The Design of NetSecLab: A Small Competition-Based Network Security Lab

Christopher P. Lee, *Member, IEEE*, Arif Selcuk Uluagac, *Graduate Student Member, IEEE*, Kevin D. Fairbanks, *Graduate Student Member, IEEE*, and John A. Copeland, *Life Fellow, IEEE*

***Abstract*—This paper describes a competition-style of exercise to teach system and network security and to reinforce themes taught in class. The exercise, called NetSecLab, is conducted on a closed network with student-formed teams, each with their own Linux system to defend and from which to launch attacks. Students are expected to learn how to: 1) install the specified Linux distribution; 2) set up the required services; 3) find ways to harden the box; 4) explore attack methods; and 5) compete. The informal write-up at the end of the lab focuses on their research into defense and attack methods, which contributes to their grade, while their competition score is dependent on their abilities to attack during the competition. Surveys were performed to evaluate the efficacy of the exercise in teaching system security.**

***Index Terms*—NetSecLab, network security, network security education, network security lab.**

I. Introduction

Modern society has entered an era where information is distributed across many uncontrolled domains (such as the Internet) and has become more dependent on networked technologies than it ever was in the past. For instance, there are many flavors of distributed networks today: wired, wireless, GPS, handheld devices, sensor networks [1], and so on. However, as the number of cyber-crime incidents has increased [2], the security of this diverse set of networks and the services they provide has become an integral part of any business today. For instance, the Internet Crime Complaint Center (IC3) [1] received 206 884 cyber-crime-related incidents in 2007 alone.

Network security courses are normally taught at the graduate school level or as an advanced elective in an undergraduate curriculum. A common approach to supplement the material that is covered in a lecture-based course is to assign homework exercises and small programming assignments to demonstrate some of the concepts that are covered. Although this introduces students to core security themes, the application of the concepts is not heavily stressed. For example, it is known that strong passwords are needed for remotely accessing a host. However, what is not stressed is what can happen if a password is compromised and how to protect the password in a hostile environment. Therefore, the first step in understanding the value that this exercise adds to a course is to consider the course in question. For instance, ECE 6612 Computer Network Security is an introductory graduate-level class in the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta. The course covers a great variety of security-related content using a lecture style of delivery. The concepts covered during class time are reinforced by homework and small exercises, such as successfully using asymmetric encryption to send private messages. This lab was designed with the purpose of increasing the amount of hands-on security experience that the students would obtain. This exercise helps to supplement the theoretical content covered in lectures by having students configure and deploy an operating system with a set of mandated services in a hostile environment. At the end of the exercise, the students will have gained a greater appreciation for the fundamentals of computer and network security.

In Georgia Tech's graduate course ECE 6612, Computer Network Security, the standard security topics are taught, but practical defense and attack are taught via the NetSecLab [3]. Homework and programming exercises are commonly used in reinforcing security concepts, and while these have proven effective, the creators of the NetSecLab want to present an engaging exercise that rewards students who find their own answers. This is critical in the face of aging security models and a rapidly changing security landscape. When students find their own answers, they learn at a much deeper and useful level.

Using student feedback as a metric for the success of the lab, positive trends have been observed. Despite the limited time in which the exercise is carried out, students report that they have found the lab informative and suggest that it continue to be a part of the course.

The rest of this paper is organized as follows. The elements of NetSecLab are introduced in Section II along with the roles that they play. The scoring metrics for the lab are also presented and explained in Section II. Section III presents a sample schedule of when to start lab preparations. The results of student surveys are presented in Section IV, followed by the positive aspects of the exercise in Section V. The conclusion is in Section VI.

II. Components of NetSecLab

The lab was designed to familiarize students with Linux security and attacks, but only requires two weeks of preparation and one week of competition. Linux was chosen because it is free and supports a lot of good security tools. To familiarize

Manuscript received June 25, 2009; revised October 02, 2009; accepted April 01, 2010. Date of publication April 26, 2010; date of current version February 02, 2011.

The authors are with the Communications Systems Center, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: chrislee@gatech.edu; selcuk@gatech.edu; kevin.fairbanks@gatech.edu; john.copeland@ece.gatech.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2010.2048215

[1] The Internet Crime Complaint Center (IC3) is a partnership between the Federal Bureau of Investigation (FBI), the National White Collar Crime Center (NW3C), and the Bureau of Justice Assistance (BJA).

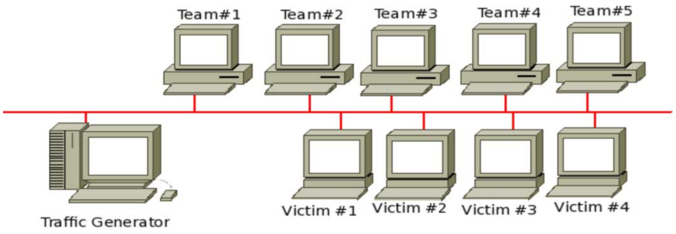


Fig. 1. Network setup of the NetSecLab.

students with Linux, they are first broken into groups with at least one person per group who is familiar with Linux. Then, each group is responsible for formatting and installing a hard drive (borrowed from the lab for the competition). The next two weeks are spent learning how to install the required services; how to upgrade, configure, and secure those services; and how to attack.

The selection of services like an SSH server, a Web server, an SMTP server, and an FTP server fulfills certain requirements for the lab. SSH is used to provide a secure way for the traffic generator to log into the teams' boxes and to check local services like mail and remote services like connecting to other boxes. The Web server allows the deployment of custom, buggy PHP services. The traffic generator uses the SMTP to check that the students correctly configured it to deliver mail, but not to allow open-relay. FTP and/or Telnet makes the traffic generator's account on the team's box vulnerable to password sniffing.

The competition takes place in a private LAN without any connection to the outside world. The network configuration of the NetSecLab is illustrated in Fig. 1. The details of each component are given in Sections II-A–G.

A. Team Formation

Six weeks before the competition, the class is surveyed for their proficiency with Linux and their desire to lead a team. A sample survey is shown in the Appendix. Four weeks before the competition date, the teams are created, and removable hard drives are given to each team within the week. The students are then responsible to set times to meet and what roles each will play in the overall research and work. The instructor typically recommends that the team members work together on the installation and setup of their system, but from there divide the research for defense and attack.

Determining the number of teams to divide the class into is highly dependent on the general skill level of the class, the number of people who want to be captains, the number of hard drives that are working and available, and the number of lab assistants available to run the competition and grade the reports. Typically, ECE 6612 has had between four to six teams, with occasionally an additional team participating via virtual private network (VPN) from remote campuses (for example, the Georgia Tech Savannah campus).

After teams are determined, each team selects a team captain, who is mainly responsible for communicating with the lab organizer on issues that matter to the competition. The team leaders are required to pass any important information or updates from the lab organizer to their team members and also to notify the class instructor/teaching assistant of any problems they face. The trend observed over five years is that typically students with more Linux skills volunteer to become a team leader. The NetSecLab can be implemented without team leaders, but it is much easier to maintain communication with six captains than with 50 students and to build an understanding of expectations. Since the interpretation of the rules can vary, the captains are invited to ask for clarification and permission to use various techniques during the competition.

B. Traffic Generator

The traffic generator is a set of PERL and Expect scripts. The scripts use a configuration file containing username and password combinations for each team, along with their computer's IP addresses and any other information needed to verify their services. The box on which the traffic generator runs must be locked down, usually by not running any services at all and performing strong checks on every input. The traffic generator currently has modules to perform the following patterns:

- SSH into team boxes and check mail;
- SSH into one team box and attempt to SSH or Telnet into another team's box from the first team's box;
- send e-mail to the traffic generator's accounts on each team box via SMTP;
- transfer files to the accounts via FTP;
- transfer files to the accounts via Samba;
- fetch HTML objects from a URL;
- attempt to send mail to a team box with a destination account of another box;
- Telnet into the accounts on the team boxes;
- select, insert, and delete records from the MySQL databases on the team boxes.

C. Team Boxes

Each team receives a removable hard drive (usually on the order of 20–40 GB) that can be inserted into the machines in the networking lab. Each team is assigned an initial IP address to use on the private LAN and is asked to provide back to the exercise manager, in a secure manner, a username and password combination to be used by the traffic generator during the competition. The exercise manager sends a description of the required operating system and services to the team captains, and then the teams can begin to set up their team box.

After the teams have installed the basic services, they need to research how to upgrade and secure each service. For example, knowing that the password will be sent in the clear via FTP or Telnet, they research techniques to lock down the traffic generator's account to prevent privilege escalation attacks, which are worth a lot of points to a successful attacker.

D. Services

Typically, six or seven standard services are required by the lab description, which change each year. SSH and one plain text authentication protocol (e.g., Telnet) are always present in the list, with the other chosen for known vulnerabilities or common misconfigurations (e.g., open-relay on Sendmail). An example list of services is given in Table I.

TABLE I
EXAMPLE LIST OF SERVICES FOR TEAM BOXES

#	Service	Port #
1	Ssh	22
2	telnet	23
3	SMTP	25
4	HTTP	80
5	Samba	139
6	Mysql	3306

In addition to the standard services, one custom-written service is required to be run on the boxes. The custom service, similar to the approach taken by iCTF [4], requires students to study the source code of an application and patch it to defend against attacks. The vulnerability in the code is typically very simple. For example, the "rotten" server, written in C, reads input from a TCP client into a buffer and then performs a ROT13 operation on all the alphabetic characters. It is vulnerable to a buffer overflow, but the shellcode must be "pre-rotted" to execute correctly. Another example is the FaultyBank PHP Web application, which was vulnerable to several command execution vulnerabilities and an SQL injection.

E. Victim Boxes

Along with the team boxes and the traffic generator, there are known vulnerable boxes, called "victim boxes," placed onto the lab network about two weeks prior to the competition. These boxes help students gain confidence and practice network enumeration, service identification, and exploitation. This part of the lab, unlike the rest, stays relatively the same each year.

In the first year, three machines were created and then reused each year: a Redhat 8.0, a Redhat 7.0, and Redhat 6.2 box. Nonetheless, in the following years, virtual machine (VM) images [5], [6] were made of each of the boxes and used instead of the original machines. Also, a Windows XP SP1 image was added to the list of vulnerable images. Each OS has certain vulnerabilities, and students are expected to search for those and learn ways to exploit them. For instance, the Redhat 8.0 image runs Samba 2.2.8, which is vulnerable to a buffer overflow attack, as described in CVE CAN-2004–0686 [7]; the Redhat 7.0 image runs LPRng 3.6.22, which is vulnerable to a format string vulnerability, as described in CVE-2000–0917 [7]; and the Redhat 6.2 box runs Wu-ftp 2.6.0, which is vulnerable to a buffer overflow, as described in CVE CA-1999–13 [7].

F. Competition Rules

Like all good games, this competition needs rules, but those rules should not be too strict nor should they allow for activities that strongly detract from the learning goals. The first rule is that no team is allowed to perform a denial-of-service attack on the network, the services on other team's boxes, or the ability for a team to use their own computer (i.e., locking them out of their accounts). Defending from packet floods, service exhaustion, and malicious destruction of the operating system is simply outside the scope of the exercise, hard to grade, and highly frustrating to students.

The second rule is to keep services up to the whole network throughout the competition. It would be too easy to defend a service if it were blocked or shut down, so the traffic generator tests the service to make sure that it is up. An opposing team may take down a service, but only temporarily and only if it is necessary for an exploit.

The third rule is to keep all evidence. Logs should never be erased, altered, or otherwise limited. The logs are vital to the teams' reports and to the grading of reports.

Aside from that, there is a general rule that if a team wants to try something nonstandard, they ask the lab managers for approval. For example, one team wanted to use a VM to run their services and use the host OS to attack. The lab manager approved this approach on the condition that all services were run in the VM and that full points would be allocated to the opposing team if they could exploit the root account of the VM. Another example is that another team wanted to multihome their box so that they could change the attacking IP over time. This was permitted as well, and the team was given 20 IPs to avoid colliding with another team's IP.

G. Scoring

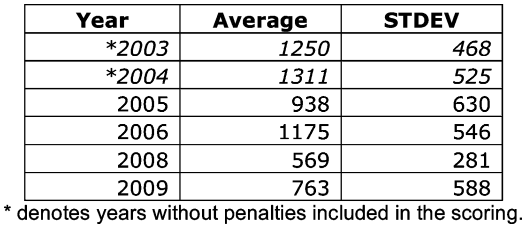
There are two scores for the NetSecLab, a competition score and a lab report score. The two scores are independent, although they usually correlate highly. The competition score is displayed publicly to the teams and on a poster in the hallway outside the networking lab. The report score contributes to the students' grades. The competition score is determined as follows:

- mapping the network (2 pts. per IP address);
- mapping services (20 pts. per box);
- OS detection (10 pts. per victim box);
- gaining user access to a victim box (30 pts.);
- gaining user access to a team box (50 pts.);
- gaining root access to a victim box and retrieving the shadow hash file (150 pts.);
- gaining root access to a team box and retrieving the shadow hash file (250 pts.);
- team box becomes compromised (-300 pts.).

Moreover, teams are encouraged to think up and implement new ways of exploiting machines, and such efforts are rewarded. Timing, efficiency, and creativity are given bonuses. For instance, if the teams can achieve certain tasks within specified times, they can accrue the following bonus points: 200 additional points are awarded if completed within 5 min, 150 points if done before 10 min, 100 points if done within 15 min, and 50 points if completed within 20 min. Finally, teams can receive a "Super Bonus" outside of the competition if they successfully crack a password (200 pts. per password), and they are allowed to continue to crack passwords up until the report submission deadline (from stolen encrypted password files).

Based on observations in previous NetSecLabs, a score of about 500 points is considered passing, a score of 700 is good, a score of 1000 is considered excellent, and anything above 1500 is considered unbelievable. The previous years' averages are given in Table II. Note that the point system in the first two years, 2003 and 2004, did not include the penalties and bonuses as do the later years. 2008 and 2009 had lower scores because

TABLE II
AVERAGE COMPETITION SCORES FOR NETSECLAB



Year	Average	STDEV
**2003*	*1250*	*468*
**2004*	*1311*	*525*
2005	938	630
2006	1175	546
2008	569	281
2009	763	588

* denotes years without penalties included in the scoring.

newer Linux distributions are more secure in their default configurations and there is more documentation available on how to secure systems.

III. SUGGESTED TIME SCHEDULE FOR PREPARATION OF THE NETSECLAB

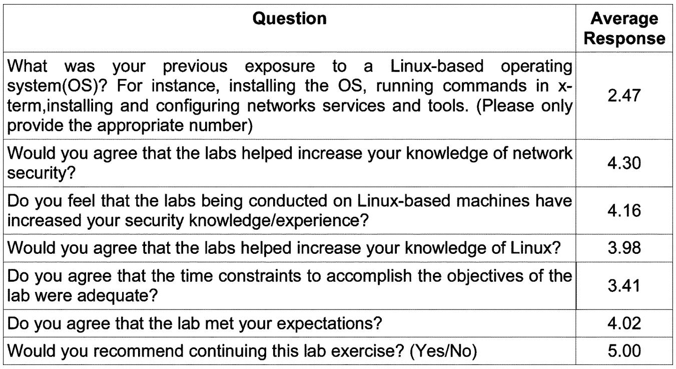
A two-lecture-hour competition actually requires some advance preparation time. In this section, the preparation, important milestones, and decisions in designing NetSecLab are discussed. Specifically, a timetable is suggested for the preparation of the NetSecLab.

- **Survey students for their skill levels and form teams:** This can be done in the first weeks of the class. Students will probably have different backgrounds and may not have previous exposure to the material. Thus, this is a very important step to distribute students of various skill levels to form teams that are as homogeneous as possible.
- **Choose the OS for team boxes:** The choice of the Linux-based OS is done by the class instructor/teaching assistant during the early phases while the class is under way. There are many freely available Linux distributions over the Web, and any can be chosen. However, a general rule-of-thumb is to choose one that is slightly unsupported and has known vulnerabilities. A particular Linux flavor can be chosen and utilized in every NetSecLab, however currently the practice is to use a different OS for each year. This can be done eight weeks prior to the competition day.
- **Decide services for team boxes:** Services listed in Table I will generally suffice. However, this list is not an exhaustive one; new services can be added each year to increase variety. This step can be done seven weeks prior to the competition day.
- **Choose the OS for victim boxes/services:** It is important to choose an earlier version of a Linux distribution as it is more likely to have more known vulnerabilities. However, finding an earlier version of particular Linux distribution can be nontrivial. This can be done six weeks prior to the competition day.
- **Install victim machines:** It is better to install victim boxes earlier than the competition days so the students can begin practicing their exploits and can collect some more information about the vulnerabilities known for the victim boxes. This can be done four weeks prior to the competition day.

IV. STUDENT FEEDBACK

This section presents the results and interprets the data of a survey sent out to students who had taken ECE 6612, and summarizes comments that were consistent across the group of surveys. The average response is compiled from 58 individual surveys spread across two semesters. The students were asked to respond to the questions on a 1–5 scale, with 1 being the lowest rating and 5 being the highest rating. A rating of 3 means that the student is being neutral or is rating their skill level at intermediate. The only binary question in the survey was whether they felt the lab should be continued.

TABLE III
SURVEY RESULTS

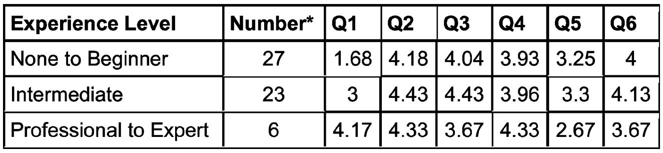


Question	Average Response
What was your previous exposure to a Linux-based operating system(OS)? For instance, installing the OS, running commands in x-term,installing and configuring networks services and tools. (Please only provide the appropriate number)	2.47
Would you agree that the labs helped increase your knowledge of network security?	4.30
Do you feel that the labs being conducted on Linux-based machines have increased your security knowledge/experience?	4.16
Would you agree that the labs helped increase your knowledge of Linux?	3.98
Do you agree that the time constraints to accomplish the objectives of the lab were adequate?	3.41
Do you agree that the lab met your expectations?	4.02
Would you recommend continuing this lab exercise? (Yes/No)	5.00

The results listed in Table III capture the perceptions that students have about the exercise. With an average of 2.47, the students gave themselves a low self-assessment of their Linux skills. Although this does not mean they are absolutely ignorant of everything about the OS, it does indicate that most students are probably not familiar with common Linux utilities or working from the command-line interface. As the questions directly related to the lab have an average above 3, this has been interpreted to mean that most of the students gain from the experience. Since NetSecLab is designed to support the material taught in class by providing a hands-on experience for the students, and the average ratings for categories related to increasing security knowledge are all above 4, it is felt that the educational objectives are being met. One thing to note is the binary nature of the question about continuing the lab in future classes. Here, a "No" answer is assigned a score of 0, and a "Yes" answer is assigned a score of 5. The fact that the average rating for this question was 5.00 denotes a unanimous agreement among all of the surveys about the value the lab adds. Among the comments, three common sentiments emerged. The first is that the students expressed a desire for the project to be longer and for it to carry more weight within the final grade of the course. The latter was a concern about the size of the groups as well as the inclusion of remote and video students. These comments will be discussed in Section VI.

Table IV displays a breakdown of the surveyed student population on the basis of Linux skill level. Three broad groupings have been made based upon the students' self evaluation of their skill. No skill equates to a ranking of 1, and expert skill equates to a ranking of 5. This table shows that most of the students participating in NetSecLab have an average to a beginner level of Linux proficiency. The Q1 through Q6 in the table correspond to the questions in Table III, with the last question being omitted. For example, results of Q2 represent the average response of the subgroup to the question regarding whether the lab helped

TABLE IV
SUBGROUP RESULTS



Experience Level	Number*	Q1	Q2	Q3	Q4	Q5	Q6
None to Beginner	27	1.68	4.18	4.04	3.93	3.25	4
Intermediate	23	3	4.43	4.43	3.96	3.3	4.13
Professional to Expert	6	4.17	4.33	3.67	4.33	2.67	3.67

increase the students' knowledge of network security. Across all three subgroups, an average greater than 4 suggests that the lab is achieving its intended purpose. Overall, most of the results across subgroups are positive, but it should be noted that students with more experience with the Linux OS were more critical of the time constraints given for the exercise (Q5) and generally wanted more from the lab (Q6). One surprising result is that the students who had more Linux experience felt that they learned more about the OS as a result of the lab than students with less experience (Q4). This result may be indicative of the more Linux-oriented students taking on the role of the main system operator while students at with lower levels of proficiency take a more observational stance.

V. Benefits of NetSecLab

The NetSecLab was first introduced to the students of ECE 6612 Network Security class at the Georgia Institute of Technology in Fall 2003. Since then, NetSecLab has been offered each semester. In this section, the benefits of NetSecLab based on observations and students' feedback over time are discussed.

- NetSecLab forces students to consider all aspects of security including physical, network, and computer security and, in some rare cases, social engineering. For instance, students install and secure an OS while having to keep their hard disk in trusted hands, topics which are not normally covered in the network security class content.
- The private network setting of the NetSecLab allows students to try all of their implementations and new or old ideas without any fear of the attacks bleeding over onto a production network.
- Graduate students often have heterogeneous backgrounds (e.g., nationalities) and experience levels. Placing students in teams as homogeneous as possible, in terms of average previous Linux experience, helps them learn from each other in a group setting.
- After the competition is finished, groups submit a report and give a presentation in the class. The report and presentation provides students with the opportunity to exchange their experience and their findings with their peers in the class.
- Students are encouraged to increase their understanding in relatively harder concepts of network security. This is done through points being awarded for the creativity and efficiency in implementing attacks.
- Virtual machines are used for victim boxes with earlier versions of Linux distributions or unpatched Windows services chosen for the OS. The use of virtual boxes allows victims to be easily reset to a consistent state after an attack.
- Although NetSecLab was originally designed as a component of a graduate course, it is highly adaptable and can be offered in junior- or senior-level network security classes.

VI. Related Work

Network security is a necessary part of any undergraduate and graduate curriculum in computer engineering and science fields [8]–[11]. There are several ways for teaching network security topics in a classroom setting. In this section, related efforts from the literature in three categories are broadly presented.

In one category, topics are taught in a classroom environment where only dedicated laboratory exercises are carried out by students. In these cases, students are expected to spend their time mostly in an isolated lab to complete specific tasks. There may or may not be explicitly associated lecture material to accompany the laboratory sessions. Examples of these include [12]–[14].

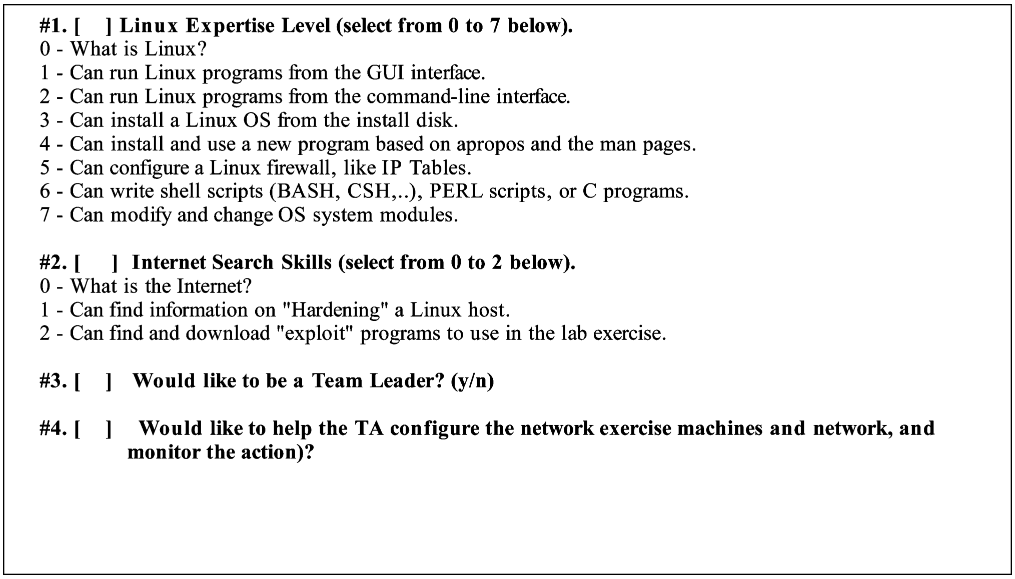
In the second category, security lectures in the classroom are supplemented with individual tools. These are generally demos of a specific security topic, and not all of the lecture material may have a corresponding demo. Students are expected to interact with these tools and complement their understanding of the security topics, and there may not be a dedicated lab environment. Examples of related work in this category are [15] and [16].

In the last category, students are expected to supplement their understanding of topics with homework or programming projects assigned by the class instructor. This is by far the most commonly adopted strategy by the educators due to its less demanding need for resources.

NetSecLab is fundamentally different from the aforementioned approaches. It is an in-class competition-based friendly exercise. Its main purpose is to supplement the classroom lecture content with hands-on experience, and it is administered toward the end of the semester in an isolated lab environment. Contrary to [4], [17], and [18], it is small and class-based, and only students who are taking the Network Security class are allowed to participate. It can well be integrated either into normal lecture-based classes (i.e., category 2) or into the body of dedicated labs (i.e., category 1).

VII. Conclusion

In this paper, the design of NetSecLab [3], an exercise developed for the purpose of increasing the amount of hands-on experience obtained by students in a lecture-based class environment, has been presented. The course in which the current version of NetSecLab is practiced is the first graduate-level Network Security class for many students where fundamental security concepts are addressed. This results in student skill levels being very heterogeneous due to differing amounts of prior knowledge about the subject matter. In order to make the competition fairer, larger group sizes are required to ensure that more competent students are paired with fairly inexperienced students. This makeup results in the weight of the exercise on the final grade being fairly light as the purpose of the lab is to increase the student experience, not evaluation. This condition also presents the opportunity for students to learn from each other through teamwork. Another factor that influences



#1. [ ] Linux Expertise Level (select from 0 to 7 below).
0 - What is Linux?
1 - Can run Linux programs from the GUI interface.
2 - Can run Linux programs from the command-line interface.
3 - Can install a Linux OS from the install disk.
4 - Can install and use a new program based on apropos and the man pages.
5 - Can configure a Linux firewall, like IP Tables.
6 - Can write shell scripts (BASH, CSH,..), PERL scripts, or C programs.
7 - Can modify and change OS system modules.

#2. [ ] Internet Search Skills (select from 0 to 2 below).
0 - What is the Internet?
1 - Can find information on "Hardening" a Linux host.
2 - Can find and download "exploit" programs to use in the lab exercise.

#3. [ ] Would like to be a Team Leader? (y/n)

#4. [ ] Would like to help the TA configure the network exercise machines and network, and monitor the action)?

Fig. 2. A sample background survey for the NetSecLab.

group size, and thereby the entire experience, includes the availability of resources. To increase fairness, homogeneous hardware should be used between the groups. With more resources, the number of groups can increase, resulting in a smaller number of people per group. Also, because the students taking the course are graduate students, a certain amount of research is expected so that they learn how to set up and configure services for themselves. Knowing this, a balance must be found in group sizes, as too small a group may not be able to divide and find exploits and vulnerabilities in the research load adequately.

The implementation of NetSecLab can be altered for more advanced security classes. In this case, the lab may be made part of an ongoing semester project where students should meet deadlines throughout the semester for configuring and running certain applications. Moreover, as the skill level in a more advanced security course will be less heterogeneous and a certain level of understanding is expected *a priori*, the group's size can be reduced. In this setting, the lab would carry more weight in the final grade as well as increase the amount of hands-on security experience students would gain.

Some challenges that must be considered are how to implement the lab if the course has remote students as well as video students. In these cases, a key factor in how these students would be able to participate would be whether they were taking the course in sync with regular students or if there were a delay in when these students receive the lectures. In the former case, the students may be on a remote campus or taking the course via video, but receive the lectures either in real time or with very little delay (i.e., downloaded the next day). For this case, one solution that has been practiced is gathering these students into one group and setting up a VPN connection for them into the NetSecLab LAN. This solution will only work if a number of remote or video students are on the same campus. In the case where remote and video students are spread into disparate locations, or the delay in the students receiving the lecture materials is more than a couple of days, a different approach has been practiced. Normally, these students are divided evenly among the local groups, and their primary contribution to the project is limited to attack, defense, and application configuration research.

The feedback provided by students, accentuated by a unanimous suggestion to continue the exercise, can be interpreted to mean that NetSecLab meets all of its primary objectives. The students must apply the concepts taught in the course on a Linux distribution. This increases their knowledge of Linux and, most of all, reinforces their security knowledge. Also, due to the nature of the lab, the students gain a more complete picture of security as they must install, configure, and secure their respective systems. This reinforces host as well as network security concepts.

There are several other institutions, projects, and class-based competitions [4], [12], [13], [19] that exist, which aim to teach security concepts with very focused hands-on experience. These efforts and tool-oriented labs, such as NetSecLab, together with future improvements in open-source software, will help fill the gap between the theory and the hands-on experience.

Appendix

In order to form the teams, students are first given a survey from which their skill levels are determined. Essentially, the purpose of the survey is to distribute students of varying skill levels to form teams as equally footed as possible. This is a very simple survey with the sample questions given in Fig. 2.

References

[1] S. Uluagac, C. P. Lee, R. A. Beyah, and J. A. Copeland, "Designing secure protocols for wireless sensor networks," in *Proc. 3rd WASA*, Dallas, TX, Oct. 2008, pp. 503–514.

[2] "2007 Internet crime report," Internet Crime Complaint Center, Federal Bureau of Investigation, The National White Collar Crime Center, and Bureau of Justice Assistance, 2007.

[3] Communications Systems Center (CSC) NetSecLab homepage, 2009 [Online]. Available: www.csc.gatech.edu/NetSecLab.html

[4] "UCSB capture the flag," UCSB homepage, 2009 [Online]. Available: http://www.cs.ucsb.edu/~vigna/CTF/

[5] VMware 2009 [Online]. Available: http://www.vmware.com

[6] "Qemu git repositories," 2009 [Online]. Available: http://www.qemu.com

[7] "Common vulnerabilities and exposures (CVE)," 2009 [Online]. Available: http://cve.mitre.org/

[8] K. Petrova, A. Philpott, P. Kaskenpalo, and J. Buchan, "Embedding information security curricula in existing programmes," in *Proc. 1st InfoSecCD*, Kennesaw, GA, 2004, pp. 20–29.

[9] Taylor and R. Shumba, "Security education: A roadmap to the future," in *Proc. 39th SIGCSE*, Portland, OR, 2008, pp. 459–460.

[10] M. Bishop, "Education in information security," *IEEE Concurrency*, vol. 8, no. 4, pp. 4–8, Oct.–Dec. 2000.

[11] Border and E. Holden, "Security education within the it curriculum," in *Proc. 4th CITC4*, Lafayette, IN, 2003, pp. 256–264.

[12] R. Abler, D. Contis, J. Grizzard, and H. Owen, "Georgia Tech Information Security Center "hands-on" network security laboratory," *IEEE Trans. Educ.*, vol. 49, no. 1, pp. 82–87, Feb. 2006.

[13] A. S. Uluagac, T. Fallon, W. Thain, and J. A. Copeland, "Development of undergraduate network security labs with open source tools," in *Proc. ASEE Annu. Conf. Composition Exhib.*, Austin, TX, Jun. 2009.

[14] J. M. Hill, C. A. Carver, J. W. Humphries, and U. W. Pooch, "Using an isolated network laboratory to teach advanced networks and security," in *Proc. 32nd SIGCSE*, Charlotte, NC, 2001, pp. 36–40.

[15] X. Yuan, P. Vega, J. Xu, H. Yu, and Y. Li, "Using packet sniffer simulator in the class: Experience and evaluation," in *Proc. 45th ACM-SE*, Winston-Salem, NC, 2007, pp. 116–121.

[16] A. Riccioni, E. Denti, and R. Laschi, "An experimental environment for teaching Java security," in *Proc. 6th PPPJ*, Modena, Italy, 2008, pp. 13–22.

[17] T. A. Yang, K. Yue, M. Liaw, G. Collins, J. T. Venkatraman, S. Achar, K. Sadasivam, and P. Chen, "Design of a distributed computer security lab," *J. Comput. Small Coll.*, vol. 20, no. 1, pp. 332–346, 2004.

[18] J. R. Aman, "Black hat/white hat: An aggressive approach to the graduate computer security course," *J. Comput. Small Coll.*, vol. 22, no. 2, pp. 52–58, 2006.

[19] W. Du and R. Wang, "SEED: A suite of instructional laboratories for computer security education," *J. Educ. Resour. Comput.*, vol. 8, no. 1, pp. 1–24, 2008.

Christopher P. Lee (M'01) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, in 2001, 2005, and 2009, respectively, as a member of the Communications Systems Center.

He has worked extensively in usable security and developed visualizations for firewalls, intrusion detection systems, Honeynets, and forensics. He is a core member of the Honeynet Alliance, the Distributed Honeynets Project, and runs the Georgia Tech Honeynet. He also teaches Information Assurance classes. His current research is on Botnets tracking and modeling

Arif Selcuk Uluagac (S'08) received the B.Sc. degree in computer engineering from the Turkish Naval Academy, Tuzla, Istanbul, Turkey, in 1997, and the M.Sc. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2002. He is currently a Ph.D. candidate with the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, as a member of the Communications Systems Center.

Mr. Uluagac is a Student Member of the ACM and ASEE. He received the 2007 Outstanding ECE Graduate Teaching Assistant Award from the School of ECE, Georgia Institute of Technology.

Kevin D. Fairbanks (S'07) received the B.S. degree (Summa Cum Laude) in electrical engineering with a computer concentration from Tennessee State University, Nashville, in 2005, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 2007 and 2010, respectively.

He is a Graduate Researcher with the Network Security and Architecture Lab, Georgia Tech, where his advisor is Dr. Henry Owen. His research interests include network security and digital forensics

John A. Copeland (LF'07) received the B.S., M.S., and Ph.D. degrees in physics from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1962, 1963, and 1965, respectively.

He holds the John H. Weitnauer, Jr. Chair as a Professor with the School of Electrical and Computer Engineering, Georgia Tech, and is a Georgia Research Alliance Eminent Scholar. He is the Director of the Communications Systems Center (CSC). This center is doing research on digital communication networks, including wireless sensor networks and WiFi and WiMAX networks, with emphasis on providing security and quality of service. In 2000, he invented the StealthWatch system for network security monitoring and founded LANcope, Inc., in Atlanta, GA, which today has deployed StealthWatch on over 100 corporate, government, and defense networks. Prior to joining Georgia Tech in 1993, he was Vice President of Technology at Hayes Microcomputer Products, Atlanta, GA, from 1985 to 1993, where he was responsible for the development of modems with data compression and error control and for Hayes' representation on CCITT and ANSI standards committees. He was Vice President of Engineering Technology at Sangamo Weston, Inc., Atlanta, GA, from 1982 to 1985, where he was responsible for R&D groups at 10 divisions. He began his career at Bell Labs, Murray Hill, NJ, serving from 1965 to 1982, conducting research on semiconductor microwave and millimeter-wave devices. Later, he supervised a group that developed magnetic bubble computer memories. In 1974, he led a team that designed CMOS integrated circuits, including Bell Labs' first microprocessor, the BELLMAC-8. His last contributions at Bell Labs were in the area of lightwave communications and optical logic. He has been awarded 43 patents and has published over 50 technical articles.

Dr. Copeland, was awarded the IEEE's Morris N. Liebmann Award in 1970 for his work on gallium arsenide microwave devices. He has served as Editor of the IEEE TRANSACTIONS ON ELECTRON DEVICES. He served on the Board of Trustees for the Georgia Tech Research Corporation from 1983 to 1993 and as Director of the Georgia Center for Advanced Telecommunications Technology from 1993 to 1996.