# DESIGN OF A DISTRIBUTED COMPUTER SECURITY LAB

T. Andrew Yang, Kwok-Bun Yue, Morris Liaw, George Collins,
Jayaraman T. Venkatraman, Swati Achar, Karthik Sadasivam
University of Houston-Clear Lake
2700 Bay Area Boulevard, Houston, TX 77058, USA
yang@cl.uh.edu

Ping Chen
University of Houston-Downtown
One Main Street, Houston, TX 77002, USA

## ABSTRACT

Across the US and the rest of the world, there exists a lack of computer security components in many CS/IT curricula. For those programs that do have such components in computer security, a common difficulty is to integrate "real-world" labs into the courses, in order to provide hands-on experiences to the learners. Due to concerns for security breaches and network hacking, system administrators are reluctant to allow computer security labs involving network sniffing, virus scripting, etc. to be deployed in the campus network. Without hands-on, real-world projects, it is difficult for the learners to integrate the acquired security theories and knowledge with up-to-date security technologies and practices. Computer science educators who are interested in teaching computer security in a "realistic" context are thus faced with a unique challenge: Setting up 'real-world' computer security laboratories and assignments, without negatively impacting the rest of the campus network. The primary goal of our project is to develop a Distributed Computer Security Lab (DCSL) to answer the challenge. We have established, across multiple university campuses, a computer lab which enables the faculty and students to analyze and study vulnerabilities of a realistic corporate network. The lab provides hands-on experience for students to study cutting-edge computer security technologies, and serves as a test bed for projects which are otherwise impossible to implement in general-purpose labs. In this paper, we first discuss the general model of the DCSL and our implementation, and then present a selected set of projects that we have conducted to aid the design of the DCSL. The paper concludes with a summary and future work.

## 1. INTRODUCTION

For the past decade, partly due to the widespread use of the Internet, computer security has become a top issue in industry, academia and government. The demand for well-trained security professionals has grown dramatically. The integration of security into computing curricula, however, has not kept up with this demand [7]. There is a large discontinuity between the demand for security professionals and the academic programs that produce them. This deficiency deepens in undergraduate programs, where few have security courses. A related problem is, despite the ubiquitous nature of security, most existing computing courses lack security components. The problems are even more serious for smaller universities where

resources tend to be limited. Our study has indicated that the overwhelming majority of existing information security programs are at the graduate level. Across the US and the rest of the world, there exists a lack of computer security components in many CS/IT curricula.

For those programs that do have courses in computer security, a common difficulty is to integrate "real-world" labs into the courses, in order to provide hands-on experiences to the learners [1]. Due to concerns for security breaches and network hacking, system administrators are reluctant to allow computer security labs to be deployed in the campus network. Unless deployed in a isolated computer lab, projects involving *hacking* techniques, such as network sniffing and virus scripting, are generally prohibited in the campus network. Without hands-on, real-world projects, it is difficult for the learners to integrate the theories and knowledge acquired in the classroom with up-to-date security technologies and practices.

Computer science educators who are interested in teaching computer security in a "realistic" context are thus faced with a unique challenge: *Setting up 'real-world' computer security laboratories and assignments without negatively impacting the rest of the campus network.* To mitigate the above-mentioned problems, we have worked on two related projects. The first project focuses on designing a distributed computer security lab across multiple sites, to simulate how a real-world corporate network would be configured. The distributed lab can be used as a test bed for projects related to security in distributed systems, such as those related to network and Internet security. One of the benefits provided by such a *distributed* lab model is its potential to enable a computer security lab to share its computing and networking resources with a smaller university or college. Such *remote sharing* capability is desirable especially for smaller colleges where resources tend to be limited. The second project is related to building a module-based computer security curriculum model that would enable easy and flexible adoption of courseware modules and sub-modules by smaller universities. The focus of this paper is on the design of a distributed computer security lab. Results of the second project will be reported in another paper.

The rest of this paper starts with a discussion of the challenges facing the design of a computer security lab. A survey of published computer security lab designs reveal how each of the challenges would be addressed by various designers. We then present our distributed computer security lab (DCSL) model, and how we have addressed the challenges discussed earlier. We then discuss a selected set of lab experiments that we have developed to aid the design of DCSL. The paper concludes with a summary and discussions of future work.

## 2. CHALLENGES IN DESIGNING COMPUTER SECURITY LABS

The use of specialized computer security labs for teaching computer security related courses has long been advocated by CS educators. Hill etc., for example, found the use of an isolated network laboratory for active learning to be more effective for teaching network security and preferred over a lecture-based course [2]. Schafer etc. describe an isolated laboratory used by undergraduate students and faculty researchers, which has become a vital part of their curriculum. It also describes the process used to create the lab using limited resources [5].

There exist many challenges in setting up realistic computer security laboratories and assignments. Some of the challenges that we have identified are listed below:

*a) Need to protect campus networks*

Due to the widespread virus attacks and hacking incidents, the system administrator is justifiably concerned with the health of the networks that he/she is responsible for. To avoid security breaches caused by security and hacking programs running in the computer security lab, most universities isolate their computer security labs from the rest of the campus network. The ISIS lab, for example, uses one-way firewall filtering to assure that traffic from the computer security test bed would not spill over to the rest of the campus network [3]. An isolated security lab, however, is contradicting to the second challenge: need to access the Internet.

*b) Need to access the Internet*

Students and faculty who use the computer security lab do have the need to connect to the Internet to, for example, get information, download security software, etc. Schafer, etc. relies on a dedicated workstation that is connected to the campus backbone for this purpose [5]. It is inconvenient for a whole class of students to share one or two workstations in order to access the Internet. Padman, etc. employs a set of workstations that sit between the campus network and the security lab test bed [3]. A user uses one of the workstations in the set to access the campus network and the Internet. The nodes in the test bed can only be accessed when a user requests a connection via one of the workstations in the set. Accessing the test bed nodes from other computers is prohibited.

*c) Difficulty to simulate enterprise/departmental level network environment*

In the real world, an enterprise or departmental network consists of a multitude of hardware and software devices, such as routers, switches, hubs, firewalls, email servers (SMTP), SNMP servers, etc. To ensure secure networking, other security appliances, such as VPN (virtual private network) servers and clients, Radius servers, Kerberos servers, etc., need to be added into the network. With increasing popularity of wireless networking, wireless devices, such as access points, wireless NICs, etc., are becoming part of an enterprise network environment. The task of installing, configuring, deploying and maintaining such a complex network has proved to be a major challenge, especially for individual instructors who are interested in teaching computer security in a realistic enterprise network environment.

Some CS educators/researchers have worked on creating specialized labs for teaching computer security courses. Tikekar and Bacon [6], for example, discuss the development of three levels of lab exercises. The beginning level includes exercises that mirror an actual enterprise and allow machines to be "attacked" while protecting the campus and external networks. The second level is designing assignments that model real-world situations like finding vulnerabilities in a system and using them to gain access to the system. The third level is designing larger exercises or projects that can be undertaken at the capstone or graduate level. Rawles and Baker [4] presents an implementation plan for deploying a public key infrastructure for use in teaching how to build a secure electronic mail system that addresses three security areas: authentication, authorization, and nonrepudiation.

*d) Difficulty in allocating various resources for different assignments*

It is anticipated that the computer security lab should support more than one project at a time. Different projects, however, would impose different requirements in terms of network configurations and needed resources (e.g., routers, web servers, honeypots, hardened computers, intended victim servers, etc.).

Padman, etc. [3] discuss their design of a configurable and secure system experimentation test bed, by using integrated technologies such as swappable hard drives and virtual LAN (VLAN). The test bed is insulated from the rest of the campus network, in the sense that activities in the lab do not affect traffic on the campus network. The test bed is reconfigurable and can support different projects' needs by having different sets of swappable hard disks for different projects, and by configuring the VLANs to simulate various network configurations.

*e) Resource needs for students to develop their solutions*

Students may not always have their own computers, and their own computers may not mimic the environment of the proposed assignments. The computer security lab needs to provide needed resources for students to develop their assignment solutions. Furthermore, to provide consistency in assignment evaluations, it is desirable for students to work in identical environments, even when they can set up similar environments in their own computers.

*f) Easy and secure access to the resources*

The resources available in the lab should be easily accessible. Students may choose to use the lab either locally or remotely (e.g., from home or from their work place). Remote access to the lab, however, tends to prompt security concerns, mainly due to the widespread hacking incidents launched over the Internet.

*g) Incorporation of latest technological development*

Computer technologies are notorious for their fast-paced change. New technologies are constantly created. To accommodate the latest technological development, such as wireless networking, secure remote access, etc., it is important that the design of the security lab be *scalable*, in the sense that additional test beds or components may be easily added to the existing network. A security lab has to take *extensibility* and *reconfigurability* into account to accommodate new technologies. Maintaining checklists or procedure lists of configuration and equipment requirements for each project or assignment, for instance, may make it feasible for a lab to be reconfigurable and to support multiple projects more easily.

*h) Overhead of configuring and maintaining the test bed for different assignments*

The procedure of configuring and maintaining the test bed for different lab assignments can prove to be a continuous and tedious process, and will require intensive resources. This challenge involves human resource management and cannot be completely solved by a well-designed lab. When a dedicated lab administrator is not available, RAs or TAs may be employed to assist the instructors in setting up lab projects and assignments. Once a disk unit is properly configured for a given project,

batch copying devices can be used to enable fast copying of large quantity of disk units for a given project [3].

## 3. DESIGN OF THE DISTRIBUTED COMPUTER SECURITY LAB

The high-level configuration of the cross-campus Distributed Computer Security Lab (DCSL) is depicted in Figure 1. The distributed lab currently consists of two sites communicating over the Internet via DSL connections, although each site is insulated from its respective campus network. In a sense, the two labs are *external* to its respective campus network.

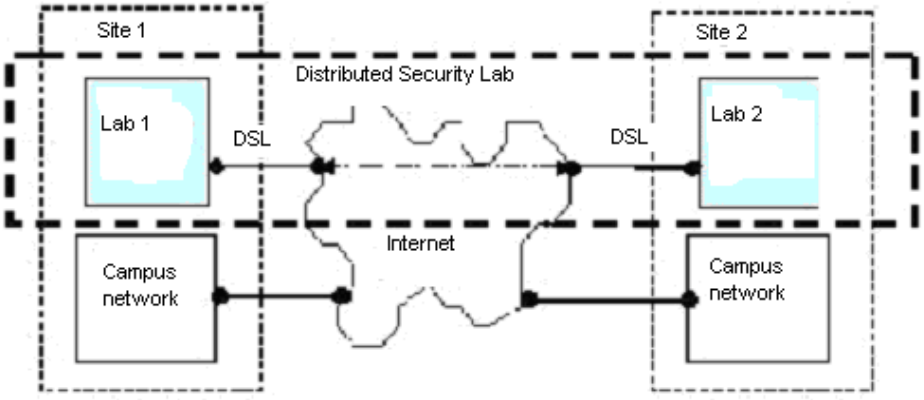


**Figure 1: A Model of Cross-campus Distributed Computer Security Lab (DCSL)**

The model is *expandable* in the sense that more sites may be added to the distributed lab. Resources across the various sites may be shared when secure remote access mechanisms (such as VPN) are implemented.

### 3.1. The Design Goals

In response to the challenges identified in the previous section, we have the following goals in mind when designing the DCSL.

1) An *insulated but connected* lab: In response to challenges *a (need to protect campus networks*), *b (need to access the Internet*), and *c (difficulty to simulate enterprise/departmental level network environment*), the DCSL has Internet connectivity via DSL connections without going through the campus backbone. Although each site is completely separated from the respective campus network, the DSL connections allow the distributed sites to communicate with each other. This separation facilitates enterprise-level distributed experiments without the danger of intruding the campus networks.

2) An *easily configurable* lab for various experimentations (in response to challenge *d, difficulty in allocating various resources for different assignments*): Learning from the ISIS lab model [3], the DCSL consists of a dedicated test bed of computers, which are equipped with swappable hard drives and are connected via switches that support VLAN. The test bed can be easily reconfigured to satisfy the requirements of different projects. The test bed can be configured to simulate a real-world environment, such as a virtual corporate network with a set of virtual LANs

connected via routers. Assignments that involve virus attacks, for example, can be deployed in such a virtual network.

3) A dedicated computer security lab with VPN support for remote access (in response to challenges *e, resource needs for students to develop their solutions*, and *f, easy and secure access to the resources*): In the DCSL lab, a test bed of workstations is provided for students' use. With swappable hard disk units in each of the workstations, different projects can use different set of swappable disk units, thus allowing multiple projects to be conducted simultaneously in a given semester. In addition, the DCSL supports VPN (*Virtual Private Networks*), which is the most commonly adopted security technology by corporations to assure secure remote access to the corporate networks and back-end servers. A student working at home may use VPN to open a secure channel between his home computer and the DCSL test bed. For those who like to work in the campus, DCSL is configured as a dedicated laboratory, and thus is separated from the general laboratories.

4) A *sharable and secure* lab: In addition to be used by the students and faculty locally, an interesting feature of the DCSL design is its potential to support *remote sharing* of the resources in the lab. Currently the DCSL comprises two local labs respectively in two campuses. The DSL connectivity allows the two labs to be remotely connected to form a distributed platform. With site-to-site VPN implemented, users at a site can securely access resources available at the other site, or run security projects across the distributed platform. More local sites may be added in the future into the DCSL.

5) Incorporation of *emerging technologies*: Wireless networks are part of the DCSL. Wireless local area networks (WLAN) and mobile networks are needed to study wireless security. The wireless LANs in DCSL are compliant to the newer IEEE 802.11g standard, which is backward compatible with 802.11a and 802.11b. It is well known that the WEP (*Wired Equivalent Privacy*) protocol, which comes as the default security feature of 802.11 protocols, is not sufficient to enable secure wireless communication. Thus, we are in the process of adding other security features, such as LEAP, VPN, SSL, and the forthcoming *802.11i* to address the security issues in 802.11 protocols. For detailed discussions of security issues in 802.11 protocols, please refer to [8].

In addition, the lab incorporates other enterprise-level security technologies such as VPN, which not only supports secure remote access, but also enables projects experimenting with secure remote access to the back-end servers.

### 3.2. Identification of Vulnerability Points in the DCSL Network

In order to design a distributed computer security lab for testing system vulnerabilities and control measures, we started by identifying the *vulnerability points* that may exist in a typical enterprise network. Figure 2 depicts a high-level configuration of a site of the DCSL for network security experiments. Different sites in the DCSL may have different set of equipments and network configurations, so Figure 2 represents a typical design. A typical site in the DCSL may consist of four test beds: (a) a local area network (LAN) to simulate a corporate or campus network with integrated firewall, VPN server, and authentication servers; (b) a wireless LAN, which is composed of several access points and wireless clients; (c) a second LAN to

simulate a remote site; and (d) remote connections through the Internet, which simulate a home office, a small office, or access over a wide-area mobile network.

Seven *vulnerability points* (VP A through G) have been identified and are marked in the figure. Each of the VPs represents a potential point of attack where an attacker may exploit various vulnerabilities in the system.

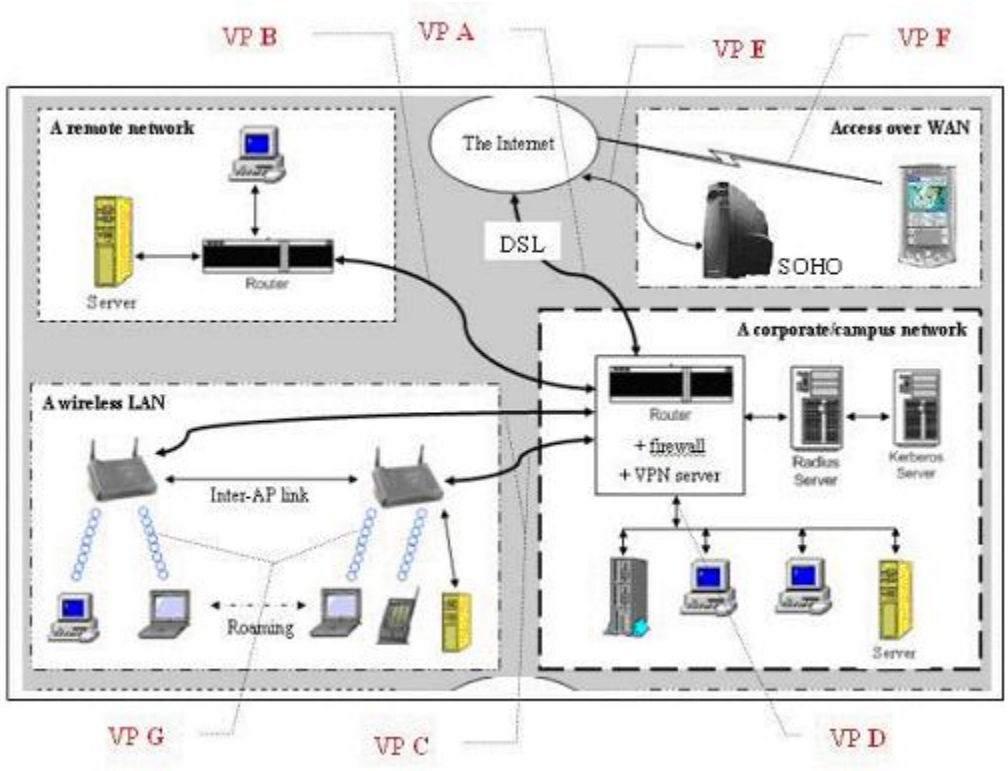


**Figure 2: Networking Security Testing Environment**

- VP-A represents attacks coming from a public network, such as the Internet. Employees may use a dialup line, a subscribed ISP service, or a mobile device/service to access the corporate network from the public network.
- VP-B represents attacks launched from a remote network, which may be one of the departmental networks of the same corporation.
- VP-C represents attacks from a wireless LAN. Special authentications such as LEAP and RADIUS are required to guarantee the wireless LAN to be safe.
- VP-D represents attacks that may be launched from within the corporate network. This type of attacks include deliberate attacks by employees as well as involuntary attacks which may be launched by, for example, an ignorant employee opening an email with a subject line that says “I love u”.
- VP-E represents attacks targeted at small offices or home offices (SOHO), where an employee uses a dial-up line or cable modem to connect to the corporate network.
- VP-F is the type of attacks taking advantage of a mobile network, in which data are transmitted through the air. Mobile networking plays a significant role in the realization of pervasive computing, which would allow users to have access to

network resource from anywhere at any time using small devices such as a cellular phone or a PDA. The growing use of mobile services, however, implies increasing attacks associated with mobile networks.

- VP-G is similar to F in the sense that data in a wireless local area network (WLAN) also travel through the air. Communications occur between a wireless client and an access point, which serves as a bridge between the client and the backend network. Recent news stories have revealed the vulnerability of the IEEE 802.11 protocol, which is the network protocol built into WLAN cards. The encryption method used in the 802.11b protocol is the WEP (Wired Equivalency Protocol), which is vulnerable to attacks, unless other methods (such as LEAP and RADIUS) are integrated to achieve a secure solution.

Some of the VPs (A, B, C, D) correspond to direct attacks at the corporate network. The others are indirect attacks at small offices or home offices (E), at wireless mobile networks (F), or at wireless LANs (G). An indirect attack may eventually become a threat to the corporate network. An attacker, for example, may take over a telecommuter's identification by attacking his/her home office and then use the exposed user information to access the corporate network. A drive-by hacker, as another example, may connect to an insecure wireless access point and then gain access to the corporate network via the access point.

### 3.3. The *Prototyping* Approach in Designing a DCSL Site

A unique feature of the DCSL is its independent Internet connection, which enables the lab to be insulated from the campus backbone, while remaining connected to the Internet. The prototype network at UHCL and UHD each has a dedicated DSL connection to the Internet, allowing them to form a distributed computer security test bed, while remaining insulated from the respective campus network.

When designing the DCSL, an early decision was made that, before making major purchase of the instrumentations, we would investigate the hardware and software components that we would like to include when deploying the DCSL. We have adopted a *rapid prototyping* approach when setting up the DCSL. Before deploying the target network using expensive devices, a prototype network was first deployed with a myriad of old servers and workstations, on which freeware firewalls, VPN servers, etc. were installed. Incrementally, a series of simple projects were developed to test the network and its various components. Exploratory projects that we have conducted include the following:

**a) Active Directory Installation:** Active Directory was configured on a Windows 2003 Server (Galileo in Figure 3). The machine was also configured as a DNS. The machine was made the first domain controller and a new forest was established. The active directory configuration was verified from a client (Alpha) in the internal network.

**b) Configuration of the Linux Router:** The simple network in Figure 3 was expanded by adding a machine running Linux Red Hat 9.0 (Einstein in Figure 4). The machine is dual homed and IP forwarding was enabled between the two interfaces. The interfaces were configured to two different Class C networks (192.168.100.0 and 192.168.200.0).

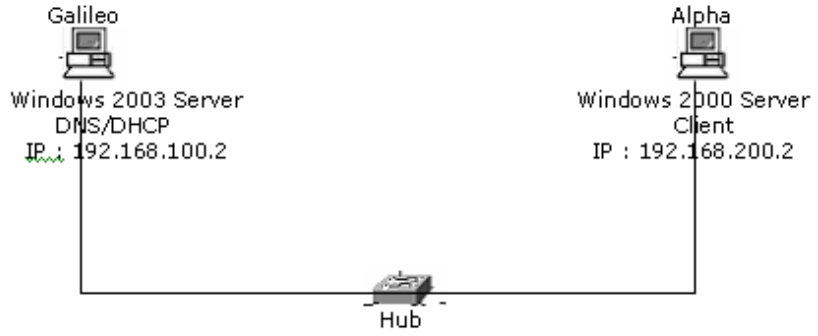


**Figure 3: Configuration for Active Directory Experiment**

c) **Configuration of VPN Server in Galileo:** Using the network shown in Figure 4, a VPN tunnel was established between a test user in Alpha (as a VPN client) and Galileo (the VPN server), and dial-in access was provided. This was tested by mapping a drive to the home directory upon login.

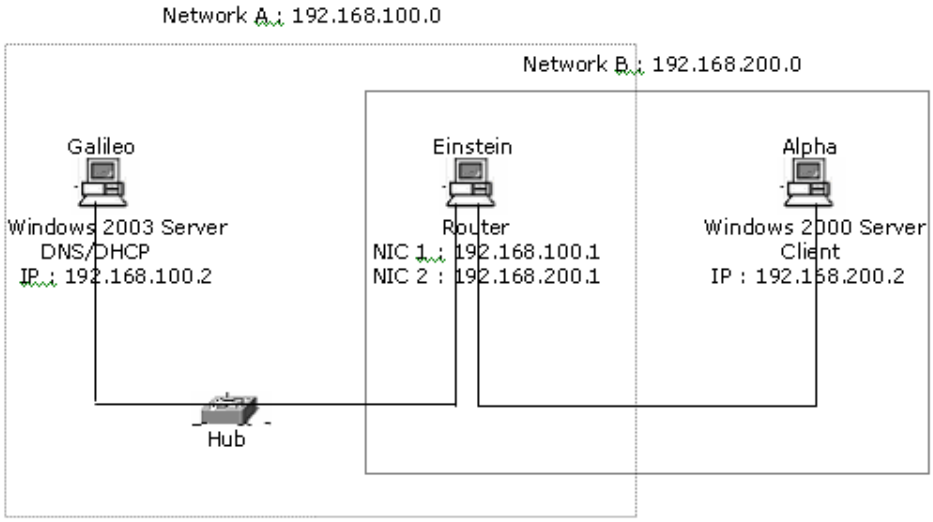


**Figure 4: Configuration of a Linux Router**

d) **Configuration of IAS (RADIUS) Server in Galileo:** A MS IAS server was installed to communicate with the Active Directory for authentication information. The VPN server on Galileo was made the AAA client and was configured to obtain the authentication and authorization information from the AAA server.

e) **Configuration of VPN Client in LinAlpha:** The VPN client was setup on Alpha and a tunnel was established with the Microsoft VPN server running on Galileo. This was tested by pinging Galileo through the VPN tunnel.

f) **Configuration of Linux VPN Client in LinAlpha:** To test connectivity between the MS VPN server and a Linux VPN client, Linux 9.0 was installed on Alpha, which became a dual boot (as shown in Figure 5). OpenSource's PPTP client for VPN was then installed. The PPTP client was configured to talk to the Microsoft VPN server running on Galileo. A VPN tunnel was established from the Linux on Alpha to the Microsoft VPN server on Galileo. The tunnel was tested by pinging Galileo through the VPN tunnel.

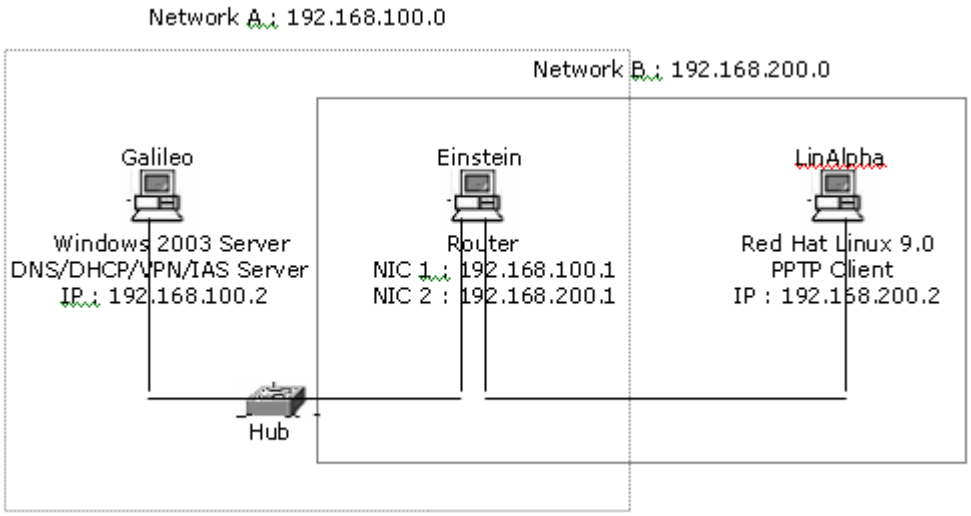


**Figure 5: Configuration of a Linux VPN Client**

**g) Configuration of a Wireless Local Area Network:** As depicted in Figure 6, a Cisco access point was installed in the network and an IIS server was installed on a wireless desktop. HTML pages on the IIS server were accessed by a wireless desktop and PDA clients.

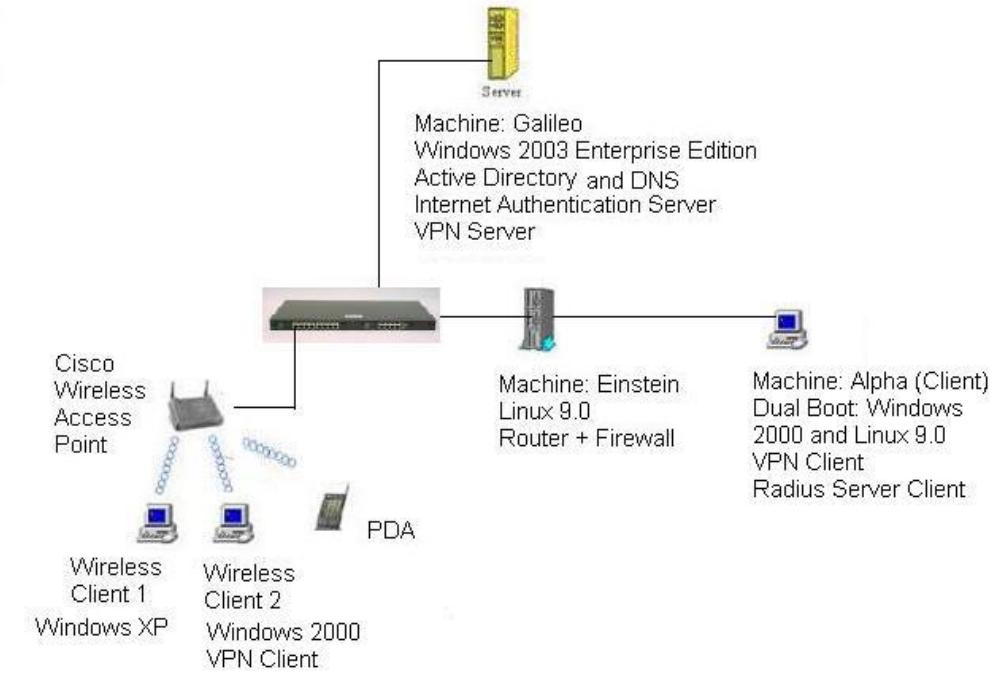


**Figure 6: Addition of a Wireless LAN to the Network**

With the prototyping projects, we have gained much insight about the detailed design of a single-site DCSL network. Details of the network prototypes and relevant experiments can be found at the project web site[1].

[1] http://dcm.cl.uh.edu/nsfsecurity/public/experiments.html

Figure 7 represents the design of the prototype network for a single DCSL site, resulting from the series of incremental experiments as discussed above.

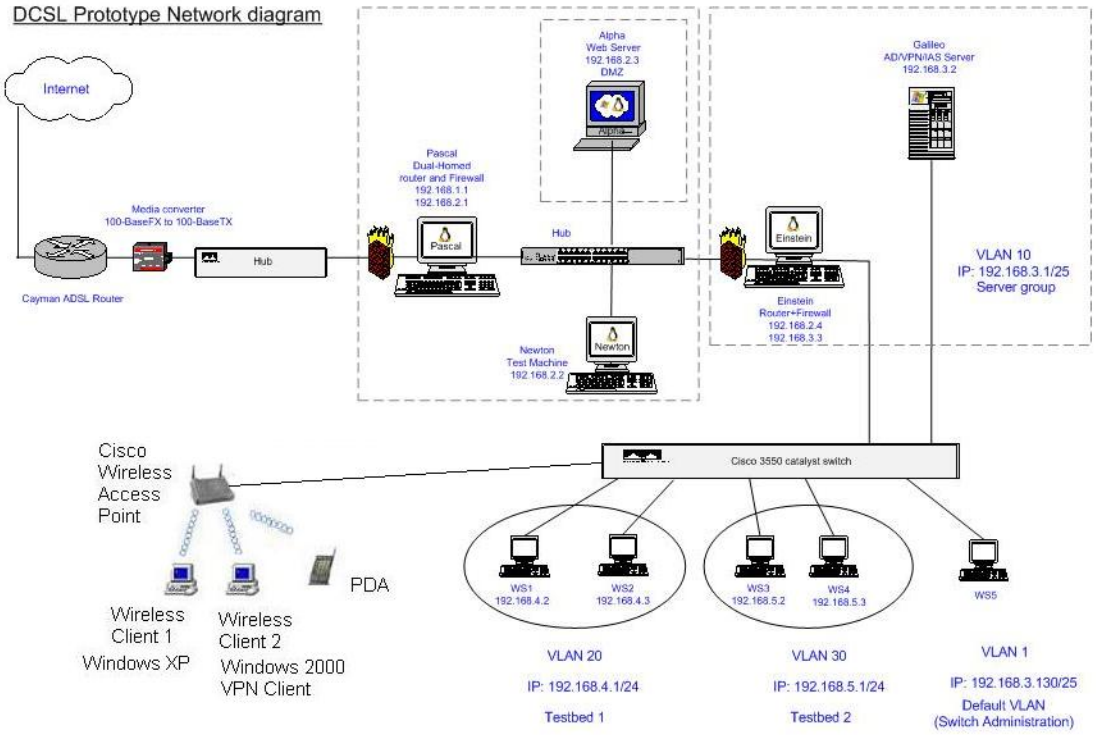


**Figure 7: Prototype Network of a DCSL Site**

To meet the design goals as outlined in section 3.1, the *target network* is composed of the following devices and/or configurations:

a. A DSL router, which connects the lab to the Internet via the DSL connection;
b. A dual-homed software router/firewall (Pascal), acting as the gatekeeper between the DCSL network and the outside world;
c. A regular switch connecting the DMZ (*Demilitarized Zone*) to the server cluster;
d. A DMZ that contains publicly accessible servers (such as a Web server) and testing workstations (e.g., for monitoring network traffic, etc.);
e. A 2nd router/firewall (Einstein) separating the DMZ from the server cluster;
f. A hub or switch connecting the 2nd router/firewall with the back-end servers;
g. A set of network security servers, including firewalls, VPN server, IAS (Microsoft's Internet Authentication Server), and Radius server;
h. A test bed of computers, which are equipped with swappable disk units and are connected via a VLAN switch and routers. The design is still evolving and we are in the process of deploying real-world lab assignments to test the design, including its usability, extensibility, reconfigurability, and scalability.

### 3.4. The Current Design of a DCSL Site

After the prototype network was deployed and evaluated, we started deploying the *production* network, which will eventually be used as the main security lab for supporting classroom projects. We decided to retain the *prototype* network for the following reasons: a) The prototype network can be used by instructors and their

research/teaching assistants for testing exploratory types of projects before installing them in the production network; b) The prototype network may be used by faculty members and their students to conduct research experiments, especially those requiring an insulated lab environment. Figure 8 illustrates how the two networks co-exist and share a DSL connection to the Internet.

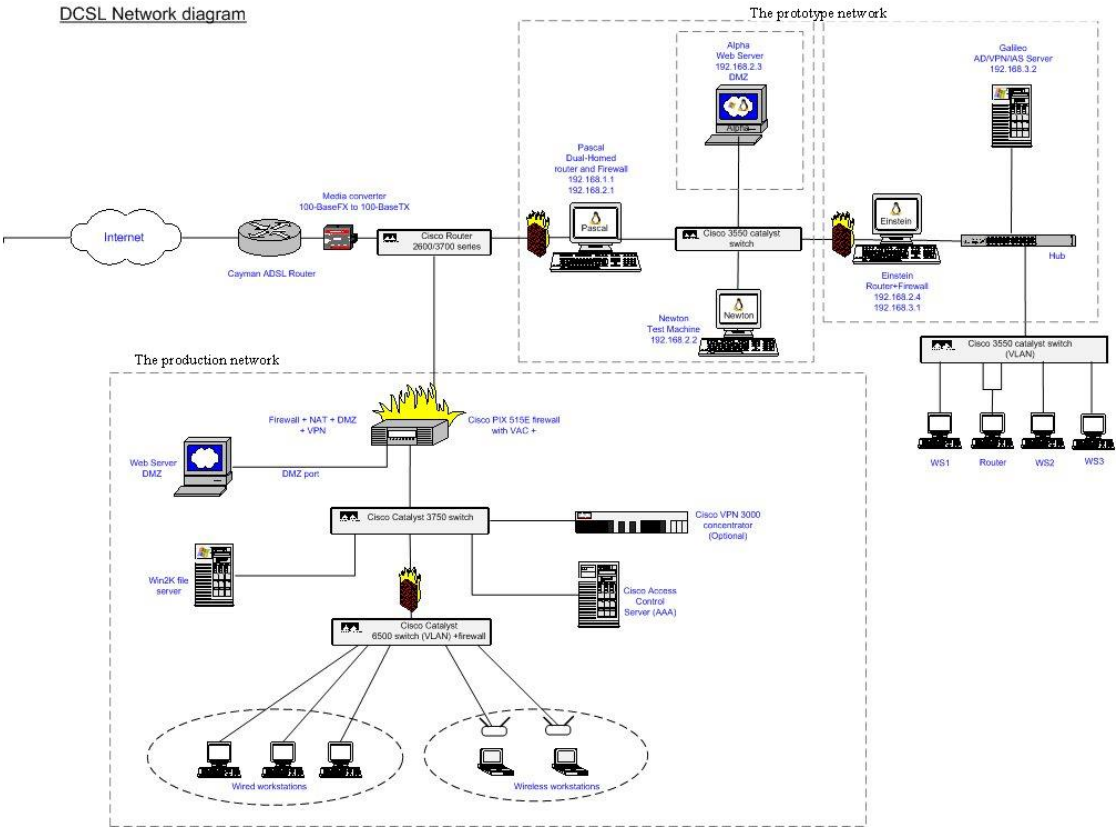


**Figure 8: The Current Network Design**
**- showing the co-existing prototype and production networks**

As of the time of writing (July 2004), the production network is being deployed. We are also in the process of designing and testing other type of projects, including labs that may be assigned by an instructor to students, who then implement the projects in the DCSL. Due to limitation of space, the student projects that we have developed will be discussed in a separate paper.

## 4. SUMMARY AND FUTURE WORK

In this paper, we present the need of a realistic distributed computer security lab, its design challenge, and our responses to those challenges. By employing separate Internet connectivity for the lab, we have mitigated the common concerns for the lab's possible negative impact on the integrity of the campus network. By adopting a *prototyping* approach in developing the lab, we had conducted a sequence of exploratory networking experiments, and have learned how to deploy and maintain an enterprise network for teaching computer security topics. For each of the experiments, the detailed steps and needed resources are documented and placed on our web site for interested instructors to access. By combining DSL and VPN, more remote sites may

be integrated into the existing DCSL network, allowing smaller universities to deploy larger-scale computer security projects.

The design of the DCSL is an on-going project. Each of the DCSL sites is being "enhanced" by industry-grade routers and firewalls, switches supporting larger number of ports, and several wireless access points to support wireless LANs. In addition, each site will be connected to a local classroom lab. Each of the desktops in the classroom lab will be equipped with a WLAN adapter to enable communications with the access points. In addition, a set of laptops will be added. Each of the laptop is to be equipped with a mobile WWAN (wireless wide area network) adapter, which will allow the laptop to be used in studying mobile data communications, such as *GSM* (Global System for Mobile Communication), *GPRS* (General Packet Radio Service), *CMDA* (Code Division Multiple Access), and mobile protocols, such as *WAP* (Wireless Application Protocols), *WTLS* (Wireless Transport Layer Security), etc.

## ACKNOWLEDGEMENT

This work is partially supported by the National Science Foundation (Grant DUE-0311592).

## REFERENCES

[1] J. Herath, A. Herath. Integration of computer security laboratories into computer architecture courses to enhance undergraduate curriculum. *Proceedings of the 30th International Symposium on Computer Architecture*. San Diego, CA, 2003.

[2] J. M. D. Hill, C. A. Carver, Jr., J. W. Humphries, U. W. Pooch. Using an isolated network laboratory to teach advanced networks and security**.** *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science*. 2001.

[3] V. Padman, N. Memon, P. Frankl, and G. Naumovich. Design and Implementation of an Information Security Laboratory. *Proceedings of World Conference on Information Security Education.* 2003.

[4] P. T. Rawles and K. A. Baker. Developing a public key infrastructure for use in a teaching laboratory. *Proceeding of the 4th conference on information technology curriculum on Information technology education*. 2003.

[5] J. Schafer, D. J. Ragsdale, J. R. Surdu, and C. A. Carver. The IWAR range: a laboratory for undergraduate information assurance education**.** *Proceedings of the sixth annual CCSC northeastern conference*. 2001.

[6] R. Tikekar and T. Bacon. The challenges of designing lab exercises for a curriculum in computer security**.** *The Journal of Computing in Small Colleges, Volume 18, Issue 5* (May 2003) Pages: 175 – 183

[7] T. A. Yang. Computer security and impact on computer science education. *The Journal of Computing in Small Colleges, Volume 16 Issue 4* (May 2001).

[8] Yasir Zahur and T. Andrew Yang. "Wireless LAN Security and Laboratory Designs". The *Journal of Computing Sciences in Colleges. Volume 19, Issue 3*. January 2004.