2006-1831: SECURITY EDUCATION

Tim Lin, California State Polytechnic University-Pomona

Saeed Monemi, California State Polytechnic University-Pomona

Security Education

Abstract:

Network security and computer security are usually hot topics whenever any intrusion incidents cause system crash and loss of work time in big corporations. In engineering colleges however security is usually a topic with least or incompatible attention.

The author has taught many upper division classes in college and also graduate course(s) and has been trying to imbue and enhance the courses with the security component. For example, C / C++ programming course was enhanced with basic ciphering like Caeasr cipher, Java programming was enhanced with Java security concepts, OS course was enhanced with the coverage of access control, buffer overflow, malicious software, etc., network course was enhanced with the coverage of insecure protocol and secure protocol etc. The author will discuss how security education can be enhanced in many college courses below.

1. Introduction

Whether caused by so many news worthy security incidents or not, security education has become a trend either as a major, a certificate program, a curriculum, or as modules to enhance the current curriculum nowadays.

Security education papers

It was worth noting that security education is noted and discussed in ASEE papers as well as by SEI of CMU (CMU has been holding some boot camp in summers on security education workshops of faculties).

In ASEE 2004, there were papers on building undergraduate security curriculum[1] or graduate certificate in information assurance[2].

In ASEE 2005, there was a paper on the security as a component of wireless communications program[3].

Carnegie Mellon University, famous for their software engineering and quality assurance standards in SEI, had recently established a Survivability and Information Assurance curriculum[4] of 3 courses, available for free download by educators: Principles of Survivability and Information Assurance, Information Assurance Networking Fundamentals, and Sustaining, Improving, and Building Survivable Functional Units.

Security education curriculum

The following is the certificate program on information and operating system security (IOSS) by Mt. San Antonia and the CIS (Computer Information System) department of Cal Poly Pomona at Los Angeles5:

New courses at Mt. SAC for Fall 2005:

- CISS 11: Practical Computer Security
- CISS 13: Principles of Information Systems Security
- CISS 15: Operating Systems Security
- New Certificate: Information and Operating Systems Security

- Finalized courses at Cal Poly Pomona
    - CIS 471: Internet Security (formerly CIS 499)
    - CIS 481: Computer Forensics
    - GBA 685: MSBA Project-Adv Computer Forensics

The college Mt. San Antonio also offers a network security certificate program with 3 courses:

CISS 21: Network Vulnerabilities and Countermeasures
CISS 23: Network Analysis and Network Intrusion Detection System
CISS 25: Network Security and Firewalls

Base on the curriculum of the other colleges and departments listed here (and also the curricula mentioned in the ASEE papers), a good information assurance curriculum (or major, focus, option) consists of the following six courses. Note the core courses on OS, Network, and Programming are also what the authors of reference 1 mentioned in the first paragraph of their ASEE 2004 paper on undergraduate curriculum in security.

Introduction to Security / Information Assurance
Operating System with Security Modules
Computer Network with Security Modules
Programming Courses with Security Modules
Application Course such as Computer Forensics
Information Security Issues like legal aspects, social aspects, policies, compliance etc.

Some courses like Computer Forensics and Information Security Issues can be offered as a team work with other departments or colleges. The current paper discusses some details of the core technical courses: Operating Systems, Computer Network, and Programming with the nitty-gritty details that the author has been instructing, developing, and enhancing with the security components in the last few years.

This paper is organized as follows: section 1 as an introduction to discuss the general security education curriculum, section 2 discusses the different stages of security proficiencies the instructor can teach the students, section 3 talks about the engineering courses that can have security education components, section 4 gives a detailed treatment of security materials in several engineering courses, and section concludes the security education with future efforts.

2. Stages of Security Proficiency for the Students

Though security education has aroused widespread interests and the general public are awakened after every publicized hacking, denial of services, or virus attacks, the security awareness and the education for the students is still below the way it deserves. Also, the perception of what, how, and how much should be covered in security education could also differ substantially.

Frankly speaking, when discussing security education, there are many different levels of how much security is educated. This is similar to the levels of comprehensions for a foreign language; some detailed definitions are needed to clarify.

3 stages of proficiency for the students of security

From the author's point of view, the security education for each aspect or area can be defined in 3 stages:

Stage 1: Awareness: know the terms, concepts, and impacts. Have the common sense and know the basic procedures to protect the computer from the general attacks. Able to perceive if some security of the computer is compromised.

Stage 2: Hands On Capabilities and Experience: able to use the tools, emulate attacks or defenses, develop software / devices for attack / defense.

Stage 3: Professionalism: know the different issues of security such as legal, social, economic, ethical, etc., achieve the technical level of professionals, pursue a career in this profession, conduct professionally (ethically, such as ethical hacking).

Most books or courses on security education cover stage 1: statement of knowledge which makes the students aware of the security issues. It is good for the students to achieve stage 2: have hands on experience to use tools or even project or development experience; or have experience in attacking or defending. Courses on computer forensics cover some of these aspects. It would be good that faculties of different schools can share the knowledge, the tools, and the experience of security education.

Symposium on policies, compliance, and careers in information assurance cover stage 3. But this would be a solid stage 3 only if the students have solid stage 1 (knowledge) and stage 2 (technical).

Because of these 3 stages, security modules development can be an on-going effort since modules developed in the past in stage 1 (or in stage 3 only) can be enhanced to include stage 2, stage 3 etc. Stage 2 is actually a very big stage that can inspire the students to work on assignments, do projects, or later on develop security products etc. Stage 3 however is a very important stage. Some professional programmers, kiddie scripters, or and crackers fell into the trap of knowing too well technically (can break into systems unauthorized) but knowing too little about stage 3 of the legal, social, and ethical impacts of their actions. The author of the disastrous CIH virus in 1998 admitted that he wrote that virus to test his programming capability and he did not realize that it went out of control after he gave his classmates copies of the program.

The difficulty of teaching stage 2 consists in how to develop and run some potentially dangerous code or dangerous procedures in a controlled manner: for example, running a program that consistently spawns new processes, generates new files indefinitely, or ping-ponging messages or emails among two computers can consume so much CPU time, using up memory or disk spaces if not monitored well. However, this can be handled by some loop control (finite number of iterations). What may be harder to control is the emulated DOS or DDOS attack in the lab that can render a computer useless faster than we think.

In section 4, the author will discuss modules developed before, currently developing, or to be developing in the core courses OS, Computer Network, and Programming, with the stages of the modules mentioned as appropriately.

Dilemma for the educators:

With this discussion, we as the educators can run into this dilemma: can you teach the students on how to break the security holes of a certain system or not?

Let me quote the famous author Andrew Tanenbaum from his classic book on Operating System[7]: " .. was hesitant to write this section (attack from outside the system) in so much detail, lest it give some people bad ideas. … , the genie is already out of the bottle. In addition it is hard for people to defend themselves against viruses if they do not know how they work".

A very simple code example as given by Tanenbaum is the one line program from his OS book, chapter 9 on security. According to Tanenbaum, this one line program can kill a Unix system since it keeps spawning new process until it consumes all the system resources (I have not tried personally myself this one yet. It is a good stage 2 assignment or project).

```
main () {while (1) fork () ; }
```

I have the same dilemma and also someway that I need to clarify to the system administrators or the other instructors teaching network security when I ask them "what is the procedure to log into a system programmatically?", "how many routers are in the network, are they password protected?".

The concern is not only "why do you need this information?" but also "are you sure your students will not use such information to work against us?".

Teach students how to monitor and also how to experiment attack in a controlled environment (not always easy to achieve). The famous CIH virus mentioned above was due to an out of control computer virus.

3. Engineering Courses (in the curriculum) With Security Modules

Several engineering courses are natural candidates of covering security as discussed so far: Operating system course with security topics like viruses, buffer overflow, access control; computer network courses with security topics like encryption, authentication, attack on certain protocols, and programming courses where programs on security can be given to let the students write compute programs doing security. These courses with enhanced security components will form in the future a security or information assurance curriculum.

In the ECE department (Electrical and Computer Engineering department) of Cal Poly Pomona (California State Polytechnic University, Pomona), they correspond to the course title and numbers as follows (I have taught each of theses courses at least 3 times, but with security components added these couple of years):

> Operating System with Embedded Applications (ECE426 and the lab ECE426L):
> > Operating System / Computer Security
>
> TCP / IP Internetworking (ECE433): Network Security
> Programming Courses: C (ECE114) / C++ (ECE256) / Java (ECE429)

I also instruct a graduate course ECE520 Network Security; but the course will not be discussed here. In the future, many other engineering courses can also be enhanced with a security education component, for example the software engineering course or object oriented design course.

4. Materials in Security Modules

In this section, the three types of course mentioned above will be discussed in more details, each of them with modules developed before (old ones), modules currently under development (evolving), and modules to be developed (new modules). These modules together will form the core of the future curriculum in information assurance.

4.1 Operating System Security

4.1.1 Modules Developed or Used Before:

There are 4 modules in this category: Security Basics module with materials from Tanenbaum, Encryption, Buffer Overrun, and File System Security (Generating Files Indefinitely) modules by the author.

Security Basics Module (Tanenbaum):

Andrew Tanenbaum has an excellent chapter on security of 80 + pages long in his OS book[7]. This covers concepts on viruses, authentication, malicious code etc. Most materials were covered in the stage 1 way, i.e. knowledge for the students. There were some however interesting stuff (stage 2) that the students can have hands on experience, for example the steganography example of hiding 5 Shakespears plays in a BMP file of zebra picture.

Encryption

The author covered the concepts of encryption including the simple symmetric Caesar cipher as an introduction and the more sophisticated RSA private key cipher. Concepts (stage 1) and code development (stage 2) were covered in the class. An ultimate stage 2 effort for RSA is to try the RSA challenge of factoring the huge RSA numbers of more than 100 digits.

Buffer Overrun

Buffer overrun (or buffer overflow) is a common problem found in C programming. Since C does not check for array out of bound, C code written can have an array a of size n defined but an element a [p] referred with p > n, causing unpredictable result at run time. There are many examples of buffer overrun in literature that can be used to explain.

The author had also used some properties of string functions in C (the strcpy function in particular), which when executed could overwrite the contents of some arrays.

Generating files indefinitely

It is easy to instruct the students how to write a program generating a file automatically (programmatically) using C, C++, Java etc. (everybody knows how to do that manually, but not every student learned or paid attention in their previous programming classes to write a program that will create a file when run).

The program was used in the OS lab to demonstrate how the file system works when files are created (some students enhanced that to create folders programmatically) and student asked to revise it to loop many times to create multiple files.

The security aspect of this lab is:

First the programming loop has to be a finite, controllable number such as looping 10 times, 50 times, 100 times with a file of small size.

Students were warned and asked to monitor the hard disk left (and memory used) when the file size is big and the number of iterations large (in the worst case, it could be an infinite while loop that keeps generating files forever).

In the worst case, the students were told that if the exe file they generated were attached as an email attachment to a friend, and the friend just clicks the attachment, then this is a kind of undocumented virus that can consume disk spaces in the place the user may not know (the behavior depends on the OS and services packs. Some OS may cause a security violation; still a good example for the students).

There is a related lab of deleting files automatically in the OS lab for file systems (and also experiments to change file attributes programmatically). These labs were used for the students to learn more about the file systems. But they are also good examples of how to create “viruses” in a small scale for the security education purpose.

4.1.2 Modules to be Developed / Enhanced (new modules)

Passwords Cracking:

Password concepts were well covered by Tanenbaum's book. It is mentioned in OS books and Network security books that password length has to do with how long it takes to break the password. It is a good module or some simple lab assignment for the students to find out how long it takes to break a password of a fixed length n (with 26 characters only in upper cases and lower cases, the amount of time will be proportional to 52^n, a good way to educate the students about strong and weak passwords).

Phishing

Phishing is a social engineering trick (or tool) to fool other people to willingly give their personal information (the ID theft problem). In the past, phishing was mentioned or covered in the class as PowerPoint (stage 1). It is worthwhile to put efforts on developing some phishing tool or phishing labs so that the students know how to develop that (stage 2). Definitely, we need to know that the students are educated and mature enough of doing this only ethically (stage 3).

Malicious Code:

Currently the malicious code module is taught covering the classification of malicious code: viruses, worms, Trojan horses etc (stage 1). Efforts will be put to either develop some malicious code (the buffer overrun example, the generating many files example are actually some kind of "virus" if they are files coming from outside) or some virus scanner code.

4.2 TCP / IP Internetworking (Network Security)

Traditional textbooks (R. Stevens or D. Comer), on computer network or TCP / IP do not put emphasis or have chapters or sections dedicated to network security. The most mentioned topic is IPSec. An excellent textbook on TCP / IP authored by B. Forouzan that I recently switched to contains one chapter (out of 28 chapters) on network security (and sporadically in the book covering the possible attacks on several important protocols like TCP). It is good treatment, but the subject probably deserves at least twice more coverage due to its importance. In the older days, people probably paid more attention trying to make networks working, and can not as so focus on the security as well.

There are many books on network security covering sophisticated math theory, standards, detail protocols or procedures, and some histories. There is an interesting book on network security form the hacker's perspective by Ankit Fadia[8] that covers the ins and outs of many ways of attacks.

4.2.1 Modules Developed Before:

IPSec

IPSec is traditionally covered to show how IP security can be done (it basically means IP is an insecure protocol) using concepts like Authentication Header, Encapsulating Security Payload in

Powerpoint files. There is in general no student involvement (in some of IPSec code for example).

WEP

WEP (Wired Equivalent Privacy), a part of IEEE 802.11 protocols, is a wireless protocol that people know well its vulnerability. The reason is due to that the initialization vectors are encrypted using the XOR (exclusive or) operation. Though XOR can scramble the bits well, it is easy to break since if C = A XOR B, then A = C XOR B from elementary Boolean algebra. As a consequence, WEP is subject to the so called known plaintext attack.

The coverage of WEP before was limited to the explanation of its vulnerability (stage 1). It can be enhanced as a demonstration in the class (or a simple class assignment) to show how data transmitted in WEP can be easily broken.

Insecure protocols and the secure counterparts

Every protocol in the original OSI model can have its secure counterpart (we can think the original protocol is not secure enough).

It was mentioned in the class of the correspondence: IP => IPSec, HTTP => HTTPS, TCP => SSL, FTP => SFTP without much detail coverage. Coverage on these topics can be more interesting if the students can do projects, develop code, compare the difference of the secure protocol with the non-secure one etc.

4.2.2 Module Under Development:

Ping-Ponging

In Schwartz's book[9] on email spam, he mentioned an interesting problem of automatic email response:

A graduate student wrote a program called vacation which sent an automatic email message saying "I am on vacation" to anybody sending an email to the program's author. This one worked fine until another graduate student copied the program. The two programs got caught in a loop and proceeded to send each other message after message until the computer disk filled up.

To educate the students about the danger of such program and protocol, the author modified a client / server lab's source code so that the connected server and client will respond with "Hi" when it receives a message. In order not to be caught in the infinite loop, the author has limited the number of ping-ponging to say 5 times or a user defined finite number for the students to observe. This lab can be changed to ping-pong with 50, 100 etc. messages with longer message and the memory usage, CPU performance, and hard disk space put under observation to see how the impact is (it is a hard one to control. Too small messages and too few times, the impact is not obvious, too big ones, the system could be brought down by this experiment).

Also, a lab on sending and reading e-mails can be modified in the future to emulate this ping-ponging as described in Schwartz's book. Again, it has to be done very cautiously so that the problem mentioned there does not happen.

4.2.3 Modules to Be Developed:

TCP SYN attack

TCP handshake is a three steps protocol: first the initiator sends a SYN package to the cooperating computer, the cooperating computer responds by sending a SYN / ACK package back to the initiator, and finally the initiator sends an ACK package to the cooperating computer, completing the 3 ways handshake (after that the data can be sent both ways). The SYN and SYN / ACK package should contain the IP addresses of both computers.

TCP SYN attack is explained in the book on Network Security[8] as follows:

The initiator sends a TCP SYN packet with invalid IP address; the cooperator attempts to respond a SYN / ACK packet to the IP address. Since it is invalid, the cooperator's response will go nowhere and the cooperator will be kept busy processing these nonsense messages.

Module will be developed to show how this looks in the attacked computer.

DOS and DDOS attack

DOS (Denial of Services) attack is that if a computer receives data packets from one computer faster than it can process, then it can not serve the other users remotely since its CPU time is all used. TCP SYN attack is one kind of DOS attack. DOS can also happen by some packet generators. What's good (or bad) about DOS attacks is that the attacked computer can use some IP tracking mechanism to find the "criminal".

DDOS (Distributed Denial of Services) attack happens when several or many computers send a lot of message to a single computer in a short time, hoping to bring it down. The criminals are hard to detect in this case.

Module will be developed to emulate the slow down (not shut down) of the attacked computer when DOS or DDOS happens. We will try to see if the attacked can detect when it is attacked, track back if possible. This will be some hard attempt, but worth of pursuing.

Ping attack and analysis (Ethereal tool)

Ping is a common command used to check if a remote system is up and running. The syntax of ping is easy: ping <hostname> and the response is easy to find:

Either the remote system is up and running and the response is a repetition of

Response from <hostname>, bytes = 32, time = 25ms, TTL =53

Or the remote is down or does not exist in DNS server and the message is either

Request Timeout

Or

Ping request could not find <hostname>.

The computer can detect that another computer is pining it using some network sniffer tool such as Ethereal.

The code of ping is also available and the author has used that to ask the students to build ping.exe from ping.c or making minor changes to code as well. The students will use the code of ping.c as a start point to do any variations of ping.

It is possible to detect if another computer has *ping*ed this computer. The freely downloadable tool Ethereal will be used (it is a tool that can be installed with supervision; it is not an exe file like ping command) to monitor the incoming packets including the ICMP packets (ping uses ICMP protocol).

Ankit Fadia[8] also mentioned the Ping of Death attack where ping command is used to send a huge datagram (such as ping –l 65540 hostname) that can cause the remote system to hang or reboot. But he commented that fortunately most systems nowadays are not vulnerable to the Ping of Death. This can be an assignment for students to try on ancient systems (under supervision of course!).

Email analysis

Email analysis was discussed in many books on stopping email spams[9]. Module will be developed that checks the e-mail header of incoming e-mail to glean as much info as possible. The module will start from some ready to do send mail and check mail source code.

4.3 Programming Courses:

Most of the modules mentioned in the Operating System Security section (section 4.1) and Network Security section (section 4.2) can be done using computer programs (a few modules or actions in the modules are the usage or procedures).

C Security Weaknesses

C is well known for its being insecure (such as not checking array out of bound). Security modules on C language consist of the security flaw in C language (including buffer overruns), implementation of security defense algorithms (like encryption) or attack algorithms (like some port scanners not mentioned in this paper).

Java Security Package

Java is in general known as more secure than C language. The modules can cover any of Java insecurities (from papers) if any, Java security package / class, or using Java to develop security algorithms like C security algorithms.

Encryption

Encryption is a good start of programming security algorithms, especially the trivial simple Caesar cipher (still not all students know how to program this). This can move on to more sophisticated Vigenere cipher, RSA private key cipher, and the elliptic curve cryptography cipher (ECC) etc.

Authentication

Authentication is another possible security algorithm that the students can implement. The common MD5, RC4 etc. can be used as the materials (students can start from scratch or modify some working code).

XOR

The XOR (exclusive or) operation is a famous easy way of encryption. Unfortunately it also has the reputation of very easy to crack as known from the WEP protocol in IEEE 802.11. It is easy to develop assignments or modules in XOR to let the students have the hands on experience of encryption as the defender and then the easy cracking as the attacker.

Breaking password

As mentioned in the OS security section, passwords can be broken by brute force and the difficulty depends on the length of the password. It is also a good simple example for the students trying to write code to break the password using brute force or heuristically (the classic paper by Morris and Thompson[10] of trying last names, first names, street names etc.). The heuristic example would be a good programming example for the students to use database programming.

5. Conclusions

It can not be overemphasized that security education is an important and integral part of all the engineering students. The discussions above hope to find out a general direction that the author and colleagues can work in the future to make the engineering students more informed and prepared for the security.

Security modules are usually covered in the classes as PowerPoint presentations that the instructors state the facts and students act as receptacles without too much participation and involvement.

The more learning centered approaches would be incorporating study, homework, lab, or project assignments in the modules so that the students will get more active participations.

There can be other creative ways of how security education can be carried out. Internet forum may be a way, where people from different places can exchange ideas (and students can get the updated materials from the instructor through some tools like Blackboard, WebCT etc.).

Future Efforts and Directions of Security Education

Two modules designated as under development while writing the draft paper were developed during this period: ping-ponging of the message for TCP / IP and password cracking and they received good responses from the students. Since new security problems are discovered all the time, the battles of security, and security education are on-going efforts forever.

The author will continue to enhance the existing modules and develop the modules designated as to be developed in section 4. There are still many security modules that need development: for example, email spam analysis, email spam buster software will be useful for computer forensics, spyware buster software will be useful for operating system security or object oriented software design, intrusion detection system will be useful for operating system security etc.

The evolvement of these security modules into a good courseware and the feedback from the students will be the topics for the future papers. The author also plan to cooperate with the colleagues from ECE department in the college of engineering, Computer Science Department in the college of science, and Computer Information System department in the college of business for the exchange and cooperation of security education modules development.

It is also possible that different instructors of the same department, different departments, or even different schools[1, 11] (George Mason University and James Madison University in reference 1; Cal Poly Pomona and H.T.C.I.A. (High Technology Crime Investigation Association) in reference 11) can team up to teach a curriculum or a course in the curriculum. These are all the possibilities leading to a more security aware academic community ready for the challenges in the near future.

References:

1. “Building an Undergraduate Security Curriculum”, Anne Marchant, Edgar H Sibley, Hugh Taze ell (Taz) Daughtrey Jr. George Mason University/ James Madison University, ASEE 2004, Salt Lake City, Utah.

2. “Development of a Graduate Certificate in Information Assurance”, Dr. Doug Jacobson Department of Electrical and Com uter Engineering, Io a State University, ASEE 2004, Salt Lake City, Utah.

3. "Integration of Security into the Development and Teaching of a New 2-Year Program in Wireless Communications" . Michael Qaissaunee, Mohammad Shanehsaz. ASEE 2005, Portland, Oregon.

4. "SEI SIA Curriculum / VTE" , Carol A. Sledge, Ph.D. CERT ® Training and Education, The 2nd Annual Information Assurance and Regulatory Compliance Symposium, 2005, Pomona, California.

5. http://rissc.mtsac.edu/RISSC_NEW/curriculum_intro.asp
6. http://securityresponse.symantec.com/avcenter/venc/data/cih.html
7. "Modern Operating Systems", Andrew Tanenbaum, 2nd edition, Prentice Hall, 2001, ISBN 0130313580.
8. "Network Security. A hacker's perspective", Ankit Fadia, Premier Press, 2003, ISBN 1592000452.
9. "Stopping Spam", Alan Schwartz and Simpson Garfinkel, Oreilly, 1998, ISBN 156592388X
10. "Password Security: A Case History", Communications of ACM, vol. 22., pp. 594 – 597, Nov. 1979.
11. "Teaching Computer Forensics", Anna Carlin and Elizabeth Sykes, The 2nd Annual Information Assurance and Regulatory Compliance Symposium, 2005, Pomona, California.