Massachusetts Institute of Technology | Handout 14
6.915: Network and Computer Security | October 12, 1995
Professor Ronald L. Rivest

Problem Set 5

This problem set is due on *Thursday, October 19* at the end of class.

Each solution is to be handed in *separately*, i.e. on separate sheets of three-hole punched paper. If the solution to a single problem takes more than one page, staple together only the pages relative to the same problem. Mark the top of each sheet with your name, 6.915, the problem number, and the date.

If you collaborate with other students, you MUST write up solutions on your own and acknowledge the people you work with.

Readings: Chapter 6 of textbook

Problem 5-1. Auditing

Identify any potential single points of failure in the auditing scheme depicted below (the command `open(a,b)` is used as an example)

Problem 5-2. User Profiles

Suppose you want to create a profile for the typical Athena user. Such profile could then be used to compare the activity in the system to a typical one and detect possible intrusions.

Come up with at least twenty parameters on user activity that you think could be considered in creating such a profile. Divide them in three groups:

- very important parameters

- somewhat important parameters
- not very important parameters

Problem 5-3. Distributed Intrusion Detection

The intrusion detection system we saw in class focused on single–system stand alone facilities. However in practice we are in need to detect intrusion into a more distributed setting as a collection of hosts supported by a LAN (local area network.)

Suppose that each host in the network runs an auditing scheme that produces a log of the security–relevant activities on that host. Where should this log be stored? In the same host or in some other host in the network? If stored on another host, should the transmission of the log be done off–line (say daily at the end of the day) or on–line?

Analyze what are the pros and the cons of each possible alternative.

Problem 5-4. Access Control

Describe a mechanism by which an operating system can enforce *limited* transfer of capabilities. That is, process A might transfer a capability to process B, but A wants to prevent B from transferring the capability to any other process.

Your design should include description of the activities to be performed by A and B, as well as the activities performedby and the information maintained by the operating system.