

A Laboratory-Based Course on Internet Security

Prabhaker Mateti
Department of Computer Science and Engineering
Wright State University
Dayton, Ohio 45435-0001
pmateti@cs.wright.edu

Abstract

We developed a laboratory-based course on Internet Security. The course is aimed at the senior undergraduate. This paper discusses the course and explains how others can set up their own labs to teach this course. All the laboratory work is conducted in a laboratory of PCs running Linux. We developed lecture notes for the course, and a web site to widely disseminate these materials.

Categories & Subject Descriptors

C.2.0 *Computer-Communication Networks*: General - Security and Protection; D.4.6 *Operating Systems*: Security and Protection; K.6.5 *Management of Computing and Information Systems*: Security and Protection

General Terms

Experimentation, Security

Keywords:

Internet security, Network security, TCP/IP exploits, Firewalls, Buffer overflow, Ethics.

1 Introduction

Computer networking has become so ubiquitous that every computer system of any importance is networked to others. The Internet Domain Survey (see

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCSE'03, February 19-23, 2003, Reno, Nevada, USA.
Copyright 2003 ACM 1-58113-648-X/03/0002 \$5.00.

www.nw.com) reports that there are 162,128,493 hosts in the DNS, as of July 2002. The older security problems of insider breaches of security are now compounded by attacks carried out remotely through the network. This is fueling the growth of what are known as “network firewalls” and increased deployment of security tools.

In our experience, graduates from typical degree programs in Computer Science and Engineering are familiar with stories of security breaches as reported in the media, but do not have a technical grasp of the security issues. There is an urgent need for training the students in computer and network security.

We developed a laboratory-based course on Internet Security. The course is a senior undergraduate course. At our university, it carries four credit hours and is scheduled over a 10-week academic term. At semester-based institutions, the sophistication and number of the laboratory experiments can be increased to become a 15-week semester long course. We developed lectures, and laboratory experiments that are to be conducted on isolated LANs. All the laboratory work is expected to be conducted in a laboratory of PC-clones that are capable of running Linux. The laboratory setup is well documented for use by other institutions planning to offer courses on Internet Security. We include lectures on social responsibility and ethics. A companion web site [3] exists to widely disseminate all these materials.

Section 2 describes our goals and objectives, and also includes a survey of related efforts. Section 3 is a discussion of the course content. Section 4 is a description of our laboratory setup. Section 5 is a summary of our experience in teaching this course at our institution. Section 6 concludes this paper.

2 Goals, Assumptions and Related Work

We have four goals: (1) Teach security improvement techniques. (2) Explain how exploitable errors have been made in the development of software. (3) Raise the level of ethics awareness. (4) Bring attention to legal issues.

2.1 Assumptions

We are aware of efforts in integrating security concepts in various required courses ranging from CS1 and up (see e.g., [4] and [5]). While we believe in the effectiveness of this approach, we found it difficult to implement in our department. It was much easier to propose and implement an elective course. Thus, we were strongly driven by the following question. What should be the scope of a course on security, if that is the only course the student may take?

“I hear and I think. I see and I remember. I do and I know.” – Confucius. We felt it essential that the student experience the attacks present in security breaches and the defenses being erected. The course we designed is thus driven by the needs of lab experiments. In order to keep the time spent under control, most of the setup of the “apparatus” is already done, and the student does a final check and adjustments.

Our course includes a discussion of ethical and legal issues. During the first day of classes, we require the student to subscribe to a statement of ethics. We devote perhaps 80% of the time to the first two goals, and the remaining time for the last two. We had a natural tendency to concentrate only on the first two, being a faculty of a computer science and engineering department, but in our estimation undergraduate degree programs are already so crowded with required courses that a student is then unlikely to take a separate course on ethics and legal issues.

2.2 Survey of Related Efforts

Web Sites “There is an oceanic amount of material on network security available over the Internet.” – A Web Page.

The available material falls into two categories: (1) articles, lecture notes, FAQs (frequently-asked-questions with answers) and (2) source code for both attack tools and protection tools.

Text Books There are easily over 250 books published on the subject of network security. At the risk of being politically incorrect, we estimate that no more than a dozen of these books can be used as academic text books. We readily admit that we cannot succinctly define what distinguishes a good text book. The books that appeal to us are: [1], [8], and [6].

Courses There are excellent compilations (e.g., theory.lcs.mit.edu/~rivest/crypto-security.html) of security-related courses offered by academic institutions. There is also an equally impressive variety of courses offered by various “training and consulting” firms. The need for developing academic

courses on security is articulated widely; e.g., there are now thirty six Centers of Academic Excellence in Information Assurance Education supported by the NSA (www.nsa.gov/isso/).

The academic courses are typically graduate level courses. We have spent considerable time on the web scanning the descriptions of courses, and all available material from each. Our course differs in content, level, and emphasis on laboratory experience.

The content of the courses we examined is also typically dominated by mathematical treatment of cryptography. Our course includes cryptography material, but is dominated by network security topics.

3 Course Content

Our web site [3] is oriented towards a hands-on academic course on network security, and contains (1) Detailed descriptions of lectures and laboratory experiments suitable for a 10-to-15 week course, (2) Articles on ethical and legal issues, (3) Detailed description of the laboratory setup based on PC hardware and Linux, (4) Pointers to archives of source code of both attack, and defense tools, (5) Samples of lab reports by past students, and (6) Past midterm and final examinations.

3.1 Topics of Lectures and Experiments

This subsection summarizes the topics covered in the course, along with a rationale for their choice.

“Internet Security”? “The network is the computer.” – A well-known company. On the other hand, we must not view all computer security issues as network security issues. Modern operating systems provide abstractions through which a file system may be made to reside entirely in RAM, and what is perceived as keyboard input strokes are actually arriving on the network from a source thousands of miles away.

The consensus (see, e.g., [7]) is that Internet Security topics include authentication, firewalls, spoofing, smurfing, routing tricks, and privacy of data en route. It also includes buffer overruns etc. because they enable attacks on network daemons even though these are software development errors.

Depth v. Breadth We felt it essential that the student be able to understand on a technical level the security breaches occurring *currently*. This determined the breadth of the topics. The need to conduct the experiments with a grasp of the exploit and the defense techniques determined the depth. The wider choice of topics does cause a lack of conceptual coherency. But, it gives the student the knowledge needed

to securely configure and monitor a personal machine, and also serves as a capstone course that ties together techniques learned in many required courses.

System Administration A course in operating systems, and a course in computer networks are necessary prerequisites. Even so, many students lack practical experience in setting up a networked computer system. They are often unfamiliar with how an operating system boots, how a system is assigned a network address, how user accounts work, and how a system logs interesting/suspicious events.

System administration work and knowledge is often not considered worthy of academic attention. We think this is unfortunate as it provides the much needed perspective for courses in OS.

Experience Serious Nuisance We hope to raise awareness of responsible behavior at the outset. The student may not have visited sites that are obnoxious, or experienced serious nuisance.

Well Known Security Breaches The value of history of any subject can hardly be debated. We must highlight both the most famous ([10], [2]) and the most recent incidents (visit www.incidents.org).

Trojan Horses, Viruses and Worms Viruses *were* a common enough experience, but virus scanners are now ubiquitous and increasingly the havoc that viruses can cause is to be shown in the laboratory to be believed. It is also educational to explain the ideas behind well-known scanners.

The media uses the term “viruses” to refer to all kinds of malware, but we can distinguish Trojan horses, viruses and worms. Today, these are network deliverable.

Privacy and Authentication of a User Poorly chosen passwords are still the norm despite the warnings from local system administrators. An effective way of changing this behavior is to show how easy such passwords are to crack.

While most students have become good web-surfers, many are unaware of the loss of privacy and security that can happen.

Design Weaknesses in TCP/IP Internet is currently based mostly on IPv4 (IP version 4) that was designed at a time when security threats were unknown. All data including various fields in the protocol headers are sent in the clear. Trade literature trumpets IPv6 as “the blueprint for 21st century e-commerce.” Among the many improvements IPv6 has the more importantly for us are support for authentication, data integrity, and

data confidentiality. A student should be exposed to the weaknesses inherent in TCP/IP and the improvements of IPv6.

Firewalls The non-specialist computer community uses the term “firewall” as being a network security system. *Packet filters* work at the IP level. *Circuit gateways* are packet filtering routers, but also maintain limited state. An *address translating firewall* hides the internal IP addresses by translating all addresses as they cross the firewall. *Stateful inspection* in a packet filter refers to the ability to remember certain details of the past traffic on the connection. *Proxy servers* generate a much finer level of control than a packet filter.

All of the terms above are now in use by the average computer literate person. It is important that CS students understand them at a technical level.

Cryptography Data encryption is at the core of authentication, privacy, and confidentiality. Students should be exposed to the practical uses and essential technical details of the *symmetric key* and *public key* cryptosystems.

Secure Configuration of Personal Machines Perhaps as many as 90% of the 162 million nodes connected to the Internet are personal machines, the rest being various servers. Perhaps 80% of these personal machines are running Windows and Linux with little supervision from system administrators.

Right out-of-the-box, these personal machines are security-wise improperly configured. Also, we can distinguish two more levels of security strength: fortification, and hardening.

Buffer Overflow and Other Bug Exploitation Networking requires large programs: layers in OS, and various clients and servers. The software development expertise today is such that all large programs, without exception, contain bugs. Further exacerbating is the fact that much of the software is based on original TCP/IP source code that was developed rather casually without much concern for rigorous correctness. Attackers have studied this source code, and experimented with this software. Carefully explaining the attack techniques is an eye opener to many students in how compilation techniques, computer architecture, and virtual memory segmentation interact.

Writing Bug-free and Secure Software Writing better, security-wise, software involves at the high-level code structure, least privilege, and narrow interfaces, and at the low-level, checking for buffer overruns, being ultra careful in writing `setuid` programs, untrusted

paths, race conditions, environment, etc. Of critical importance is a discussion of type-safety, assertions and invariants. Software engineering courses discuss some of these but without the security slant we would like.

Security Standards We think it is worthwhile to survey a few government originated standards such as the Orange Book.

Server Security Web browsers are common place now, and even though most organizations are forbidding personal servers we predict that a decentralization of Web-, file-sharing, and other servers is around the corner.

3.2 A List of Experiments

This section is a near-complete list of what we developed and readily available from our web site [3]. On average, the lab experiments take two hours each, not counting the time spent in writing the lab report. In a 10-week, 4-credit-hour course, perhaps eight of these experiments will be chosen by the instructor.

- (1) *Experience serious nuisance* by visiting a few selected sites (e.g., www.digicrime.com) and write a personal summary of their reaction to these sites.
- (2) Run, experience, and examine at least one each of *Viruses, Worms, and Trojans*.
- (3) How does a machine *boot* from powering up to login prompt? For an attacker, this is an excellent time to install Trojans.
- (4) *System Administration*. Use and understand various audit trail files on a Unix system. Understand the role of `/etc/passwd` and `/etc/shadow` files. Understand password aging mechanism. Check if user passwords are easily guess-able.
- (5) Use *password cracking tools*. Two possible projects are to enhance their speed, and to use idle computing power from other nodes.
- (6) Use *one-time passwords, and secure shell*.
- (7) Use “pretty good privacy” (PGP) (www.pgpi.org and www.openpgp.org). Use digital signatures. Visit www.anonymizer.com.
- (8) *Securely configure* a Linux PC according to “Unix Configuration Guidelines” of www.cert.org. *Securely configure* a Windows machine according to the guidelines in [9].
- (9) *Fortification* Search a white-hat security tools site such as packetstorm.decepticons.org, and select, install and experience tools, such as `tcpwrapper` that fortify a Linux installation.

(10) Apply security patches, such as `grsecurity`, at the source code level to the stock Linux kernel, and rebuild a *hardened* kernel.

(11) Setup a LAN of three machines, with one of them as a *router*. This experiment is a prerequisite to several others.

(12) Setup an isolated LAN. Install a *network sniffer* on one machine, and observe the contents, including passwords in clear text, of all packets.

(13) Setup an isolated LAN, and *hijack an on-going telnet session*.

(14) Learn how user, node and services are *authenticated* and *spoofed*.

(15) Setup an isolated LAN. Install a *DNS spoof* on one machine, and observe how traffic is misdirected.

(16) Download a *rootkit* and install. Check which of the detection tools is able to discover the presence of a rootkit.

(17) Install and discover *back doors* in well-known programs and OS.

(18) Study the source code of one or two *buffer overflow exploit* programs, and learn how to fix the bugs, and speculate on how the bugs could have been avoided.

(19) Select one of the *white-hat security tools* (such as SAINT, Nessus, Tripwire) available in open source, install and study the code.

(20) Set up a *packet filter* on Linux. Setup a *firewall* based on Linux or Windows.

(21) Scan an isolated LAN within the lab *probing for weaknesses* using, say, SAINT. Study its internals.

(22) Cause and observe *denial-of-service attacks*, such as the famous “SYN flood.”

(23) Observe how *design weaknesses of TCP* can be exploited by killing connections by RST, closing a connection by FIN, and connection hijacking.

(24) *Security audit* the local LAN and local computer systems, and prepare a report.

(25) Build IPv6-enabled Linux kernel, and related tools. Setup an isolated LAN of, say, three machines with one of them as a router.

4 Description of Laboratory Setup

4.1 PC Hardware and Operating Systems

Students and faculty have become used to high speed PCs, but if we could persuade them to be objective about the needs of the Security Lab, a decently equipped i486 PC will do. However, for reasons of main-

tenance, we suggest low end configurations of current models, with two network cards, from any well known PC systems vendor.

Linux distributions are now very stable. We currently use the Linux distribution by Mandrake.

4.2 Network Setup

Each PC in the Lab should have two or more network cards. With a quick disconnect/connect, they can be made to belong to different private 192.168.*.* LANs. The LAN of the Lab is isolated. When there is a need (e.g., in the software maintenance of the Security Lab), one of these PCs running Linux can be rebooted as the designated gateway to the departmental LAN.

4.3 Repeated Installation of OS Images

Various experiments that the students are expected to perform will destroy the set up of individual PCs. It will be frequently necessary to re-install and re-configure entire OS on these.

4.4 Miscellaneous

Some labs require an isolated LAN of at least three machines. Thus, in a lab facility with 30 machines only ten students can work simultaneously on such experiments.

The lab work is unscheduled, and the laboratory is accessible 24x7.

The offering of a course such as this is viewed as a labor-increasing item by the local sysadmins. They are also alarmed that the students are permitted to be super-users on these machines.

5 Summary of Teaching Experience

The course consumes a substantial amount of time both in keeping up-to-date the course content and the lab setup. The software tools used, the Linux kernel, and the distributions are being so constantly revised that the lab setup needs to be updated at least twice a year.

Course evaluations by students indicate that they value the course highly but find it heavy.

If a course in operating systems and a course in computer networks were not mandated as prerequisites, the students would have greater flexibility in scheduling the courses into their programs of study. We tried to drop these prerequisites on a finer level of granularity based on each lab experiment, but now conclude that it is not possible.

6 Conclusion

Computer networking has become ubiquitous. Security breaches have become common. Recent graduates are familiar with stories of security breaches, but do not have a technical grasp of the security issues. There is an urgent need for training the computer science and engineering undergraduate students in computer and network security.

We developed a laboratory-based course on computer network security. We developed a web site [3] to widely disseminate laboratory experiments, laboratory setup, packaged lectures and laboratory experiments suitable for a 10-or-15 week course, collected articles on ethical and legal issues.

Our students take the course as an elective in their senior year. They value the course highly but find it heavy.

This work was supported in part by NSF DUE-9951380.

References

- [1] Garfinkel, S., and Spafford, G. *Practical UNIX & Internet Security*, second ed. O'Reilly & Associates, Inc., 1996.
- [2] Howard, J. D. An analysis of security incidents on the internet, 1989 - 1995. Ph. D. Thesis, Carnegie Mellon University, April 1997.
- [3] Mateti, P. CEG429: Internet Security Course website. www.cs.wright.edu/~pmateti/InternetSecurity (2002).
- [4] NCISSE. *Sixth National Colloquium for Information Systems Security Education*. www.ncisse.org, 2002.
- [5] Oblitey, W., Paul Mullins, Wolfe, J., Fry, M., Winters, E., Calhoun, W., and Robert Montante. Integrating security concepts into existing computer courses. *SIGCSE 2002* (2002).
- [6] Rubin, A. D. *White-hat Security Arsenal*. Addison Wesley Publishing Company, Inc., 2001.
- [7] Shirey, R. Internet Security glossary. Internet Request for Comment RFC 2828, Internet Engineering Task Force, May 2000.
- [8] Stallings, W. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 1998.
- [9] Sutton, S. Windows NT security guidelines. www.trustedsystems.com (1998).
- [10] Zimmerman, C. Hack attacks revealed - a history of famous hacks. *SIGCSE 2002 Student Activities* (2002).