AC 2008-89: BOUNDARIES AND FLOWS: A STRATEGY FOR INTRODUCING INFORMATION SECURITY TO UNDERGRADUATES

Richard Smith, U. of St. Thomas - St. Paul

Assistant Professor at the University of St. Thomas and author of two books on information security.

Boundaries and Flows: A Strategy for Introducing Information Security to Undergraduates

Abstract

Outside of 2-year technical colleges, most postsecondary students aren't offered coursework in information security until they have fulfilled upper division prerequisites in mathematics, software systems, and networking. This is because many textbooks present information security in terms of those other topics. We are experimenting with a different approach: a lower division undergraduate course that introduces students to the concepts of boundaries and information flows. Professional security engineers often analyze problems in terms of these basic concepts. The course introduces security concepts by starting with security issues of small scale perimeters, and incrementally expands the scope by looking in turn at shared single computers, local area networks, and the Internet.

1. Introduction

When the Computer and Information Sciences Department at the University of St. Thomas began to develop an information security program, two objectives emerged. A natural objective was for the program to draw new students into the department. A second goal was to provide an introductory security course that was accessible to as many students as possible. Ideally, this would be a lower division course available to sophomores and even qualified freshmen. The prerequisites would be limited to one introductory programming course and a college math course: this would provide a pool of students typically pursuing engineering and the sciences.

A lower division course like this, however, does not match the typical pattern for a computer security course. In most four year institutions, information security coursework begins with an upper division course whose prerequisites include networking, operating systems and, in some cases, advanced math courses. This was not the introductory course we wanted to teach.

We decided to plan a course with the following properties:

- Prerequisites limited to introductory programming and a college math course
- Course work would promote higher order thinking skills according to Bloom's taxonomy of cognitive learning levels[4].
- Content would teach students *practical* information security skills: skills that would help students analyze real-world security situations

In our search for support of this alternate course model, we examined numerous textbooks. At the high end are books like Bishop[1] that focus on a mathematical treatment of the subject. Other texts, like Whitman and Mattord [11] do not require the mathematical background, but do not teach analytical techniques. Instead, they present lists of technologies and processes, which often yield courses based more on rote memorization or simple applications of predefined solutions to recognized problems. This does not prepare students to analyze real world problems, which evolve continuously in the face of escalating security threats.

This led us to develop our own course from scratch. Originally entitled "Elementary Information Security," the first word was eventually dropped. The course covers the following:

- Security policy concepts
- Risk and vulnerability assessment
- Security technologies
- Security planning and architecture

While developing this course, the CIS department also developed an information security major program that would fulfill US government requirements for information security education. The department would seek certification for the major under the Information Assurance Courseware Evaluation Program[9]. Since the curriculum focuses on training "information security professionals," the department sought to meet the requirements of Instruction 4011 of the Committee for National Security Systems[7]. Although this document dates from 1994, it is still the curriculum standard used by the US government. To support this, the course includes exercises to cover almost all "learning outcomes" listed in the standard (omitting detailed penetration analyses, as noted below). The course also covers roughly 80% of the core and elective learning outcomes of the Information Technology computing curriculum proposed in 2005 by the Association for Computing Machinery[2].

2. Boundaries and Flows

These two concepts were chosen as the organizing concepts or "themes" of the course.

A *boundary* separates two elements of a system and may thereby provide a degree of protection. As stated in the US government's Information Assurance Technical Framework (the IATF), "Information assets exist in physical and logical locations, and boundaries exist between these locations"[8]. This statement reflects the fact that students must consider a system's physical as well as logical elements when assessing security. Some authors talk of boundaries and perimeters in the context of *tamper resistance*, (see Chapter 14 of Anderson[1]), but tampering is only relevant after we define a perimeter intended to resist tampering.

An *information flow* indicates a possible path for information to move (or "leak") from one region of a system to another. We can diagram an information flow using similar elements to those used in the data flow diagrams typically employed in software design. The analysis of information flows is a classic approach for analyzing systems that require high security assurance as described in Chapter 16 of Bishop[3].

While not all security practitioners talk specifically about analyzing security in terms of boundaries and flows, the approach is common in some communities, notably the US government. The two concepts are applied a whole range of systems: from software-only and embedded systems all the way up to distributed, network oriented systems. At the embedded level, for example, cryptographic systems are often designed to enforce "red/black separation," which defines a boundary between elements of a device[10]. The IATF goes so far as to define

security objectives for network oriented environments in terms of boundary protection and the flows in and out of protected regions.

These two concepts provide particular benefits when presenting the information to students. In particular, issues and concepts can often be rendered in terms of diagrams that appeal to today's more visually-oriented students. Examples can also be based on tangible systems in which the boundaries are physical and the flows are visible via networking wires or other paths. In some cases the examples are already familiar to the students and/or they may physically visit them.

3. Course Syllabus

To meet the objectives, the course developed a particular topic progression, included essential non-security background topics, and selected reading materials, including a book to serve as the principal text for students. This section outlines those elements of the course. Section 4 outlines student work that was intended to reflect key course outcomes.

3.1 Topic Progression

Topics within the course are presented in "geographical" order. We start with a single computer and a single legitimate user, and then incrementally expand the user population and the network connections. The progression goes like this:

- Single desktop computer – introduce basics of physical security, security policy objectives ("Don't touch my computer!"), intrusion recovery, etc.
- Shared desktop computer – introduce user based access control, process protection in operating systems, and then file and volume encryption as an alternative, which leads to cryptography.
- Local area network – introduce networking basics, and the concept of user roles.
- Viruses and worms – malicious logic that tries to spread.
- Wireless networking – introduces "link layer" encryption
- Internet access – the problem of safe browsing on the web, especially from a LAN. This introduces firewalls.
- VPNs – safely connecting LANs across the Internet. This introduces public key cryptography as used in IPSEC/IKE.
- Socket layer encryption – protecting traffic for Web browsers. This introduces RSA public key encryption.
- PKI – this introduces digital signatures, certificates, and their problems.
- E-Commerce – this introduces the risks to a site that provides service to the Internet, particularly Web service. This covers additional features of firewalls.

This progression introduces the students to the problem of assessing security perimeters that divide a system into more-trustworthy and less-trustworthy components. The students start with a simple household-based scenario, then they work with a small business LAN, and incrementally build up to more sophisticated systems. At each level, the students review fundamental features of computer hardware and software at all levels of a system.

Information security engineering involves a life cycle starting with requirement analysis, progressing through design analysis and deployment, and repeating the cycle following a period of system monitoring and incident response. At each point in the progression, students look at specific problems that are solved by particular security measures applied to the system at that level. The problem drives the requirements specification, and the available technologies drive the design development.

3.2 Filling in Missing Background

Students generally understand the role of doors, walls, locks, chains, and so on, in physically securing an object like a bicycle, and often have an intuitive appreciation of how these are rooted in mechanical interlocks. Hardly any understand the mechanisms underlying *logical* protections on a computer. Some have only a hazy notion of what logical protections exist. These logical protections rely on fundamental concepts from operating systems or computer networking, notably the following:

- Internal, CPU-based protection mechanisms
- Processes and process separation
- Layering in software architecture
- Role of message exchange in network protocols
- Role of "layering" in network protocols

This is why many security courses are junior- or senior-level courses: they assume that such coursework was finished first. We can't simply ignore these topics, since certain essential mechanisms rely on these concepts. The course provides an introduction to each of these topics so students are aware of the mechanisms underlying higher-level security concepts.

The central processing unit (CPU) provides the starting point for logical security protections. The course introduces the kernel mode/user mode distinction which provides the fundamental mechanism for security. Students are shown how this puts all input/output and memory management operations under the operating system's control. These mechanisms prevent a user program from arbitrarily changing RAM or file contents without prior permission.

The concept of *processes* provides the fundamental concept for understanding the execution of programs within a computer. Lower division students aren't always familiar or comfortable with thinking about processes, but the concept is essential to reasoning about logical protection within a computer. When working above the operating system level, which is the focus of this course, all logical protections are in the context of processes and what those processes are allowed to do.

The concept of layering in software architecture is an essential part of security: we can often achieve a security objective if we can interpose a software layer that filters all operations and successfully distinguishes between those that should be authorized and those that should not. Lower division students rarely see this, but they need the concept in order to understand network protocol stacks.

Network protocols seek to exchange data between computers by exchanging just enough messages to ensure that both machines agree as to what data has been exchanged. Students need to understand this mechanism as the basis of network protocols and its role in ensuring coordinated data transfer and reliability. The students should not, however, need to understand particular protocols, except perhaps when trying to understand particular attacks.

Network protocols use layering in order to divide the work of data transfer among appropriate elements of the network system. Students need to understand how this affects the architecture of network software (layers of processes or procedures) and of network data packets (encapsulations of higher layers by lower ones). Layering plays a critical role in the use of network encryption, so the location of cryptographic software affects what data and services are protected.

3.3 Textbook

As noted in the Introduction, existing textbooks either require upper division course work or they fail to really teach techniques that incorporate higher order thinking skills. Information security professionals perform a variety of tasks: security policy development, risk assessment, security design evaluation, optimal password or key security estimation, and security plan development. There are specialized professional books that cover several of these topics. There are very few books that present these at an introductory level so that students can learn the fundamental concepts.

The pilot version of this security course used the book *Internet Cryptography*[10] in conjunction with other readings. The instructor developed assignments and exercises for the course since the book is not a textbook and does not provide exercises. The book was chosen because it makes extensive use of perimeter and flow concepts to address network security.

During the pilot course, the instructor secured a publisher for an appropriate textbook based on this course concept. The proposed book provides a topic progression similar to that described here. It also provides tutorial background material so that students do not need courses in operating systems and networking before studying information security.

4. Student Work

Students work with security concepts in a practical context by performing analyses that are used by information security professionals: risk assessments, vulnerability assessments, and security plans. They learn how to use some basic scanning tools to survey small networks and assess possible vulnerabilities. They perform simple risk assessments in which they must balance the impact of security measures against the potential reduction in risk. They also write security plans in which they describe the security measures that must be implemented to block specific weaknesses in a system.

Each type of student work here is considered in the light of Bloom's taxonomy for cognitive learning objectives[4]. In Bloom's taxonomy, there are six levels of cognitive learning objectives:

- 1. Knowledge
- 2. Comprehension
- 3. Application
- 4. Analysis
- 5. Synthesis
- 6. Evaluation

The upper 3 are generally considered "higher order" thinking skills. The lower order objectives generally involve memorization or the relatively simple application of rules to solve problems. The higher order objectives involve more sophistication, insight, thought, and creativity.

The Bloom taxonomy is not ideal for characterizing this type of work[5], but it is explicitly required for assessing course activities for certain purposes at the University of St. Thomas, so it is used here. The objective of this course is that important student tasks should require Bloom Levels 4 or 5. Bloom Level 6 would correspond to a sophisticated vulnerability or penetration analysis, which we are deferring to a more advanced course.

4.1 Perimeter Analysis

In a perimeter analysis, the students look for a real-world example of a computing system. They develop a high-level policy statement for the system, which summarizes the system's purpose so that they can identify risks as threats to that purpose. Then students describe the protective boundary around that computer in terms of physical and logical protections.

In a simple case, students might look at their own computer at home, in a dorm room, or in an apartment. The policy illustrates what they rely on the computer for and who is allowed to use that computer. Risks are described in terms of unauthorized use that interferes with their own use, or physical loss. The perimeter is described in terms of how the device is physically protected. This type of analysis typically involves "application" of security concepts, placing at Bloom Level 3.

A more sophisticated problem incorporates networking and logical protections. This usually involves Internet access: identifying network-based risks as well as any network-oriented or internal (anti-virus) protections used. For example, students might look at a larger-scale system used on campus or at a business with which they are familiar. They establish a policy statement for the system and identify risks. Then they describe the perimeter in terms of both physical and logical protections.

Although problems and solutions can become quite complex, these problems can often be solved through the "application" of security concepts, placing it at Bloom Level 3.

4.2 Risk Assessment

A risk assessment identifies and prioritizes potential risks to a system in terms of attacks that could interfere with its operation. Effective operation is defined in terms of policy statements about the system. The assessment identifies computing activities and resources that exist and are

essential to achieving enterprise objectives. Then it identifies threats (active agents like hackers, spies, embezzlers, etc.) and potential attacks (exploitations of vulnerabilities) that could interfere with those resources. The assessment is completed by establishing the relative significance of different attacks by assessing likelihood and the potential costs of different attacks.

A simple risk assessment may only involve the application of security rules and concepts, making it a Bloom level 3 exercises. However, a sophisticated risk assessment needs to be more subtle in identifying and prioritizing possible risks, requiring a level of "analysis" that makes it a Bloom Level 4 exercise.

4.3 Analyzing exhaustive attacks

These exercises analyze different types of exhaustive (trial-and-error) attacks on computing systems. These include attacks against various secrets including authentication devices, cryptographic keys, and hash-based integrity checks.

In most cases these techniques simply require "extrapolation" of standard calculations for different types of attacks, which places this activity at Bloom Level 2. However, some problems, like estimating the number of possible passwords that match a complex construction rule, may require higher levels of mathematical reasoning, making it a Bloom Level 4 activity.

4.4 Crypto: Spot the Plaintext, Spot the Key

In these exercises, students draw a diagram of a network architecture that incorporates one or more layers of cryptography complying with well-known standards, like wireless encryption, IPSEC for corporate sites, SSL for web sites, or e-mail message encryption. They must identify where the protocol layers reside, where encryption takes place, where plaintext is visible, and where ciphertext is visible. Figure 1 provides an example of this type of exercise.

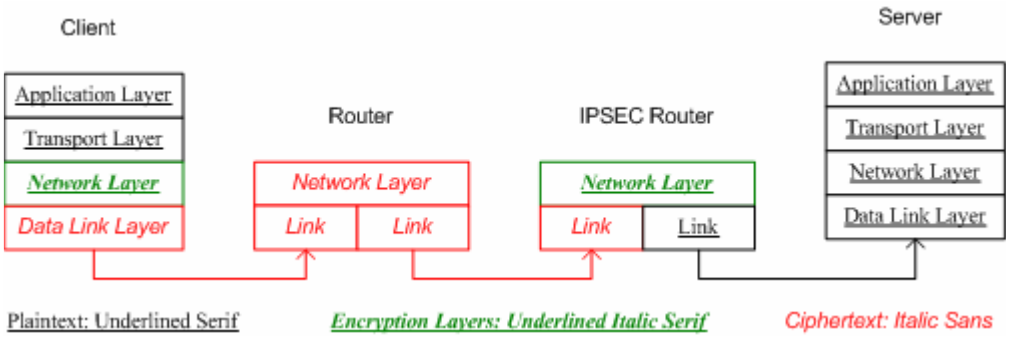

Figure 1: Example exercise in marking plaintext and ciphertext on a network

These exercises can follow fairly simple rules and templates that are simply layered, one upon another, to yield a complex result. The complexity of the result may appear to deserve a higher order of cognitive effort, but in fact, students might not need to exercise greater than Bloom level 3 in completing it.

4.5 Security Plan

In a *security plan*, students establish appropriate protections for a set of objectives and a set of computing resources. Real-world plans achieve high levels of sophistication and clearly represent a "synthesis:" a Bloom level 5 cognitive effort. However, simpler plans may be constructed from rules or simple analyses. The simplest security plans are reflected in simple perimeter designs. More sophisticated plans involve protecting files within a multi-user system by applying appropriate file protections. By the end of the course, students were developing network security plans that incorporated encryption and firewall rule sets to protect against a range of attacks while providing well defined services. Such plans represent "designs" in the sense of the Bloom taxonomy, which is a form of Level 5 synthesis.

5. Course Assessment

There are three different assessments relevant to this course: an assessment of the Bloom Level of student activities, a standardized collection of student feedback, and qualitative observations regarding student homework by the instructor. The Bloom Level assessment was described in the previous section. The other two assessments are discussed below.

At the University of St. Thomas, there is a formal process for collecting student feedback at the end of every course, called the Student Report on Teaching, or SROT. Three separate courses in computer security have been taught over the past four years. The course described here shows a 17% improvement in SROT responses as compared to the earliest "general computer security" course and a minor increase (3%) over a network security course.

The course and assignments described here seems to have yielded the most successful results in terms of student homework. Unfortunately there were no specific numbers collected and retained regarding student homework in earlier courses, so a qualitative assessment must suffice. Earlier courses used assignments from standard textbooks which students found frustrating, partly because the University's computer science curriculum focused on information systems design and did not provide the more theoretical background (and associated analytical techniques) that typical security textbooks assumed as prerequisites. Even a faculty member who had sat in on an earlier course was moved to observe in a peer review that the assignments taken from the course textbook[6] could be very challenging.

5. Conclusion

At the University of St. Thomas, we have offered several computer security courses over the years. Originally we offered mathematically-oriented courses in cryptography. More recently we have offered courses on network and computer security that served as test beds for teaching

security concepts. This most recent course has yielded the most satisfying results, in terms of student achievement and student satisfaction, as well as presenting practical techniques accessible to lower division undergraduates.

Acknowledgments

This work was supported by new faculty funding and a Research Assistance Grant at the University of St. Thomas. I would also like to thank my colleagues and my students in the Department of Computer and Information Sciences for their valuable input and assistance. I would also like to acknowledge the input and recommendations of colleagues who participated in the University of Minnesota Summer School for Information Assurance, notably Dr. Youngdae Kim and Dr. Nick Hopper.

Bibliography

1. Anderson, Ross (2001). Security Engineering, John Wiley, New York.
2. Association for Computing Machinery (2005). Computing curricula: information technology volume. Association for Computing Machinery. October 2005.
3. Bishop, Matt (2005). Introduction to Computer Security, Addison Wesley, Boston.
4. Bloom, B.S. (1956). Taxonomy of educational objectives; Book 1: Cognitive domain. Longman, New York.
5. Fuller, U., Johnson, C., Ahoniemi, T., Cukierman, D., Hernan-Losada, I., Jackova, J., Lahtinen, E., Lewis, T., Thompson, D., Riedesel, C., Thompson, E. (2007). Developing a computer science-specific learning taxonomy, Proceedings of the 12th annual conference on innovation and technology in computer science education, June 2007, Dundee, Scotland.
6. Kaufman, C., Perlman, R., and Speciner, M. (2002). Network security: private communication in a public world (second edition). Prentice Hall, Upper Saddle River, New Jersey.
7. National Security Telecommunications and Information Systems Security Committee – NSTISSC (1994). National Training Standard for Information Security (INFOSEC) Professionals. NSTISS Instruction 4011.
8. National Security Agency – NSA (2000). Information assurance technical framework (release 3.0). Information Assurance Solutions. September 2000.
9. NSA (2007). National IA Education and Training Program, web site, http://www.nsa.gov/ia/academia/acade00001.cfm (retrieved 27 June 2007).
10. Smith, R.E. (1997). Internet cryptography. Addison-Wesley, New York.
11. Whitman, M., and Mattord, H. (2004). Designing and teaching information security curriculum. Proceedings of the InfoSecCD Conference '04, October 2004, Kennesaw, GA.
12. Whitman, M., and Mattord, H. (2005). Principles of information security, second edition. Thomson Course Technology, Boston.