Network Security: Focus on Security, Skills, and Stability

Susan J Lincke
University of Wisconsin-Parkside
Kenosha, WI, susan@lincke.org

Andrew Holland
University of Wisconsin-Parkside
Kenosha, WI, syclops@ehollands.net

Abstract - **Computer network security is a new and fast moving technology in the field of computer science. As such, the teaching of security is still a moving target. Security courses originally focused on mathematical and algorithmic aspects such as encryption and hashing techniques. However, as crackers hacked away at networks and systems, courses arose that emphasized the latest attacks. These attacks soon became out-of-date with security software responses. As security technology continues to mature, there is an emerging set of security techniques and skills. Network security skills emphasize business practices, legal foundations, attack recognition, security architecture, and network optimization. These skills tend to stabilize network security course(s). This paper summarizes skills relating to network security, and discusses active-learning exercises that assist students in learning these important skills.**

Index Terms - Security, Audit, Service Learning.

INTRODUCTION

Computer and network security is a new and fast moving technology and as such, is still being defined. When considering the desired learning outcomes of such a course, one could argue that a network security analyst must be capable of analyzing security from the business perspective in order to adhere to recent security legislation, and from the technical perspective in order to understand and select the most appropriate security solution. The analyst must be able to use security tools – but also apply the results to his or her organization. The analyst must be able to configure routers, firewalls, and an intrusion detection system (IDS) – but in an efficient and effective way. This partial list of skills improves security course stability and effectiveness. This paper investigates an outcome-based network security course emphasizing skill-development.

Network security originally focused on algorithmic aspects such as encryption and hashing techniques. While these concepts rarely change, these skills alone are insufficient to protect computer networks. As crackers hacked away at networks and systems, courses arose that emphasized the latest attacks. Currently, many educators believe that to train people to secure networks, they must also learn to think like a cracker [1-2]. However, Logan and Clarkson argue that teaching attack techniques is dangerous because it may lead to criminal behavior, takes course time away from important security techniques, and may fail debates during the Security course approval process [3]. However, additional reasons (learned from experience) is that 1) countering hacks results in lectures can become facts-based (e.g. identifying Microsoft vulnerable ports) instead of skills-based, leading to boring lectures; and 2) hacks become nearly irrelevant as soon as security software is enhanced to counteract the hack. Thus, an emphasis on hacking techniques can result in continual changes in the course material that often becomes out-of-date with the next minor/major OS or other software release.

An example is the Microsoft Windows null user, who used to have access to information about users, password restrictions, share availability, etc. The null user did not require a login ID or password [4]. Since Microsoft has fixed this vulnerability, this hacking lab is now outdated (assuming patches are applied or a new OS version is in use).

A second example is that traditional stages of hacking include Reconnaissance, Scanning, Gaining Access, and Exploit [2, 4]. With modern security techniques, including fortified border routers, firewalls and IDSs, the Scanning and Enumeration stages are mostly ineffective on a secured network. Instead, in 2006, virus/worms (usually emailed) were most frequently encountered and caused the largest financial losses of any attack type [5]. Optionally, direct web hacks, which bypass the early stages of hacking, may now offer a speedier, successful attack.

As security technology continues to mature, there is emerging a sophisticated set of security techniques and skills. Although students enjoy and learn from the labs, they have problems applying audit tool results to real-world problems within the required community-based project. Thus, security lectures and labs must help students to think or apply material, instead of memorizing fact after fact. A detailed statement of intended learning outcomes is necessary to define specific skills that enable students to be more effective in applying security techniques to real-world problems.

An undergraduate network security course is preferably widely applicable to any company. In today's world, companies expect employees to use existing packages for efficiency. Emphasizing security tools moves the course from a Computer Science (CS) focus into an Information Systems or Information Technology focus. However, it enables a broader set of security concepts and skills to be taught in a shorter time. Programmers can benefit when selecting and integrating security packages into application software;

1-4244-1084-3/07/$25.00 ©2007 IEEE

programming systems, networking, or security software; managing their own personal computers; or contributing to security discussions at their company.

The University of Wisconsin-Parkside (UWP) has recently implemented a Network Security certificate program, which awards a certificate to students taking three courses, selecting from Network Security, Web Security, a computer networks course, and a planned course on Information Systems Security. The Network Security certificate is offered to CS and Management Information Systems undergraduates and Computer Information Systems graduate students. This paper emphasizes the Network Security course, which includes the security topics: routers, firewalls, intrusion detection/prevention systems, encryption, and virtual private networks (VPN). As a course that integrates concepts of auditing, it also includes a high-level overview of how to manage security, including a section on computer security law, hacking, and incident response. The course project is a community-based learning audit project. Some described topics will/do overflow into the Web Security and Information Systems Security course.

In order to train a network security analyst, this course is result-oriented and emphasizes skill development. Analysts must be able to create a security plan, configure network security equipment, audit networks, and recognize and respond to attacks. This paper outlines these skills and concepts, and then briefly describes a number of successful active-learning exercises and labs to emphasize skill development. Section two outlines concepts and skills required by the program. Three sections describe network, hacking, and information systems labs. The last sections include lessons learned and a conclusion.

A FOCUS ON SECURITY

The Network Security program emphasizes training students to secure a network. The following background information in security helps in making correct decisions. Some areas are *concept*-oriented, but can benefit from demonstrations and exercises:

- Attack Recognition: Recognize common attacks, such as spoofing, man-in-the-middle, (distributed) denial of service, buffer overflow, etc.
- Encryption techniques: Understand techniques to ensure confidentiality, authenticity, integrity, and non-repudiation of data transfer. These must be understood at a protocol and at least partially at a mathematics or algorithmic level, in order to select and implement the algorithm matching the organization's needs.
- Network Security Architecture: Configure a network with security appliances and software, such as placement of firewalls, Intrusion Detection Systems, and log management.

To secure a network, certain *skills* must also be practiced:

- Protocol analysis: Recognize normal from abnormal protocol sequences, using sniffers. Protocols minimally include: IP, ARP, ICMP, TCP, UDP, HTTP, and encryption protocols: SSH, SSL, IPsec.
- Access Control Lists (ACLs): Configure and audit routers and firewalls to filter packets accurately and efficiently, by dropping, passing, or protecting (via VPN) packets based upon their IP and/or port addresses, and state.
- Intrusion Detection/Prevention Systems (IDS/IPS): Set and test rules to recognize and report attacks in a timely manner.
- Vulnerability Testing: Test all nodes (routers, servers, clients) to determine active applications, via scanning or other vulnerability test tools – and interpret results.
- Application Software Protection: Program and test secure software to avoid backdoor entry via SQL injection, buffer overflow, etc.
- Incident response: Respond to an attack by escalating attention, collecting evidence, and performing computer forensics.

The last three skills incorporate computer systems security, since they are required to counteract internet hacking. Additional skills that relate to computer system security are beyond the scope of this paper.

Network security applies business decisions in a technical manner. Business requirements drive security implementations. Business-related skills include:

- Security Evaluation: Use risk analysis to determine what should be protected and at what cost.
- Security Planning: Prepare a security plan, including security policies and procedures.
- Audit: Prepare an Audit Plan and Report.
- Legal response: Understanding and interpreting the law regarding responding to computer/network attacks, corporate responsibility (e.g., Sarbanes-Oxley), and computer forensics.

Security skills require extensive technical knowledge of internal operation, in order to recognize normal versus abnormal sequences. Additional general skill requirements include:

- Continuous Learning: Research in-depth information by oneself. (A limitation on course time and the evolving nature of computer science, networking, and security limits course expectations.)
- Writing and Communication.

COURSE MANAGEMENT

Active-learning exercises and labs are useful because the student becomes a participant in learning and develops skills. Lectures provide an introduction to the topic, providing breadth, background information, and help in interpreting labs. Labs allow students to practice with professional tools in order to obtain hands-on experience with each topic covered in the course. Labs are self-paced tutorials, where students work in pairs. Pairs are advantageous, because students with a networking background can be paired with students with an operating systems background. Students follow the directions of a lab handout and then answer questions on what they observe. They may be asked to

analyze how the tool can be used by both a cracker and a system administrator. Most labs are completed within one to two hours of this three-credit-hour per week course. With a small class, the instructor can assist and discuss lab results with pairs to ensure comprehension. For more complex labs, a review may occur during the next class period. References for most tools include [2, 8-9].

The homework project assignments for the course involve community-based learning, where students work with customers to audit a component of their network and complete a security product evaluation. The labs help to prepare students to successfully complete these audit projects. Graduate students have interviewed organizations related to their information systems security practices, which helps the students, the organization, and the instructor learn more about business security practices.

Students are mainly graded on the homework project assignments and exams. The audit plan and report are graded on format, completeness, and results interpretation. A security product evaluation is graded on format and technical analysis. A class participation grade, worth 10% of their total grade, requires students to perform the labs. Students can make up missed labs in off hours with advanced notice.

The course focus is on Microsoft, the most widely available and thus commonly hacked system. It is assumed that the concepts learned can also be applied to UNIX and other systems, although details may vary. Below, some of the most useful labs are described.

NETWORK SECURITY

Network labs may be expensive to work with since they rely on equipment being available. Our lab has one router per four students and this has proven adequate. Many important software tools are free, including sniffers, nmap, and Snort. The active-learning labs require that each group of students have simultaneous access to the relevant set of tools.

Sniffing Tools. Security analysts must be able to recognize attacks and write ACLs and rules for routers, firewalls, IDSs, and proxies – which means analysts must be able to understand and recognize protocol sequences. In the first networking lab, students are introduced to windump (tcpdump on UNIX) and ethereal sniffers (at www.tcpdump.org, http:\\windump.polito.it www.ethereal.com) [2, 8-9]. Sniffers enable a network analyst to view packets being transmitted over the network.

Specific applications can be started in order to observe TCP, UDP, ICMP and ARP in action, using telnet, web pages, ping, ssh, and arp –a, hacking tools, etc. These sniffing tools are used in scanning and other labs to observe audit or hacking tools' behavior. Since IP fragmentation is a source of attacks through firewalls and routers, the lab includes recognizing IP fragments created using ping.

Scanning Tools. Any open application on any machine can introduce vulnerabilities in security. Nmap (at www.insecure.org) is a tool that can scan a network and look for open applications [2, 8-9]. Therefore, it is useful to open one or more applications (like telnet) as part of the lab, so that students learn how to close unnecessary applications, when necessary. It is interesting to do this command with both the PC's firewall on and off to see how the firewall responds. Since any single tool can provide false positives or false negatives, it is recommended to run a couple of tools and compare results. Other useful tools include Superscan (www.foundstone.com) and Nessus/NeWT (www.nessus.org) [8-9].

While running a scanning tool is relatively simple, interpreting results is a challenge. Open port numbers listed by the tool need to be interpreted as to their necessity and vulnerability. Students should look up information about each open port or refer to lecture notes. While lab time is often short to complete such an analysis, it is emphasized as part of the audit homework project.

Finally, this lab is further enhanced by asking students to identify other capabilities of the tools and how hackers can use these same tools to break into systems.

Router Configurations. Writing ACLs is a skill, because it is error prone and can result in faulty filtering [7]. ACLs written inefficiently reduce router/firewall throughput. A lecture on routers reviews ACL formation, syntax, efficiency, and conflicts [7]. Cisco is the most commonly implemented router, and thus is the focus in the UWP security labs.

In the first year's router lab, students configured lab routers. In subsequent labs, students use nmap and ethereal to audit the configuration of the routers, by learning which TCP/UDP ports and ICMP messages the router passes or blocks in either direction. Students are given a set of policy objectives. They review the router configuration to determine what changes must be made to achieve the policy objectives and improve ACL efficiency. Alternative planned labs require students to program and test a router configuration that is both complete and efficient. Labs can also be enhanced to implement and test VPN options.

Snort NIDS. Network Security Analysts must be able to write rules for a network IDS (NIDS). Working with Snort allows students to observe a NIDS architecture, its configuration, and its programming. Snort (www.snort.org) is an example of a free NIDS that can be loaded onto each lab workstation [2, 9]. A useful lab is for students to interpret, write, and trip IDS rules. Snort can be run in sniffer or IDS mode. It is possible to create rules by simply adding a rule to a rule file. Alerts can be logged both to \Snort\log and the Windows event log. The format and an example rule put into the (e.g.) \Snort\rules\telnet.rules file is:

> <cmd> <protocol> <sourceIP> <sourcePort> -> <destIP> <destPort> (msg:"Alarm message text"; content:"String you want to monitor"; nocase;)
>
> alert tcp any any -> any any (msg:"Accessed the password file!"; content:"/etc/shadow"; nocase;)

If any session accesses the UNIX password file, this rule will trip and a log will be generated.

Encryption. Selecting an encryption technique and recognizing its advantages and risks can only be accomplished by understanding basic encryption techniques.

Encryption can be taught in a fun way that avoids complex mathematics while teaching basic concepts. (In the first year, this exercise was listed as the most interesting from multiple students.) Pfleeger's text [11] is an excellent and understandable reference to develop exercises from. Three example exercises follow.

A cryptogram is an example Substitution Cipher, where each letter in a short paragraph is replaced with another letter in a consistent way. Students can easily translate these when provided with the encrypted version of "Login" and "Password".

Figure 1 shows an example of a Columnar Transposition Cipher. In this cipher, data is output by column instead of by row, and the key provides the row size and column order (alphabetical order within key) [11]. In this exercise students decipher ciphertext given a key.

Key: D E C R Y P T

In: T h e W e a p
o n s A r e I
n T h e M i l
k b o x a b c

Out: eshotonkhntbaeibwaex…

FIGURE 1
COLUMNAR TRANSPOSITION CIPHER

A third example is a Block Cipher, which is the foundation for DES, AES, and other secret key ciphers [11]. Figure 2 demonstrates a small and simple cipher that uses a block size of 8 bits and two substitution (S-Box) and one transposition stage. Bits are exclusive ORed to provide the indicated ciphertext. In Chaining Mode, the output of one block becomes the key in the top S-box of the Block Cipher for the next block of input.

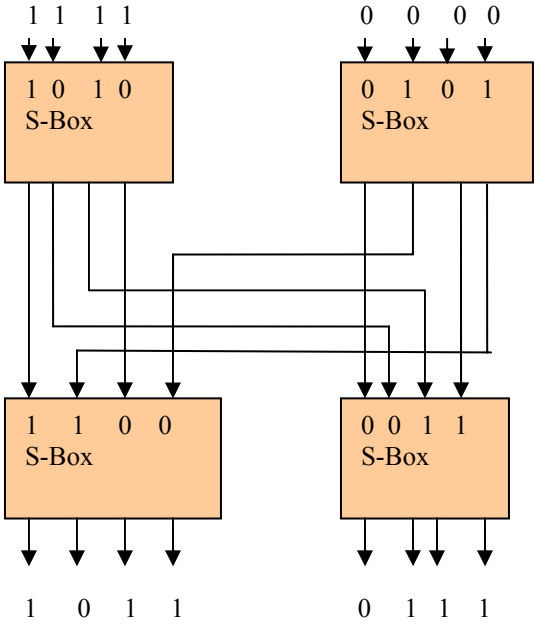

FIGURE 2
BLOCK CIPHER

Countering web hacking requires audit and programming skills to secure web services. Testing for SQL injection, and programming to defend against it is only one of many web and application-based attacks. These skills are emphasized in our Web Security course.

HACKING DEFENSE

System security labs (such as for servers, PCs, etc.) arose out of a three-week series on hacking. Lectures emphasize the four hacking steps (Reconnaissance, Scanning, Gaining access, and Exploit) and an introduction to Microsoft security [2, 4]. Community-based auditing projects have shown that server audits are popular. The challenge of an audit project is to analyze results of a vulnerability test, including checking for false positives and false negatives, and identifying which open ports are necessary. This requires a technical understanding of the results, which requires continual learning skills.

Vulnerability Testing. Vulnerability testing is important as an audit exercise. Microsoft provides an automated tool that self-audits a system, called Microsoft Baseline Security Analyzer (MBSA) [8]. This tool performs basic security checks on Windows OS and applications such as Internet Explorer, Internet Information Services (IIS), Microsoft Office and SQL Server. It also checks for outstanding hotfixes and performs minimal password vulnerability checks. Winfingerprint (of www.sourceforge.net) [9] is another useful automated tool that looks at the system from the inside in order to self-audit. Students need to research one or more detected vulnerabilities to determine if it is critical, and how the vulnerability should be addressed. More extensive analysis is required as part of the audit homework.

A manual audit can be performed with the Dumpsec tool (by www.somarsoft.com) [8]. The tool has a Report drop-down menu that can display information on password settings and privileges for groups and individuals. This tool can be used to audit what specific users can and cannot do.

Baseline Configurations. It is very difficult to recognize that a system has been hacked if a user is not familiar with how a Microsoft system runs internally. Therefore, running a baseline configuration on a known clean system is useful in order to have a basis of comparison for when a system may be compromised. Basic system information can be obtained through the commands: ver, winver, psinfo –h, psinfo –s, and (on Windows XP) System Tools-> System Information. A baseline of basic processes and drivers can be created by saving output from the following commands to a separate drive: psservice, sc query commands. Additional information that can be saved includes information on users (net user), ports (fport, netstat –o), dynamic linked libraries (listDLLs), and processes (tasklist). The file compare utility (fc /N) can then compare a known good baseline with a current baseline. Some tools are available as part of the \Support directory of the Windows installation CD, from www.sysinternals.com (PSTools), and www.foundstone.com.

Incident Response. When a system has been hacked, an autopsy of the system is performed to learn who entered, what they did, and to save information for legal purposes. It is important to immediately obtain the list of current processes, active network connections, open files, logged in users (and the entire system memory and finally the main disk if possible). Commands used to generate the information, including PsTools, listDLLs, and fport ([2, 9]), should be sandwiched between date and time stamps. It is then possible to compare a configuration against a baseline in order to detect differences and uncover a hack. Incident response tools should be executed off of a non-writeable CD, memory stick, or floppy: the host machine cannot be trusted since commands may have been overlaid to hide a hack. A more extensive list of useful utilities include: pslist, psservice, driverquery, listdlls, psloggedon, nbtstat, net session, net use, psFile, fport, netstat –an, netstat –r, arp –a, ipconfig /all, psloglist, sfind hfind, and hunt.

The goal of this lab is to have students find an active networking applications' directory path and command name, find the connection characteristics including protocol and IP addresses/ports, and name the networking dynamic linked libraries (DLLs) used [2]. The listDLLs command provides all except the protocol information. The lab has students open a network neighborhood connection, another active TCP connection (such as SSH), and a hacking tool during the lab. These help students to recognize normal and abnormal applications (assuming they are not hidden).

BUSINESS SECURITY ANALYSIS

The business aspects of security require that organizations are skilled in security planning, risk analysis, audit, and legal response. Because the Network Security course focuses on audit, lectures introduce all topics but active learning labs emphasize audit and legal response.

Law. Security analysts must be able to understand the basics of the law in order to successfully prosecute security cases. (Since student projects involve auditing community organizations, lectures also emphasize how to avoid prosecution by obtaining customer signatures on a detailed audit plan). Following an introductory lecture, small group discussions consider cases by trying to understand and apply the law.

Business security compliance is required by law for publicly-traded companies (Sarbanes-Oxley), federal agencies (FISMA), and the health care industry (HIPAA) [6]. Best-in-class references can be divided into low-level technical recommendations, such as those provided by the Center for Information Security (www.cisecurity.org), and high-level recommendations, such as COBIT (www.isaca.org/cobit.htm). COBIT defines business practices that are useful in achieving compliance with Sarbanes-Oxley. A lecture introduces COBIT, including an overview of its maturity model and areas of compliance. In an active learning lab, students learn to reference these recommendations to answer questions about specific best practices.

Audit. Preparing an audit plan and audit report is an important skill since it is necessary to comply with legislation such as Sarbanes-Oxley. Security auditing is used by over 80% of organizations as reported by the 2006 CSI/FBI Computer Crime and Security Survey [5]. Following a lecture on auditing, students follow an audit plan in a lab. They perform tests to validate that logs are created for specific actions and complete an audit report worksheet. The worksheet requires that they look up best-in-class standards, reinforcing concepts from the previous active-learning lab.

A course on information systems security could extend these labs to work with security planning, risk, and computer forensics.

LESSONS LEARNED

Students like the labs overall. Student feedback demonstrates that most students believe that the "labs were very useful to learn the material". The usefulness of the labs rated an average of 4.58 and a median of 5.0, where 1= Very False, 2=False, 3=Neutral, 4=True, 5=Very True. The rating of "The labs were clear and understandable" got a slightly lower score the first year, but was expected to rise as minor problems encountered were corrected.

Advanced students with long-time network administrator responsibility find some of these labs simplistic. Advanced students are given the freedom during the lab time to: 1) review research papers on security; 2) read and test using advanced documentation available in the lab; or 3) attempt to break into the security lab and to fix such break-ins. In all cases, the instructor monitors and approves the learning goals of their individualized lab.

CONCLUSION

The security field is a new, fast-moving career. A focus on security stabilizes course material, reduces worry about student hacking, and helps to provide students the skills necessary to become security analysts. This paper defines a set of skills required by network security analysts, and describes a set of useful labs that help students become adept at securing a network. The active-learning exercises help to reinforce the lecture material, emphasize the application of security tools, and move students from 'exposure' to 'competency' in performing security tasks required in industry.

ACKNOWLEDGMENT

This work is partially supported by NSF Grant 0313712, Aug 2003 to the University of Wisconsin consortium. The authors wish to thank NSF and Tim Fossum, a key player in obtaining the grant.

REFERENCES

[1] P. Mateti, "A Laboratory-Based Course on Internet Security", *Proc. Of 34th SIGCSE Technical Symp. on Computer Science Education*, ACM, 2003, 252-256.

[2] *Computer Network Defense Course (CNDC),* Army Reserve Readiness Training Center, Fort McCoy WI, http://arrtc.mccoy.army.mil, Jan. 2004.

[3] P. Y. Logan and A. Clarkson, "Teaching Students to Hack: Curriculum Issues in Information Security", *Proc. Of 36th SIGCSE Technical Symp. on Computer Science Education*, ACM, 2005, 157-161.

[4] S. McClure, J. Scambray, and G. Kurtz, *Hacking Exposed 5th Ed.*, McGraw Hill, 2005.

[5] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, *2006 CSI/FBI Computer Crime and Security Survey*, Computer Security Institute, GoCSI.com.

[6] U. A. Pabrai, *The Art of Information Security*, www.ecfirst.com, 2005.

[7] H. Hamed and E. Al-Shaer, "Taxonomy of Conflicts in Network Security Policies", *IEEE Communications Magazine*, 44, 3, (March 2006) pp. 134-141.

[8] *Auditing Networks, Perimeters, and Systems Hands-On Workbook*, Audit 507 – Auditing Networks, Perimeters & Systems Course, SANS Institute, www.sans.org, 2005.

[9] M. Shema and B. C. Johnson, *Anti-Hacker Toolkit*, 2nd Ed., McGraw Hill, 2004.

[10] *Advanced Systems Audit: Windows NT/2000*, Audit 507 – Auditing Networks, Perimeters & Systems Course, SANS Institute, www.sans.org, 2005.

[11] C. P. Pfleeger and S. L. Pfleeger*, Security in Computing, 3rd Ed.,* Pearson Education, Prentice Hall, 2003.