

ITEE



What's New:

You should aim to catch up on previous tutorials; this is the last tutorial, so there will be no further new material; the last week of lectures will include practice questions leading to the exam.

Learning objectives for this week:

- Understand the major concepts we've covered, including:
 - different types of protection
 - different aspects of the security problem
- Apply these concepts to reasoning about protection and security
- Apply these concepts to evaluating alternatives
- Understand issues in improving security

As before, you should aim to answer all of these questions by the end of the course.

1. Concepts

- a. When you use a UNIX file system, to what extent can you control access to files?
- b. How does an access control list relate to or differ from a capability?
- c. What is the difference between symmetrical and asymmetric encryption?
- d. What are the advantages of public-key encryption?

2. Protection

- a. Capabilities have been widely explored in distributed systems as a way of transferring rights to remote users. What specific features are important if capabilities are to be used in this way, without creating security holes?
- b. Consider each of the following scenarios, and describe how you could use access control lists, capabilities or UNIX file permissions to achieve what you are aiming for:
 - i. your header files and given source files for an assignment should be accessible to the whole class, but your personal implementation should only be visible to you, your tutor (and not others), and the lecturer. You should be able to read and modify your files (including compiling and running your finished code), and the others who have access to them should be able to read them, and run executables, but not modify files. Does it make a difference in any case if the differences in access rights are organized around directories or files?
 - ii. you are doing a group project, and the same conditions as in (i) apply, except other members of the group can read your files, but not modify them. There is a central directory accessible to the whole group where final linking takes place, but compilation of individual components happens in your directory, where others should not be able to compile, so only you can make a compiled version of a component you are responsible for available to the group as a whole.
 - iii. you are the administrator of a distributed system and would like to allow a specific user access to a remote file system.
- c. Draw an access matrix corresponding to scenarios of 2(b)(i-ii). Would it have been easier to answer the question if you did this first?
- d. Why, in general terms, is it hard to handle failures in distributed systems?

3. Security

- a. Remembering passwords is a huge problem. Discuss which of the following could contribute to the solution, or add to the problem.
 - i. force users to change passwords regularly
 - ii. force users to use long passwords which aren't similar to English words
 - iii. allow users to store all their passwords in one place with a master password
 - iv. a graphical user interface hides passwords behind pictures meaningful to the user, but unlikely to be guessed by anyone else (e.g. a map in which clicking on a location either reveals a correct password or a fake one to put intruders off track)
 - v. allow an access with an incorrect password without any warning of an error, but redirect the access to a secure environment with tripwires
 - vi. replace passwords by a "guessing game" in which your guesses are compared with previous times you've played the game: exact matches aren't looked for, but rather a pattern of similar thought processes (some AI could apply)
- b. Before World War II, France had massive defences on their German border. Germany attacked France through neutral countries, ignoring French defences. Can you think of a security attack on a computer system with similarities to the German approach?
- c. In each of the following, classify the security problem as a Trojan horse, a worm, a virus or a denial of service attack. If more than one applies, either select the most applicable, or a combination – try to justify your answer as the best variation:
 - i. a program attached to an email message installs itself by exploiting a feature of your mail client's handling of attachments. It scans your address book and sends itself to all the addresses in the list (it is a standalone program, which has its own implementation of the mail protocols)
 - ii. a program attached to an email message inserts itself into your mail client (exploiting a similar security hole to that in (i)) so that whenever you send a message, it attaches itself to the message you send.
 - iii. a program like that of (i) repeatedly resends messages at short intervals, and includes a reply-receipt header (on receiving the message, the receiving mail client will send a reply automatically)
- d. Describe how public-key encryption could combine with capabilities to provide a secure way of allowing an outsider *limited* access to distributed computer resources. This question should build on 2(a).
- e. Discussion: why would anyone develop a virus, worm or other malicious software? What should we do if such a person is caught?