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

Christopher P. Lee, A. Selcuk Uluagac, Kevin D. Fairbanks, John A. Copeland

Communications Systems Center, School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332, USA
{selcuk,chrislee,kevin.fairbanks}@gatech.edu,
john.copeland@ece.gatech.edu

Abstract

This paper describes a competition-style of exercise to teach system and network security and 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 dependant on their abilities to attack during the competition. Surveys were performed to evaluate the efficacy of the exercise in teaching system security.

Keywords: *NetSecLab, Network Security Education, Network Security Lab, Network Security*

I. Introduction

Modern society has entered an era where information is distributed across many uncontrolled domains (e.g., 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, hand-held devices, sensor networks [1], etc. Many of these networks are hybrid in nature and have rich sets of networking functionalities (e.g., multimedia, location based services). 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. Therefore, this situation compels us to better network security practices.

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 that demonstrate some of the concepts that are covered. Although this is an accepted practice and introduces the 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

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).

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. 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 ECE6612, 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 are proven effective, the creators of the NetSecLab want to present an engaging exercise that rewards students to 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 is 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 general technical details and the competition format of the NetSecLab are articulated in this section.

The lab was designed to familiarize student with Linux security and attacks, but only require two weeks of preparation and one week of competition. To familiarize students with Linux, they are first broken into groups with at least one person per group who is familiar with Linux. We choose and recommend the Linux operating system because it is free (i.e., no licensing) and there exist a lot of good security tools. 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, learning 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 have 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.

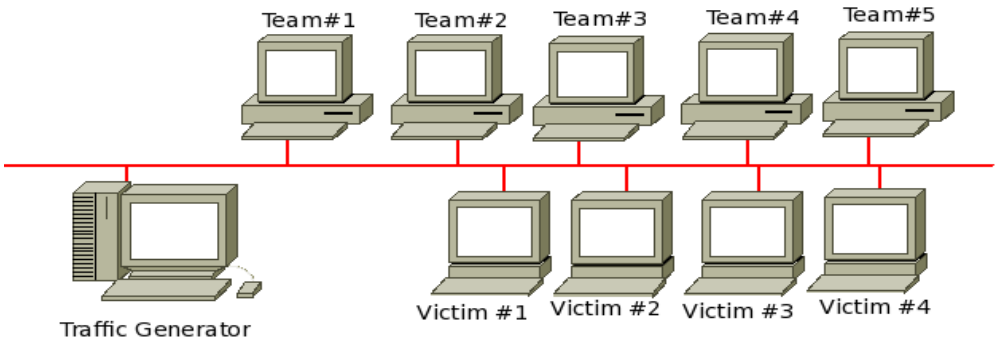

Figure 1: Network setup of the NetSecLab

The competition takes place in a private LAN without any connection to the outside world. The network configuration of the NetSecLab is illustrated in Figure 1. The details of each component are given in the following subsections.

A. Team Formation

6 weeks before the competition, the class is surveyed for their proficiency with Linux and their desire to lead a team. An example survey is show in Appendix A. 4 weeks before the competition date, the teams are assigned 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 dependant 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, ECE6612 has had between 4 to 6 teams, and occasionally an additional team participating via VPN from remote campuses (e.g., 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, updates from the lab organizer to their team members and also notify the class instructor/teaching assistant of any problems they face. The trend observed over five years is that typically students with more skills volunteer to become a team leader. The NetSecLab can be implemented without team leaders, but it is much easier to maintain communication with 6 captains than 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 do 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 email 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 on 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, 6 or 7 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 mis-configurations (e.g., open-relay on Sendmail). An example list of services is given in Table 1.

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.

#	Service	Port#
1	ssh	22
2	telnet	23
3	SMTP	25
4	HTTP	80
5	samba	139
6	mysql	3306

Table 1: An example list of services for team boxes

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, 3 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 images [5, 6] were made of each the boxes and used instead of the original machines. Also, a Windows XP SP1 image was added to the list of vulnerable images. Each certain O/S 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].

E. 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 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 non-standard, they ask the lab managers to get approval. For example, one team wanted to use a virtual machine to run their services and use the host O/S 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 multi-home their box so that they could change the attacking IP over time. This was permitted as well, but was given 20 IPs so as to not accidentally collide with another team's IP.

F. 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' grade. 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 minutes, 150 points if done before 10 minutes, 100 points if done within 15 minutes, and 50 points if completed within 20 minutes. Lastly, 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 below in ***Table 2***. Note that the point system in the first two years, 2003 and 2004, did not include the penalties and bonuses as the later years.

Year	Average	STDEV
*2003	1250.833333	467.696661
*2004	1310.75	525.2195573
2005	937.6666667	629.7903355
2006	1174.5625	546.1409706
2008	569.128	280.674
2009	763.145	588.224

Table 2. The average competition scores for NetSecLab

III. A 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 time table 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 varying student skill levels to form teams 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 un-supported and has known vulnerabilities. A particular Linux flavor can be chosen and utilized in every NetSecLab, however, currently the practice is to utilized a different O/S for each year. This can be done 8 weeks prior to the

competition day.

- **Decide services for team boxes:** Services listed Table 1 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 7 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 non-trivial. This can be done 6 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 4 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 have 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 to 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.

The results listed in Table 3 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

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

Table 3: Survey Results

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 1. 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.

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

Table 4: Subgroup Results

Furthermore, Table 4 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 ranking of 1 and expert skill equates to a level 5. This table shows that most of the students participating in NetSecLab have either average to a beginner level of Linux proficiency. The Q1 through Q6 in the table correspond to the Questions in Table 3 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 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 O/S

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 O/S 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 Georgia Institute of Technology in the Fall of 2003. Since then, it has been offered each semester of ECE 6612. 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 O/S 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, 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 encourages 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 utilized for victim boxes with earlier versions of Linux distributions or unpatched Windows services chosen for the O/S. The use of virtual boxes allows victims to be easily reset to a consistent state after an attack.
- Although NetSecLab was originally 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, we broadly present such related efforts from the literature in three categories.

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 an explicit 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 demo 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. [15, 16] are the examples of related work in this category.

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 nature for resources.

NetSecLab is fundamentally different from 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 is administered towards the end of the semester in an isolated lab environment. Contrary to [4, 17, 18], it is a small in-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 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. Other factors that influence group size and thereby the entire experience includes resources. To increase fairness, homogeneous hardware should be used between the groups. With a larger amount of resources, the number of groups can increase resulting in a smaller number of people per a group. Also, because the students taking the course are graduate students, a certain amount of research is expected so that they learn how to setup and configure services for themselves. Knowing this, a balance must be found in group sizes as too small of 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 throughout

the semester students should meet deadlines 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 groups 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 highly depends on whether they are taking the course in sync with regular students or if there is delay between 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 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 Virtual Private Network 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 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 where their skill levels are determined. Essentially, the purpose of the survey is to distribute varying student skill levels to form teams as equally-footed as possible. This is a very simple survey with the sample questions given in Figure 2.

```
#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)?
```

Figure 2: A Sample background survey for the NetSecLab

References

[1] A. S. Uluagac, C. P. Lee, R. A. Beyah, and J. A. Copeland, ”Designing Secure Protocols for Wireless Sensor Networks,” Proceedings of the 3rd International Conference on Wireless Algorithms, Systems and Applications (WASA), Dallas, Texas, October 2008.

[2] Internet Crime Complaint Center, “2007 Internet Crime Report”, Prepared by Federal Bureau of Investigation, The National White Collar Crime Center, and Bureau of Justice Assistance

[3] Communications Systems Center (CSC) NetSecLab Homepage, www.csc.gatech.edu/NetSecLab.html, Accessed July 2009

[4] UCSB Homepage "UCSB Capture The Flag", http://www.cs.ucsb.edu/~vigna/CTF/ Accessed July 2009

[5] http://www.vmware.com , Accessed July 2009

[6] http://www.qemu.com , Accessed July 2009

[7] Common Vulnerabilities and Exposures (CVE), http://cve.mitre.org/, Accessed July 2009

[8] Petrova, K., Philpott, A., Kaskenpalo, P., and Buchan, J. 2004. Embedding information security curricula in existing programmes. In *Proceedings of the 1st Annual Conference on information Security Curriculum Development* (Kennesaw, Georgia, October 08 - 08, 2004). InfoSecCD '04. ACM, New York, NY, 20-29.

[9] Taylor, C. and Shumba, R. 2008. Security education: a roadmap to the future. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Portland, OR, USA, March 12 - 15, 2008). SIGCSE '08. ACM, New York, NY, 459-460.

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

[11] Border, C. and Holden, E. 2003. Security education within the IT curriculum. In *Proceedings of the 4th Conference on information Technology Curriculum* (Lafayette, Indiana, USA, October 16 - 18, 2003). CITC4 '03. ACM, New York, NY, 256-264.

[12] Abler, R., Contis, D., Grizzard, J., Owen, H., “Georgia Tech Information Security Center “Hands-On” Network Security Laboratory”, IEEE Transactions on Education, 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” Proceedings of the American Society for Engineering Education (ASEE), Annual Conference of Composition and Exhibition, Austin, TX, June 2009

[14] Hill, J. M., Carver, C. A., Humphries, J. W., and Pooch, U. W. 2001. Using an isolated network laboratory to teach advanced networks and security. In *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education* (Charlotte, North Carolina, United States). SIGCSE '01. ACM, New York, NY, 36-40.

[15] Yuan, X., Vega, P., Xu, J., Yu, H., and Li, Y. 2007. Using packet sniffer simulator in the class: experience and evaluation. In *Proceedings of the 45th Annual Southeast Regional Conference* (Winston-Salem, North Carolina, March 23 - 24, 2007). ACM-SE 45. ACM, New York, NY, 116-121.

[16] Riccioni, A., Denti, E., and Laschi, R. 2008. An experimental environment for teaching Java security. In *Proceedings of the 6th international Symposium on Principles*

and Practice of Programming in Java (Modena, Italy, September 09 - 11, 2008). PPPJ '08, vol. 347. ACM, New York, NY, 13-22.

[17] Yang, T. A., Yue, K., Liaw, M., Collins, G., Venkatraman, J. T., Achar, S., Sadasivam, K., and Chen, P. 2004. Design of a distributed computer security lab. *J. Comput. Small Coll.* 20, 1 (Oct. 2004), 332-346.

[18] Aman, J. R. 2006. Black Hat/White Hat: an aggressive approach to the graduate computer security course. *J. Comput. Small Coll.* 22, 2 (Dec. 2006), 52-58.

[19] Du, W. and Wang, R. 2008. SEED: A Suite of Instructional Laboratories for Computer Security Education. *J. Educ. Resour. Comput.* 8, 1 (Mar. 2008), 1-24.

Biographies

Christopher P. Lee (chrislee@gatech.edu) obtained his Ph.D. in the School of Electrical and Computer Engineering at Georgia Institute of Technology in 2008 as a member of the Communications Systems Center. He also received both his Bachelor of Science and Masters in Electrical and Computer Engineering at Georgia Institute of Technology. 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 (selcuk@gatech.edu) is a Ph.D. candidate in the School of Electrical and Computer Engineering at Georgia Institute of Technology , Atlanta, GA as a member of the Communications Systems Center. He received his B.Sc. in Computer Engineering from the Turkish Naval Academy and M.Sc. degrees in Electrical and

Computer Engineering from Carnegie Mellon University in PA, USA, in 1997 and 2002, respectively. He received “2007 Outstanding ECE Graduate Teaching Assistant Award” from the School of ECE at Georgia Institute of Technology. He is a student member of IEEE, ACM, and ASEE.

Kevin Fairbanks (Kevin.Fairbanks@gatech.edu) has a Bachelor of Science in Electrical Engineering with a Computer Concentration from Tennessee State University where he graduated Summa Cum Laude in 2005. Since that time, he has obtained a Master of Science in Electrical and Computer Engineering from the Georgia Institute of Technology. Currently Kevin is a PhD Candidate at Georgia Tech. He is a graduate researcher in the Network Security and Architecture Lab where his advisor is Dr. Henry Owen. Mr. Fairbanks expects to complete the requirements for his degree by the Spring 2010 semester. His research interests include network security and digital forensics.

John A. Copeland (jcopeland@ece.gatech.edu) holds the John H. Weitnauer, Jr., Chair as a professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, 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.

Prior to joining Georgia Tech in 1993, Dr. Copeland was Vice President, Technology at Hayes Microcomputer Products (1985-1993), and Vice President, Engineering Technology at Sangamo Weston, Inc. (1982-1985) and served at Bell Labs (1965-1982).

He began his career at Bell Labs conducting research on semi-conductor microwave and millimeter-wave devices. NLater, 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. At Sangamo Weston he was responsible for R&D groups at ten divisions. At Hayes was responsible for the development of modems with data compression and error control, and for Hayes' representation on CCITT and ANSI standards committees. In 2000 he invented the StealthWatch system for network security monitoring, and founded LANcope, Inc. which today has deployed StealthWatch on over 100 corporate, government, and defense networks.

Dr. Copeland received B.S., M.S. and Ph.D. degrees in physics from the Georgia Institute of Technology. He has been awarded 43 patents and has published over 50 technical articles. In 1970 he was awarded IEEE's Morris N. Liebmann Award for his work on gallium arsenide microwave devices. He is a Fellow of the IEEE and has served that organization as the Editor of the IEEE Transactions on Electron Devices. He served on the Board of Trustees for the Georgia Tech Research Corporation (1983-1993), and Director of the Georgia Center for Advanced Telecommunications Technology (1993-1996).