Computer Science CS 6/79995 Operating System Security
Summer 2010
First Midterm
Due by email July 8 at 10:00 PM

Please be brief: Irrelevant or incorrect material will cost you points.

Keep in mind that, even though you might get the answer wrong, I will also grade your reasoning.; if you make a good case, you might get close to full credit even if you get the wrong answer.

1) The security triad talks about confidentiality (or secrecy), integrity and availability; the text (and this course) have been very silent on the issue of availability. Why would that be?

There are two aspects two availability: internal availability and external availability; internal availability refers to the reliability of the operating system and is beyond the scope of the course; similarly, external availability refers to the network and network security, again a different topic.

2) Regarding confidentiality and integrity:
a) Is it possible to have integrity in a system which does not enforce confidentiality? Why or why not?

Yes, it is possible; the fact that you can read the data does not mean you can modify it.

b) Is it possible to have confidentiality in a system which does not enforce integrity? Why or why not?

No, it is not possible; if you can modify the system, you can modify the access policy and then do what you want.

3) Is it possible to make Windows completely secure? Why or why not?

This is a loaded question; if you answered “yes” and justified it, you got full credit; my answer is “no” because there is just too much to secure, and there is too little information to go on.

4) Taking each of the security design principles in turn, discuss how applicable they are to the design of secure operating systems.

The principles are:

Least Privilege

Definitely a must; in fact, it is applied in many OS already covered.

Fail Safe Defaults

Violating this principle would negate all access controls.

Economy of Mechanism

In our case, we have been talking about minimizing the TCB; it is the same thing.

Complete Mediation

We have been requiring this principle in all evaluations.

Defense in depth

For example, the rings in Multics.

Open Design

In order to verify the security of the TCB, it is is necessary to know what is its design. That is what this principle is about.

Separation of Privilege

Has not been applied much, however, there have been applications.

Least Common Mechanism

Important to avoid covert channels.

Psychological Acceptability

If a system is too difficult to use, nobody is going to use it, so, definitely, this principle is applicable.

5) A truly secure operating system cannot be vulnerable to a confused deputy problem. How can this be accomplished?

A confused deputy happens when a program/system call with special powers is tricked into violating system policy by manipulating its arguments. This can be avoided by carefully checking each argument for correctness.

6) What information would be necessary to store in a system in order to implement a Lattice model protection scheme?

Short of storing the whole partial order relation, this is still an open problem. Probably the best way would be in a two-d array.

7) How would you implement a Clark-Wilson integrity policy?

There are many designs for this problem; the important part is to be able to enforce the condition inherent in CW, which would mean implementing a totally different structure to store the UDI's and CDI's and where execution and definition of TP's and IVP's makes sense; there, constraints can be placed on their execution.

8) What would be necessary to efficiently emulate Windows on the SCOMP?

For reasons similar to those discussed in class in the Unix case, this would be close to impossible.