6.857 Computer and Network Security **Fall Term, 1996**

Lecture 1 : September 4, 1997

Lecturer: Ron Rivest *Scribe: Stacey Blau*

6.857 is a subject on computer and network security. It is intended primarily for seniors and first-year graduate students. Students should have taken 6.042J (*Mathematics for Computer Science*) and 6.033 (*Computer System Engineering*) before taking this class. They should feel comfortable with algorithmic concepts and modular arithmetic.

The topics covered in 6.857 can be divided into three main parts. The first part of the course begins with a brief introduction to the topic of security, specifically physical security. The course will then delve into a fundamental tool that we will use during the rest of the course: *Cryptography.* We will discuss classical secret–key cryptosystems (DES) and more recent public–key security systems (PGP) as well as the mathematical bases for these systems. We will introduce digital signature schemes and discuss some more sophisticated cryptographic protocols like secret sharing, oblivious transfer, commitment, certificate authorities, and zero–knowledge proofs. We also will talk about current topics like key escrow, pay TV security, and the export of cryptographic technology.

In the second section of the course, we will talk about key exchange and key agreement and then move on to *Computer Security.* This area will cover topics like authentication, authorization, auditing, computer viruses, covert channels, secure operating systems, smart cards, and biometric technologies like thumbprints and eyescans. We also will discuss formal models for computer security, modes of risk analysis, methods of attacks, and security testing.

Finally, we will move on the area of *Network Security* with a special focus on the Internet. We will talk about the Internet Protocol and its security issues and study firewalls and network protocols like Kerberos. We also will look at the World Wide Web, Internet commerce, and electronic payments. In addition, we will study electronic voting and electronic intellectual property protection, which encompass many of the computer and network security issues we will have discussed.

Students will be responsible for team term projects. At the conclusion of the course, all students will present their projects to the class.

1 Security Objectives

In this class we will learn how, given a computer system, we can identify and possibly solve its security problems. This means that we have to define what we mean by security. The computer system can be anything you may think of: a shared system, a Web server, a bank ATM, the cable–box on your TV, and so on. For each case, defining security means specifying a *security policy*, i.e., a set of desired goals. For example, in an electronic voting system, we decide to set up the system so that only registered voters can vote; for a Web server, we may decide that clients accessing the server should pay a fee; in a shared system, only authorized users should be able to log in; and so on.

Once a security policy has been specified we need to put in place *security mechanisms*, i.e., tools that make sure that the desired goals are met. For example, a verification process for registered voters should ensure that only registered voters can vote. A payment protocol should ensure fee payments for clients accessing a Web server. A password scheme should give access to a shared system only to authorized users.

Once the security mechanisms are put in place, they need to be reviewed in order to assess possible *vulnerabilities* of the system, i.e., weakness that leave the system open to attacks. Most systems are not bulletproof. Hostile parties who exploit weaknesses in order to create damage are known as *threats*.

If a vulnerability is identified, then a *safeguard* or *countermeasure* must be designed to eliminate the weakness.

1.1 Security Policy: Goals

There may be many different security goals depending on the purpose of the computer system in question. We can, however, group goals into three broad categories:

Confidentiality: Contents of a computer or contents sent across a network should not be accessible by unauthorized parties. (For example, a student's grades should be kept private, and he should have secure remote access to them.)

Integrity: Contents of a computer or of contents sent across a network should not be modifiable by unauthorized users. (For example, only professors should be able to modify grades, and no one should be able to alter them if they are transmitted across a network.)

Availability: The system should be available to authorized users. A *denial of service attack* is an attack that infringes upon availability of the system. (For example, students should be able to view their own grades.)

Other goals may also come up. For example, in the protocol for an electronic payment, anonymity might be required for the parties involved. Such an example does not seem to fit neatly into any of the categories above (although confidentiality is probably the closest).

1.2 The Design of a System

The design of a security system involves a loop of three main steps:

Design: The description of the system is introduced, and the designers present a proof that the system is secure.

Implementation: The design of the system is implemented, and its security is tested and verfied to a reasonable extent.

Breakdown: The security system fails. The process returns to design.

1.3 Mechanisms

Security mechanisms evolve over long periods of time. For example, the current design of bank vaults took years to evolve; they were improved as thieves found new ways to break into them. It is difficult to know precisely when security has been achieved because it is much harder to prove a negative (that no security hole exists) than it is to prove a positive (that a concrete security flaw exists). Also, a security mechanism can be proved correct, but doing so requires assumptions about what potential adversaries can do (passive attacks, active attacks, etc.). Such assumptions are sometimes difficult to pin down and often new ones arise as situations change.

In the course of this class we will explore many possible security mechanisms. Some typical ones include in order of more sophistication:

User awareness: Educating users about the security goals of a system and its possible risks is one of the most effective security mechanisms.

Physical protection: Locks and keys prevent unauthorized access to buildings where computer systems reside. Shredders make snooping in the trash harder. Degaussers actually delete the content of a floppy (differently than the DELETE command that usually just removes a pointer to them).

Cryptography: As we will see later in the class, encryption systems can be used in different ways to enforce both confidentiality and integrity.

Access control: Only people authorized to have privileges to certain machines, files, etc., should have access to them. In UNIX, for example, access control lists specify which users can read, write, execute a given file.

Auditing: Recording all the activities that take place in a system makes it possible to detect security breaches that could not be prevented.

Some security mechanisms are aimed at *prevention* of security breaches while others are aimed at their *detection*. The latter is particularly important since we never can be sure that we have ruled out all possible attacks. It is therefore important to detect something that has gone wrong and to take necessary countermeasures.

1.4 Principles

The design of effective security mechanisms involves some general *security principles*.

Principle of least privilege: Give a user or process only privileges needed to perform task at hand. No more, no less.

Minimize number of trusted components: Identify which components of the system need to be trusted and aim to keep those small and simple.

Don't aim for perfection: Perfection is basically impossible. Instead, be prepared to practice risk management, to detect problems, to design backups and countermeasures, and to recover from attacks.

Keep it simple: If the system can't be used easily, it won't be used.

Be skeptical: Don't accept people's claims that something is secure. Investigate for yourself.

Be paranoid: People put a lot of effort into attacking a system. Don't underestimate them.

2 Handouts and Readings

There were two handouts: Course Information (Handout 1) and Student Information (Handout 2). The latter was to be filled out and returned at the end of the first class.

There is no textbook for this class. However, there are a few useful books that you may want to consider purchasing. The books provide a good background on many of the issues that we will study in 6.857. They also are useful reference tools for the subject matter in general beyond the course.

The latter two books are somewhat similar and may be a bit pricey. You may want to consider purchasing only one of the two of them.

Computer-Related Risks, by Neumann (Addison-Wesley). A collection of case studies on how systems go wrong. Very useful and mostly fun reading.

Cryptography, Theory and Practice, by Stinson (CRC). A good, solid textbook on cryptography.

Handbook of Applied Cryptography, by Menezes, et al (CRC). A very detailed reference book on cryptography.