MOBILE – A MOBILE Instructional Laboratory Environment for Hands-On Computer Science Education

Project Description

Background and Need

Computer science educators need better and more flexible “hands-on” educational training environments for teaching as well as training other computer science educators. The field of computer science involves not only a large set of base principles, but increasingly makes use of a rapidly evolving array of languages, frameworks, tools and environments. Providing hands-on laboratory environments for effective and efficient faculty training has thus become a challenging task [3].

As background and as an example, consider the interest in faculty/educator training at the national Association for Computing Machinery (ACM) Special Interest Group in Computer Science Education (SIGCSE) conference. Each year, approximately thirty workshops are offered to hundreds of educators. Until 2006, the approximately ten “hands-on” workshops were offered in a computer lab at a local university, though several problems with this model of workshop delivery became apparent over a number of years. First, the host university has a limited number of suitably equipped laboratories. Second, access to the laboratories is controlled by the host site, not by the workshop providers. This separation of control has led to significant delays in access (e.g. SIGCSE 2004 workshop #8; SIGCSE 2005 workshop #11). Third, the host site often does not want their computer configurations changed (e.g. SIGCSE 2004, SIGCSE 2005). This limits the ability of workshop providers to install their software in the host environment. Fourth, the workshop software cannot easily be pre-tested in the host environment. This often leads to software that functions partially or not at all, leading to delayed or cancelled hands-on activities during the workshops. For a specific example, at a SIGCSE 2005 workshop, the presenter had planned on using participant laptops with a CD of material and software that he had created. While he had tested this on desktop systems, the participant laptops used had different CD drives/drivers and would not read the CDs, leading to the failure of the presenter’s planned hands-on activities. In fact, we have seen these types of problems happen at more than half of the hands-on workshops we have attended at SIGCSE over the past five years.

Our Prior Work

In the past three years, we have presented two workshops each year on computer security issues at SIGCSE, and decided to avoid the above problems by bringing our own mobile computing laboratory. Using funding from the National Science Foundation CCLI grant 0309818 [1] and matching funding from
a local donor at our university, we purchased twenty laptop computers and a network switch, and brought together individuals from our computer science department faculty, systems staff, and campus networking support group to develop not only the computer security workshops, but also a prototypical set of network and system tools that could configure the set of laptops as an isolated computer network with its own domain name service, web servers, database servers, file servers, and both Windows and Linux workstations. In less than one hour at SIGCSE 2005, SIGCSE 2006 and SIGCSE 2007, we set up the laptops and network switch and used our prototype software tools to set up each laptop. We connected the laptops to the switch using Ethernet cable, to ensure we had a solid connection free from interference from outside systems and networks. Each pair of workshop participants then worked with one “host” laptop which hosted (using the virtualization software VMWare [12]) both a (virtual) Linux and a (virtual) Windows workstation as client systems. Participants could easily switch between the two virtual workstation environments to use and practice with various computer security tools in each environment. Our workshop team consisted of five individuals: the presenter, a second instructor who monitored the workshop activity, and three systems/network staff. As a result, we experienced no delays, no failures, and had none of the other problems typically associated with hands-on workshops at SIGCSE. And we delivered our workshops without any aid or intervention by the host site.

The workshops were very successful. Our overall workshop evaluation score for the three years was 4.5, 4.6 and 4.6 on a 5.0 point scale. More significantly, when asked if the participants would be interested in using our prototype software configuration tools, the interest level for the two years was 4.5, 4.7 and 4.6 on a 5.0 scale. (Two support letters from past workshop participants are included in our Supplemental Materials section, Appendices 1 and 2.) The full evaluation results are publicly available on our CLICS project website [2] in the Workshop Materials section. We also had our workshops externally reviewed by Dr. Sujeet Shenoi and David Greer, information assurance and security specialists at the University of Tulsa, and the full text of their review is also publicly available through our project web site [2]. Finally, our approach was discussed by the SIGCSE program committee for wider application to future conference workshops, and onsite laptop workshops have been the preferred format for SIGCSE workshops since 2006 (see support letter, Appendix 3). This illustrates both the need for tools such as the ones we developed, and also confirms to us that our approach is of value to others.

Of course, we understand that most instructors need to present workshops on their own, not with three to five staff members in support. There is also a need to combine such software tools into a single system with an easily used interface. We, too, have those needs; we must reduce our dependence on large numbers of support staff, and we must create a simple user interface to the software prototype. We also would like to develop a system that could be used on any group of laptop computer systems, as laptops are now very common and most computer science faculty own or have access to them. We have
investigated currently available tools for such purposes, and while networked system imaging tools (e.g. Symantec’s Norton Ghost [4]) and even remote laboratory configuration and management tools (e.g. RemotelyAnywhere [9]) exist, and we are aware of individual instructors and others developing tools that do part of what we propose, we have found no generally-available system that specifically supports rapid and dynamic configuration of educational workshop and laboratory environments that can be used in the manner that we demand as computer science educators.

Goals and Objectives

This NSF CCLI phase 2 proposed project builds on the prototype work we have already completed for our computer security workshops. We propose to combine our existing tools into a software system which will support a mobile laboratory for delivering educational workshops at the university level. This proposal focuses on demonstration of this system in the computer security subfield, as that is our area of expertise. However, we envision our mobile network system as being useful for workshops and training in a variety of topical areas in computer science, not simply computer security. Part of our proposal is to train other instructors to develop workshop or course exercises in the computer science area of their choice, in order to support a wide array of usage within the overall field of computer science.

This project involves the further development and significant improvement of our prototypical portable network system in several areas. First, we propose to add wireless support to our system (we currently cable the laptops together through a wired switch) to reduce the initial workshop setup time. This will have to be done in a way that not only establishes a stable connection for the network, but avoids interference from other wireless activity in the area. Second, we propose to develop this system so that it can be installed and supported on a heterogeneous suite of computer systems (e.g. a set of laptop computers provided by the workshop participants) from a portable storage device such as a USB drive or iPod. In short, participants and/or workshop hosts provide a set of computer systems, presenters provide the workshop content, and we provide the software that will deliver that content on those computer systems in an easy to configure and reconfigure networked model. Third, we propose to integrate the various pieces of our current system into one unified system, so that university instructors can develop and present workshops such as ours on their own.

We envision a software system that will support the following general workshop scenario: a workshop presenter will be able to present traditional lecture material (e.g. using PowerPoint slides) and also control the workshop’s hands-on exercise components through a single integrated system. The files and exercise components to be distributed to the workshop participants for hands-on exercises will be organized in advance by the presenter (or some other domain expert in the area of the presentation) using the configuration and content management components of our system. The workshop participants will have
virtual operating system execution software (such as the free VMWare Player [14]) installed on their computer system prior to the workshop, and they will also be given a self-installing client on a portable storage device such as a USB drive or iPod with which to connect to and register on the workshop network. The presenter will be able to configure the software for the hands-on laboratory at the beginning of the workshop, including confirming the registration of each workshop participant’s system to participate in the workshop group. The presenter also will be able to control the participants’ system software configuration during the hands-on exercises that are part of the workshop. Finally, the presenter will be able to monitor the status of participant systems, as well as show results of the current exercise to all participants. All of this is independent of the specific content domain presented in the workshop.

We have the following three specific goals:

1. We will develop a mobile computer network configuration and management software system involving virtual computer systems that can be used in a variety of university-level educational workshop scenarios. We define “mobile” to mean that our system can be set up and used anywhere with any group of networked computer systems that are capable of running a virtual operating system execution environment like VMWare Player and support wireless networking. The software will allow the rapid configuration of multiple virtual client operating systems and application environments on networked host computers (which could range from user-provided laptop systems to host-institution based desktop systems, as long as they support a minimum configuration which includes the free VMWare Player and certain other system and network characteristics, discussed in more detail below), and will also allow for dynamic modification of the virtual environments including the addition or removal of services, the generation or removal of needed files, and the generation or termination of program activity on the client systems.

2. We will package this functionality with an easy-to-use interface that can be managed on site and in real-time by the presenter. This will allow an instructor or a workshop presenter to control the participant’s environments in conjunction with the workshop presentation.

3. We will illustrate the value of this generalized system to educators by testing it in the computer security domain, starting by re-applying it to a fourth national offering of our updated hands-on computer security workshops at SIGCSE 2009 (subject to acceptance) and follow this with at least two additional national offerings and at least two local or regional offerings of these workshops. We will also host (at the University of Wisconsin – Eau Claire) a workshop on developing content-based portable workshops using our system, so that others can develop their own workshops, and at this point we will not limit the development of content to computer security, but encourage the development of workshops in various areas of computer science. Finally, we will make the system, as well as suitable documentation and basic training materials,
publicly and freely available through the world-wide web so that other instructors can use the results of our work.

The PI, Dr. Paul Wagner, will coordinate the project and direct the system analysis and design. Jason Wudi and Jamison Schmidt will provide system expertise, and Tom Paine and Daren Bauer will provide network expertise. All four will be primary developers of the system; Wudi, Paine and Bauer have been heavily involved in the prototype development and workshop offerings done to date, and Schmidt has local expertise with this system from its usage in a computer security course exercise at our campus. The project will also utilize a modest number of students as secondary project developers (furthering their educational experience). All five faculty/staff participants will be involved in testing the system across multiple domains. Also, another faculty member (Dr. Susan Harrison) with no prior involvement in the project but with expertise in user interface design will assist us in evaluating and improving the system’s user interface (see committal letter, Appendix 4).

It should be noted that while the SIGCSE workshop environment will clearly benefit from this system, it is certainly not the only situation where a portable workshop management software system could be used. In addition to other national and regional conferences (e.g. ITiCSE, regional CCSC and SIGCSE conferences), individual universities could use this system in faculty training and for mobile student laboratory sessions in a classroom environment as well. Having a flexible and dynamic hands-on environment can be a significant benefit for both instructors and students in the field of computer science. In fact, we envision this software to be useful to instructors in any STEM field (e.g. biology, chemistry, physics), and possibly beyond. The development and use of the system is not specific to computer science.

Phase 2 (Expansion) Project Components

Since we have already developed prototypical tools for meeting the above goals, we see this as a phase 2 (Expansion) project, as we have successfully implemented a prototype system but there remains significant work to develop these tools into a unified and easily usable tool for other instructors. In particular, previously we required five "experts" to deliver our computer security workshops using our prototype, and our goal is to reduce this number to one presenter and one domain expert with some computer systems expertise.

This project primarily supports faculty development. The primary benefit of the project is to improve the learning environment and experiences for computer science educators, allowing better integration of presentation material and hands-on laboratory-style exercises during a workshop. We focus here on developing a system that fundamentally changes the way faculty can be educated in a workshop environment – faculty will get hands-on experience using their own laptops but under the guidance and
control of the workshop presenter. We admit that this project does not directly affect a large number of
students, as do many traditional proposals that directly change techniques or content in undergraduate
education. However, we see significant indirect benefits to students, both in terms of faculty being
better trained through the workshop environment and in faculty potentially using this technology for their
own classroom presentations and exercises, allowing students to have a larger number of hands-on
experiences as well as better trained and prepared instructors.

We view this project as involving two of the five project components applicable to CCLI proposals:

1. **it develops faculty expertise**, as it will enable faculty to gain hands-on expertise in new
curriculum areas and in turn will enable them to emulate and develop new teaching techniques
that combine lecture and hands-on practice. Our approach is one of “training the trainers” – i.e.
developing a system under which faculty can be better trained to take the hands-on experience
from laboratory exercises back to their classrooms.

2. **it implements educational innovations**, in that it sets forth a new approach for faculty education
and training development and enables new teaching strategies for high technology environments.
The development of a mobile teaching laboratory is a significant innovation that can change both
how faculty learn and how faculty teach. As the changes in the format of the SIGCSE conference
workshops have shown, hands-on active workshops using computing technology are becoming
the preferred technique for engaging and informing computer science instructors in our rapidly
changing field.

**Detailed Project Plan**

This project builds on the previous work we have accomplished in developing a portable laptop network
that we used in presenting computer security workshops at SIGCSE 2005, 2006 and 2007. To be clear,
though, our goal there was to develop and deliver the computer security workshops, not to create a
prototype workshop software system. The development of that prototype was a side effect, though one
that we discovered was a significant accomplishment in its own right. Hence, our primary goal here is to
further design and fully develop the workshop software system for the delivery of hands-on laboratory
exercises, a system that will be used primarily to train computer science instructors. This will expand the
usage of this hands-on approach to be suitable for wider usage, thus benefiting many more computer
science instructors.

A secondary benefit will be to provide a framework which others can use and further adapt to improve the
wider realm of undergraduate science, technology, engineering and mathematics (STEM) education. By
providing more hands-on experience in faculty education and training, we expect that those using this
technique will see new and novel applications of flexible laboratory technology, which they can in turn adapt and improve for their teaching environments. We envision this project as part of the cycle of innovation in STEM education that will in turn generate more new educational techniques and further improvement in teaching by supporting better training of educators.

We believe that improved hands-on experiences for educators will translate to better instruction and better laboratory exercises for the students of those instructors. In short, better hands-on training for faculty will result (indirectly) in better hands-on learning for students.

More detail on the individual project goals follows below.

Goal 1 – We plan to develop a mobile computer network configuration and management software system involving virtual computer systems that can be used specifically for computer security education, but also generally in a variety of university-level educational workshop scenarios. Our aim is to produce a state-of-the-art product that can change the way hands-on computer science workshops are delivered, and therefore improve the ability of university educators to learn about and practice new material and technologies. This system involves the integration of the following components:

- connection of host systems meeting standard configuration requirements on a wireless network.
  The standard configuration includes requirements for:
  - system resources, including main memory size, disk size, processor speed, hardware component specifications (e.g. network card) and operating system
  - installation of virtual OS support software (we are currently using VMWare and the freely available VMWare Player, though we are investigating the possible use of other products such as Microsoft’s Virtual PC [11], the open-source User Mode Linux [10], the Xen virtualization software [18], and the Parallels virtualization products [7])
  The wireless connectivity will have to be stable enough to allow the network management and workshop content delivery to proceed without problems, yet be secure from outside wireless interference as well.

- installation of client software for our system that will enable communication to the network management systems. This initial client software (and the virtual OS software or a link to it, depending on licensing restrictions) will be distributed to the participant computer systems through any portable data distribution system, such as USB drives, CDs or DVDs, or Apple iPods

- connection of one or more network management systems to this network

- establishment of a switch-based or hub-based network for the connected machines (we have achieved either using a standard network switch)
- use of custom software tools to allow the dynamic configuration, re-configuration and management of the client operating system environments
- use of custom software tools to monitor the status of the network and the client systems, including whether they are functioning correctly in the workshop environment, what services they have running, and their status in terms of the domain of the particular workshop
- use of custom tools to provide any other needed services or activities on the host network

Virtualization software provides the abstraction layer that gives us the power and flexibility to make our system work. We, and other domain experts, can create virtual client computer systems (currently in Windows and Linux, and we’re investigating generation of Mac OS X systems as well) that run on top of the workshop participants’ hardware (which in turn can be Windows or Linux). Many different virtual client systems (containing prepackaged operating system and application bundles) are already publicly available on the world-wide web, such as through the VMWare Appliances collection [13]. Still, given the unique environments that computer science instructors may want for teaching, users of our system will need to generate their own custom client systems as well. Within these virtual client systems, we can create modules for hands-on exercises in various computer science (and other) domains. Then, as the workshop begins, participant systems can register on our virtual network, and these client systems and modules can be distributed to the participants. The participants’ use of these modules can be monitored during the workshop, and the modules and client systems can even be changed or reconfigured in real time during the workshop.

We realize that there is tension between several models for providing the computer systems used with our workshop delivery system:

1) workshop instructors can provide pre-configured systems to participants, such as the laptops we have provided for the computer security workshops. The advantages include full control over the systems and a homogeneous environment, but the disadvantage is that having a pool of systems just for such workshops is expensive and not feasible for most institutions.

2) instructors can use participant provided systems (generally laptops). The advantage is that these are becoming fairly common and the participants have control over these systems, at least as far as being able to install software packages. The disadvantage is that these systems vary and do not provide an entirely homogeneous environment.

3) instructors can use existing desktop laboratories at host institutions. The advantage is that these are widely available and are (generally) homogeneous, but the disadvantages include potential lack of access and lack of control, especially for outside instructors.

The recent availability of free virtual operating system execution software such as VMWare Player [14] and the ability to create virtual networks using such software has given us a way to reach each of these
three environments. If a participant system can run VMWare Player and then host a virtual client operating system image that we provide, we can have enough control over the system to present and manage a flexible hands-on workshop environment without threatening or affecting the host computer system or the host network. VMWare Player currently runs on both Windows and Linux systems, and supports a large variety of client operating systems (many versions of Windows, many versions of Linux, FreeBSD, and Solaris). Our system currently works on our provided systems (scenario 1), and we are confident we can make it work for user-provided laptops (scenario 2) if users can pre-install VMWare Player and our client software. This has now been made easier as we can provide it rapidly through portable data technology such as USB drives, CD/DVDs, and iPods. While we can’t control the lab access issues (in scenario 3), we are investigating ways of circumventing the control issues in a host institution’s laboratory by finding ways of running our system entirely on the portable data technology. In summary, by using virtualization player and portable data technology, we see our system being useful in all three of the system scenarios.

The custom tools mentioned above have been prototyped during our previous work, but further development and refinement is necessary to make them fully functional. The first goal of the project will be the further development and integration of these various tools and functionalities into a single mobile workshop management system.

Goal 2 – We will package the above functionality with an easy-to-use interface that can be managed by the presenter. This involves not only automating many of the system tasks as noted in Goal 1, but also providing an easy-to-use interface that allows a single workshop presenter or support person to configure and manage the workshop environment on site. We therefore plan to carefully design the interface to give easy access to the needed functionality, to be usable with a minimum of training, and to support rapid configuration, re-configuration, and dynamic management of a given workshop in real time. This goal will thus require the following steps:

- identification of all functional usage scenarios
- careful interface design for both complete functionality and efficient usage
- interface testing, both through the development process and in test workshop delivery situations

A user-interface expert (Dr. Susan Harrison) will evaluate this interface and provide guidance for improvement in it.

We will also provide suitable online documentation of our system, to be used as a reference by any instructors using our system to develop their own workshop scenarios.
Goal 3 – We plan to demonstrate the value of this generalized system to educators by presenting mobile 
hands-on computer security workshops to educators and also by teaching educators how to use this 
system for their own computer science education workshops in a variety of areas. Specifically, we will 
do the following to reach these goals:

- we will re-offer the computer security workshops we have given in the past (with updates to 
  keep their content current) at SIGCSE 2009 and at two other national offerings. This will 
  allow us to demonstrate to others the functionality of the system and the possible uses of it in 
  computer security. As the scenario we create involves the dynamic addition to and removal 
  from the network of both participant and other “bait” systems for our security exercises, these 
  workshops will demonstrate the overall functionality of our system.

- we will offer a week-long workshop at the University of Wisconsin – Eau Claire to teach 
  twenty-five other computer science instructors how to use our system to develop their own 
  workshops. We envision a two-phase approach of:
    o dissecting our computer security workshop, and showing the participants how we use 
      our system, both to create the original system images to hold the content we desire 
      and to dynamically configure and manage all systems during the workshop itself.
    o Helping the participant instructors develop a workshop on the computer science topic 
      of their choice, using our system to support the content development and future 
      management of the participant’s own workshop.

- we will make the system, documentation, training materials, and sample workshop content 
  available on the world-wide web, as a resource to other educators who wish to use our 
  system.

Experience and Capability of Principal Investigator and Other Primary Participants
The PI is well qualified to carry out this project. PI Wagner is highly knowledgeable in the area of 
software development, and has significant experience in this area. He has taught project-oriented 
software engineering and database systems courses over the last fifteen years. He has been part of the 
development team for two large industrial software systems: a multi-area production and engineering 
database for compact disc production and a respirator selection expert system, both for the 3M 
Corporation. He helped design a heterogeneous computer security laboratory at the University of 
Wisconsin – Eau Claire (UWEC) that can be switched between an isolated network and a general purpose 
laboratory. He has also been the primary instructor at the computer security workshops offered at 
SIGCSE 2005, SIGCSE 2006, and SIGCSE 2007 that demonstrated the feasibility of our prototype 
system.
The other primary individuals that are part of this project are also well qualified to carry out this project. Jason Wudi was a system administrator for the UWEC’s Department of Computer Science, as well as the campus Learning and Technology Services (LTS), for six years. He has also worked as a consultant over the past three years on projects ranging from operating systems deployment to application development. Jamison Schmidt is the current system administrator for the Department of Computer Science at UWEC, and has supported the UWEC computer science department software needs over the last five years. Tom Paine has been a network specialist for UWEC LTS for the past six years. His work involves the management of the university’s network, and the use of tools and development of software to manage and monitor this network. Before that, he was a systems engineer, networks engineer, and network technician for nine years. Daren Bauer has worked on network support for UWEC Information Services for five years, with his work including diagnosis and repair of network problems as well as software tool selection for network and security management.

Wagner, Wudi, and Paine were part of a team of five individuals that implemented a mobile isolated network for the previously mentioned computer security workshops under NSF CCLI Phase 1 grant #0309818. Wudi and Paine developed tools for the following aspects of these workshops:

- multicast distribution of initial system images and configurations over the mobile network,
- a prototypical tool to execute any arbitrary scriptable command on a remote machine (including file copying, command execution, shutdown, etc.),
- a prototypical tool building on the MetaSploit framework [6] to remotely execute attacks on systems participating in the workshops as part of the hands-on exercises, and
- a tool to monitor (in real time) the status of each machine, including the maintenance of required system services for the workshops and its vulnerability to the exploits mentioned above.

Given the prior work we are building on, the workshop system will be primarily designed by Wagner, Wudi, and Paine, all of whom already have familiarity with the tools and functionality needed for this system. However, we have budgeted for the participation of two additional staff (Bauer and Schmidt) in order to gain additional networking, systems and software tool expertise. Bauer has participated in the preparation and offering of our computer security workshops, and Schmidt has managed our local computer security laboratory as part of his current job duties. Both have significant knowledge and expertise in networking and the configuration and offering of workshops using technology. We have also budgeted for three students to participate in the development and testing of this system, and to give them larger-scale software development experience. This approach is consistent with our university emphasis on faculty / undergraduate student research collaborations, and we think this is an ideal project in which to engage students in that activity.
We have also budgeted for the addition of a human-computer interaction (HCI) specialist, Dr. Susan Harrison, to assist us in the development and testing of the interface to the proposed system. We have heard from our workshop participants that ease of use would be essential for instructors to take a system and use it for their own workshop development. Harrison has written seven papers over the last fifteen years on screen design, web site usability, and web site testing, and is well qualified to assist in the development and evaluation of the interface to our system.

**Evaluation Plan**

Our evaluation plan is driven by the three goals above. For each of these goals, we have identified *expected and measurable outcomes*. We have also developed an *overall project evaluation plan* that will assist in monitoring our progress, assessing project success, and guiding future work.

Expected outcomes for goal 1 (development of a mobile laboratory software management system)
- development of an improved but still prototypical system within the first year of the project
- development of a full system within the second year of the project
- generation of positive evaluation results when used by others in our workshop or later in general use

Expected outcomes for goal 2 (development of an easy-to-use interface to the system)
- development of the interface at both levels (prototypical and final) as mentioned above
- approval of the interface as usable by a human-computer interface expert
- determination of whether the system can be supported by a single individual
- generation of positive evaluation results re: the interface by those using it – in our workshop or later in general use

Expected outcomes for goal 3 (demonstrate and teach usage of the system)
- successful application of the improved portable network system to our computer security workshop, as measured by functionality of the system during the workshop and the participant response
- successful offering of a general workshop at our institution to teach twenty-five other computer science instructors how to use our system, as measured by evaluation of the workshop and those instructors being able to complete workshop development for topics of their choosing.

Our overall evaluation plan involves the following:
- Development of system requirements and design documents for comparison with the implemented project system
• Generation of a Gantt chart for all steps of system development, and tracking progress on system development against the steps to determine project progress.
• Generating a test and evaluation plan for self-evaluation of our prototypical system, and subsequently evaluating the system according to this test plan.
• Working with Dr. Susan Harrison to develop an evaluation document for the user interface, and subsequently asking her to evaluate and critique the prototypical system interface and final system interface.
• Generating an evaluation document for overall user evaluation of our final system (in the user workshop and for subsequent outside usage), and requiring those using the system (either at our workshop in or later usage) to submit this evaluation document to us for inclusion in our overall evaluation of the system. This document will evaluate several aspects of the system, including:
  o Its overall functionality
  o Its overall usefulness
  o Its ease of use through its interface
  o Its ability to support the specific domain being worked with in a particular exercise
• Keeping statistics for the downloading and usage of the software.

Dissemination of Results
We will disseminate the results of our project in the following ways:

1. We will develop a project web site that describes the project and the software system we have developed, allows download of the software, and maintains documentation for the system. This site will include information on using the system in multiple domain areas, per our third goal. We will work to make our information on our resources available through alternate sources such as the Computing and Information Technology Interactive Digital Education Library (CITIDEL) and open source repositories such as sourceforge.org, and are aware of the need for specific inclusion of NSDL metadata tags to sustain information availability.

2. We will submit papers describing the system to national conferences such as SIGCSE or ITiCSE, and regional conferences such as the Midwest Instruction and Computing Symposium (MICS) or the regional CCSC conferences. These papers will primarily discuss the usage of this software in an educational setting, based on our test usage of the system.

3. We will hold a workshop at the University of Wisconsin – Eau Claire that will train 25 people from around the United States in the usage of our software system, and demonstrate the use of this software in the computer security areas while helping the participant instructors develop their own workshops in the computer science topic of their choice. This workshop will be advertised
nationally, and participants will be selected in part to ensure a broad geographic and institutional range of participation. Presenting this workshop will help to develop expertise which will be further shared through internet and personal contact.

4. We will hold at least three workshops at national computer science education conferences such as SIGCSE (starting in 2009), to give participants additional opportunities for hands-on experience with this system focusing on the field of computer security, and to demonstrate the possible wider usage of this system in other areas.

Results from Prior NSF Support
Wagner recently completed four years of work as a co-PI (with Andrew T. Phillips) on the NSF-funded project “CLICS – A Computational Laboratory for Information and Computer Security” [2], NSF DUE award #0309818. An initial two-year award of $99,850.00 was made in 5/2003, and a supplemental award of $31,775.00 was made in 5/2004 for the purpose of developing the computer security workshop mentioned previously. Annual reports have been submitted through the NSF FastLane system. The original grant has led to the successful development of two complete courses (1) Computer Security (CS 370) and (2) Cryptography and Network Security (CS 491) at the University of Wisconsin – Eau Claire, approximately ten security modules for use in a variety of computer science courses, two published papers [8, 16] and four presentations. The supplemental grant has led to three national and three local offerings of our two computer security workshops (1) Information Gathering through Vulnerability Assessment, and (2) System Hardening and Intrusion Detection to a total of well over 100 computer science instructors and professionals, as well as another published paper [15]. Our cyberwar laboratory work has been referenced in several other papers and presentations [e.g. 5, 17].

Conclusion
Computer science instructors are challenged to stay up to date with current issues and technology in the area of computer science. Innovative technological innovation such as the system we propose to develop further can be part of the continuing efforts to support the computer science education community.

We think that this project has intellectual merit in that it provides a tool that significantly increases the range of what we can do in computer science education training workshops and beyond. It is an innovative approach in that it combines a variety of previously developed tools and technology in a package that can be used for workshops training computer science educators. We have the resources and capabilities to successfully organize, manage, and carry out this project with the budget specified and within the requested time frame.
We think that this project has **broad impact**, as the potential usage of the workshop system to be developed extends from well-established computer science educator training workshops at conferences like SIGCSE and ITiCSE to local training workshops for computer science departments to other applications involving faculty and student laboratory exercises. Our system provides increased support for the computer science education community, which directly improves how educators can teach computer science and indirectly improves the way computer science students are educated, and in turn benefit the way computer science helps society.