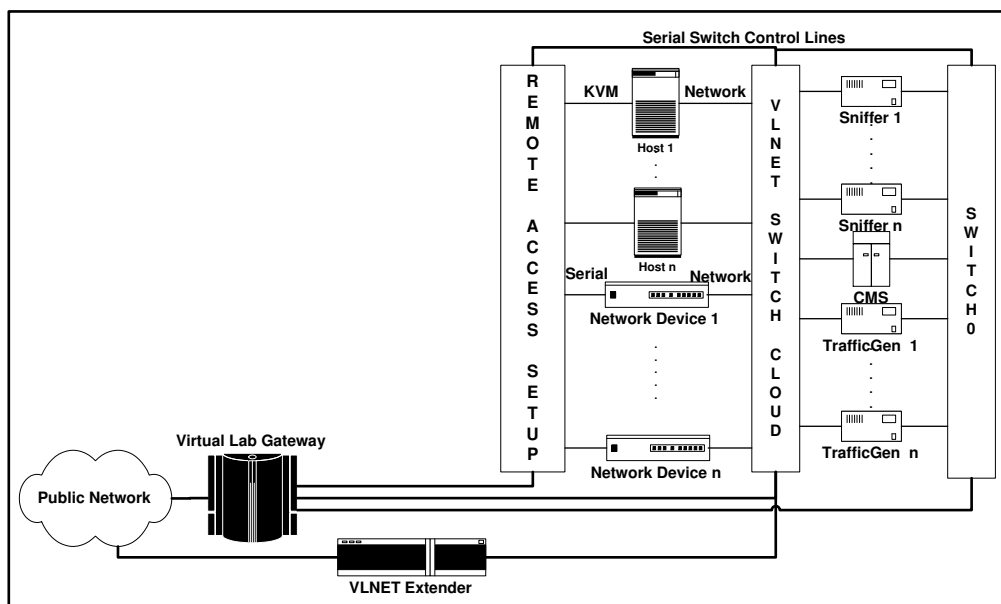


Fig. 3. VITAL Architecture

C. DESIGN OF VITAL - A VIRTUAL LABORATORY FOR INFORMATION ASSURANCE

In this section we present the design of *VITAL* - a Virtual Information Assurance Laboratory, that meets, to a reasonable degree, most of the design goals of an efficient and effective IA virtual laboratory listed above. The architecture of *VITAL* is depicted in the Figure 4. The major components of *VITAL* shown are:

- *VLnet*: is a set of networking resources allocated to a partner from the *VITAL* resource pool. Each partner can create one or more virtual networks remotely using the resources provisioned to them.
- *Switch 0*: is a simple switch that provides connectivity among virtual lab gateway, sniffer pool (sniffer 1, ..., sniffer n) and "client machine" server. This switch is not a part of VLnet and controlled by the host institution to extend the capabilities of *VITAL*.
- *VLnet Extender*: is VPN a concentrator that allows several virtual labs to connect with each other securely over public network. The extender allows for sharing a partner's resources with that of *VITAL* to accommodate any special hardware or software requirements not supported by *VITAL*.
- *Client Machine Server (CMS)*: is a server that hosts one or more client machines. A client machine simulates the internet user; it acts as an internet client which accesses the hosts in VLnet. These machines are hosted by CMS, which runs VMware ESX server software, and can simulate Windows, UNIX, and Linux clients.
- *Sniffer 1, ..., Sniffer n*: In additions to the hosts (see below), these machines are used to sniff the network traf-

fic of an entire VLnet. This provides for observability of networks. The number of sniffers required depends on the limitation dictated by the switches in the switch cloud.

- *Traffic Generator 1, ..., Traffic Generator n*: Traffic generators are used to create realistic network traffic, like web traffic or route advertisements. In addition, the generators can also be used to flood a network if it may deemed necessary by an instructor.
- *Host 1, ..., Host n*: Hosts are general purpose x86 machines capable of running various operating systems. A host can be real x86 machines, or a virtual x86 machines. Virtual Machines are hosted by VMware ESX server. Each host has more then one network interface connected to a switch in the switch cloud. Each host in the virtual lab can run any operating system that supports x86 architecture. An instructor could specify any OS supported by the virtual lab facility for a host or all hosts provisioned to her and the requested OS will be loaded automatically when the host power cycles. The virtual lab gateway stores master images of all supported operating systems, when a host power cycles it checks with the virtual lab gateway, if its OS needs to be reloaded with a new OS then the virtual lab gateway sends a copy of the master image to the host.
- *Network Device 1, ..., Network Device n*: A Network Device is either a router, or a firewall, for example: CISCO 2600 series router or PIX 5XX firewall. Each network device is connected to a switch in the switch cloud.
- *VLnet Switch Cloud*: is the core of the VLnet, which can be configured remotely by an instructor to create the logical networks needed. The switch cloud is made up of interconnected switches to which hosts and network devices

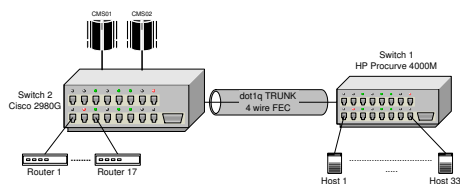


Fig. 4. Example VLnet connection

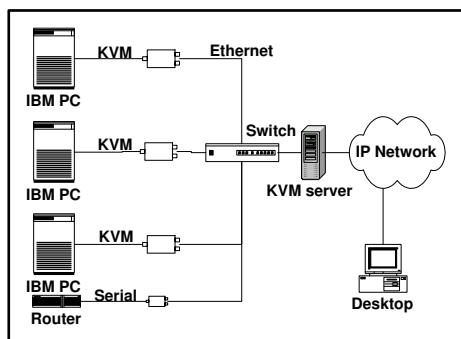


Fig. 5. KVM over IP

are connected. The number of switches in the switch cloud depends on the number of hosts and network devices and the capacity of each switch.

- *Remote Access Setup*: is a collection of KVM and serial devices. KVM stands for Keyboard Video and Mouse. KVM is needed to gain access to a host's console. We will be using KVM over IP to provide remote console access to hosts and serial console over IP to provide remote access to network devices. KVM and serial over an IP network will allow multiple users to access the computers simultaneously through a web browser or some client software. In case of virtual machines, we will be using VMware remote console software to access virtual machines over IP network.

- *Virtual Lab Gateway*: is the heart of *VITAL*. It provides a user friendly web interface to access VLnet, Remote Access Setup, Sniffers, Traffic Gens, and CMS. It also allows the instructors to configure VLnet hosts and network devices.

There could be any number of hosts and network devices depending on the size of the virtual lab facility being built. The only constraint is the available budget. The number of hosts and network devices dictates the number of switches in the cloud, and number of KVM/Serial lines devices needed to build Remote Access Setup. In other words, the total cost of the virtual lab is based on the number of hosts and network devices it contains. Hosts and the network devices are shared with other universities based on their class size and available resources.

Although *VITAL* is mostly made of hardware components off-the-shelf some in-house software development is

necessary for user friendly remote configuration of VLnets, resource and user management, lab scheduling, Virtual Lab Gateway, and web interfaces for students, instructors, and administrators.

D. AN EXAMPLE

How a user of *VITAL* can access and configure a virtual network could be best explained by means an example. Let's assume that the following VLnet is already configured for a certain lab assignment as shown in Figure 6. Where *R1* is an external router, *R2* is the internal router, *ES* is an external server, *IN* is internal network, and *IS* is internal server. We are assuming that the class has been divided into four groups and each group representative has control of the network which they are responsible for. Now when a user from a particular group logs on to the virtual lab gateway what s/he would see in the web browser is shown in Figure 7. The user would be able to access the console of their hosts and network devices by just clicking the "Console" icon that appears near the device, and can observe the network traffic in DMZ and internal network by just clicking the "View Traffic" icon. They will also have an option to save the viewed traffic to a file for further analysis. When an instructor logs in s/he would be able to see the whole virtual network, would have an option to change the VLnet options, and be able to manage users. The options for instructors may vary depending on the needs and implementation constraints of *VITAL*.

III. LAB EXERCISES AND EXPERIMENTS

The laboratory design that we have described facilitates a rich variety of lab exercises and experiments. In this section we describe some typical exercises that are done in a Network Security class and briefly discuss how they could be done in a virtual laboratory, in pretty much the same manner as can be done in an isolated physical laboratory like ISIS. We group the assignments by categories, and discuss them in logical order. Typically, the class would progress from exercises in one category to the next. Similar sets of exercises could be designed and listed for other IA courses like computer security, digital forensics, etc.

1 Network auditing and fingerprinting tools:

A typical class of exercises initially done in a network security course are usually aimed at exposing students to various UNIX and Windows network configuration and monitoring utilities. Expertise in using these utilities are basic requirements for performing more complicated exercises in network security. We can further subdivide this category of assignments into five categories as shown below:

1. *Network setup and monitoring tools*: Exercises on using various network configuration and monitoring utilities such as ifconfig, ping, arp, netstat, ipconfig, nbtstat, . . . , etc.

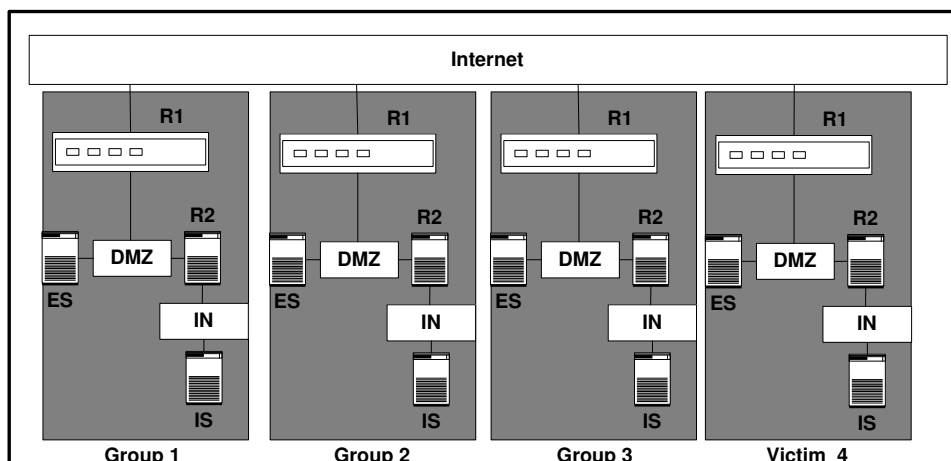


Fig. 6. A Simple VLnet Configuration

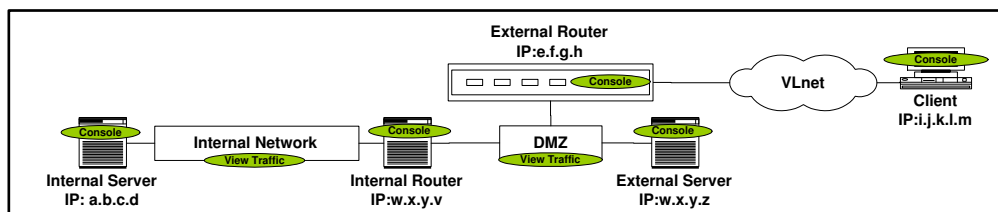


Fig. 7. A User's View of Her Network

2. *Understanding network auditing tools:* Exercises on using various network auditing tools such as tcpdump, traceroute, nmap, nesses, . . . , etc.

3. *Writing network applications:* These category of exercises typically serve to introduce network programming to students. The complexity of the code written would depend on the level of the class and the time assigned to perform the task. We give a couple of examples below.

4. *Packet capture and analysis:* In these class of exercises students are asked to write a packet capture and/or analysis tool. For example, they could be asked to write a sniffer to capture and decode ethernet packets. Another example is to write a packet analysis tool which takes some standard sniffer output and captures cleartext passwords, http URL's, or to make a note of packets with a specific string. In general, the objective is to teach students how to use packet capture libraries. Due to time constraints these type of exercises may typically give students a "skeleton" code and ask students to develop the rest.

5. *Traffic generation and shaping:* Traffic generators and shapers are used for testing and evaluation of security application and systems. In these relatively advanced type of exercises students are given a chance to explore these tools and use them to simulate and/or test .

All the above types of assignments can be easily carried out in a virtual lab with little compromise on the quality

of the learning experience imparted. Typically for each of them, an instructor would have to set up a network with appropriately configured *nix machines and assign a machine to each student or group of students. Specifically, in *VITAL* , using the remote configuration utility depicted in figure 1 an instructor can simply create a network with hosts allocated to her. *VITAL* also allows the instructor to choose what operating system each machine would use. When a student logs into *VITAL* she will see the machine(s) she could access and by clicking on the console button on host she will be able to access its console.

2 Exploring TCP/IP vulnerabilities:

An essential part of any network security course involves examining the strengths and weakness and solutions of the TCP/IP suite of protocols. In such experiments students typically explore vulnerabilities of ARP, PING, ICMP, TCP, UDP and various routing protocols such as RIP, OSPF, and BGP.

This class of assignments requires at least a small hierarchical network similar to the one depicted in figure 6. Usually a lab administrator would have to manually configure the IA lab to create this network. In *VITAL* , the instructor can simply draw the network and upload the configuration file and the network will be created accordingly.

3 Exploring common vulnerabilities in client/server applications:

Most of the application that are being used today over the Internet are based on client/server computing model. In this category of experiments, students learn about some popular client/server applications and their vulnerabilities.

1. *Buffer overflow and format string attacks:* Client/Server applications are subjected to wide varieties of attacks, but amazingly 90% of these attacks are due to software vulnerabilities such as buffer overflow and format string attacks. Hence it is important for a security professional to master the nuts and bolts of buffer overflow and format string attacks over the network. In this type of exercise, students write exploit code to attack a vulnerable service deliberately planted by the instructor. In the process they get to learn the nitty gritty of a buffer overflow like attacks and how it can be used over the network.

2. *Application vulnerabilities and exploits:* Vulnerable distributions of applications such as HTTP, FTP, telnet, SSH, SQL server, SSL, and DHCP are installed in a server and students are asked to find the vulnerability and prove it by exploiting the vulnerability. Attacks on distributed system such as DNS can also be explored.

In the above category of assignments, the same hierarchical network used in the previous category could be used here. Except some machines have to now be configured with services such as http, ftp ... etc. In *VITAL*, all an instructor would have to do is right click on a host, for example, and select the services that would be needed on that host.

4 Security defence mechanisms:

The types of exercises described up to this point were more about vulnerabilities in networking systems and applications, and how they can be exploited. We now describe exercises where students are exposed to security tools designed to protect against attacks on vulnerabilities in the network. Typically students are asked to install and configure hardware and software including :

1. Firewalls
2. Proxies
3. Intrusion Detection Sensors
4. Network forensics tools and analyzers
5. Network security testing tools (traffic generators and analyzers)

Depending upon the level of the course and available time students may also be asked to do the projects such as:

1. Integrating IDS and firewalls
2. Anomaly based IDS
3. Specialized traffic analysis engines
4. Network forensic tools

This class of assignments requires network devices such as routers and firewalls which can be added to the hierarchical network created using *VITAL*. Again, using the *VI-*

TAL configuration utility all an instructor has to do is to drag the network device she need from the resources available to her and connect them appropriately. The student who are allowed, by the instructor, will have full access to the network device.

5 System auditing

A secured OS is an important component in a system and a major part of the equation that determines its overall security. Therefore, it is important for students to learn about auditing an operating system and securing it if needed. Exercises in this category include:

Identifying installation vulnerabilities: In this set of exercises students learn about various auditing tools to check various operating systems, such as Linux, Windows, Solaris, and BSD, and are asked to evaluate and fix the problems identified by the auditing tools.

This class of assignments would only require a setup similar to the setup used in the first class. Except a different operating system have to be installed for each assignment. Again, this can be done in *VITAL*.

6 War gaming:

Finally, one class of assignments this is found most interesting, for both students and instructors, is one in which students compete with each other to capture/compromise each others information system. A network setup similar to the one depicted in figure /ref/vlnet3 would be sufficient for this exercise. Again, in *VITAL* this can be done easily.

IV. CONCLUSION

In this paper we have presented the design of *VITAL* - a virtual laboratory that allows multiple institutions to share a single physical laboratory. We have identified the key characteristics that a well designed virtual laboratory must demonstrate in addition to the desirable characteristics of an IA laboratory. These characteristics include remote configurability, thereby providing instructors using the lab to individually provision and configure desired networks without interfering with the labs set up by other instructors. We the presented a brief description of a particular design that is able to achieve the stated goals and characteristics at a reasonable cost. Finally we gave a list of exercises that can be run on the proposed design. Currently the laboratory is being built under an NSF capacity building grant and a more detailed report on our experience with the lab will be presented in the near future.

REFERENCES

- [1] V. Ananthapadmanabhan, P. Frankl, N. Memon and G. Naumovich. Design of a Laboratory for Information Security Education, *Proceedings of World Conference on Information Security Education*, pp 61-73, Monterey, CA, July 2003.
- [2] M. Bishop. What Do We Mean by "Computer Security Education"? *22nd National Information Systems Security Conference*, Oct. 1999.

- [3] Cisco Professor Bootcamp. http://www.cisco.com/security_services/ciag/initiatives/training/bootcamp.html
- [4] ISSL: Information Systems Security Laboratory, Iowa State University: <http://www.issl.org/>
- [5] John M. D. Hill et. al. Using an Isolated Network Laboratory to Teach Advanced Networks and Security. *Proceedings of ACM SIGCSE Technical Symposium on Computer Science Education*, Charlotte, North Carolina, pp 36-40, Feb. 2001.
- [6] Cynthia E. Irvine. Amplifying Security Education in the Laboratory. *Proceedings IFIP TC11 WC 11.8 First World Conference on Information Security Education*, pp 139-146, Kista, Sweden, June 1999.
- [7] Prabhaker Mateti. A Laboratory-BAsed Course on Internet Security. *Proceedings of ACM SIGCSE Technical Symposium on Computer Science Education*, Reno, Nevada, Feb. 2003.
- [8] National Colloquium for Information Systems Security Education. <http://www.ncisse.org/>.
- [9] National Coordination Office for HPCC. Committee on Information and Communications (CIC) Strategic Implementation Plan. http://www.whitehouse.gov/WH/EOP/OSTP/NSTC/html/cic/cic_plan.html.
- [10] National Information Assurance Training and Education Center, University of Idaho, (Director Corey Schou) <http://cob.isu.edu/schou/niatec.htm>
- [11] NSA Centers Of Academic Excellence in Information Assurance Education <http://www.nsa.gov:8080/isso/programs/coeiae/index.htm>.
- [12] V. Padman, N. Memon, P. Frankl and G. Naumovich. Design and Implementation of an Information Security Laboratory. *Journal of Information Warfare*, Volume 2, Issue 3, 2003.