CS167 Homework Assignment 4

Due Wednesday, May 2, 2018

1. [20 points] In the SELinux example given in the text in Section 8.2.2 (slides XXVI-27 through XXVI-31), roles are established for users participating in an accounting activity. Suppose user betty attempts to run the program requestPO. Assuming no SELinux rules other than those given in the text, explain what prevents user betty from using this program to request a PO.

2. One desirable (but perhaps unattainable) property of an RPC protocol is exactly once semantics: the guarantee that each remote procedure call issued by a client is executed exactly once on the server. Assume that the RPC protocol itself does not deal with reliability issues, but leaves this to the underlying communication protocols.

    a. [10 points] If RPC is layered on top of a TCP connection, and we can guarantee that the connection continues to exist for the duration of an RPC session, can exactly once semantics be guaranteed? Explain.

    b. [10 points] If the TCP connection is lost for some reason, then reestablished, can the RPC session (layered on top of the now two connections) still maintain exactly once semantics? Explain.

    c. [10 points] The OSI model defines a session layer on top of the transport layer. TCP is a transport protocol. Among the session layer's duties is to maintain the notion of a "session," which is analogous to the transport layer's connection, but covers the entire period of an RPC session (of many RPC requests and responses), even if there are multiple transport connections. If the state of the session layer is maintained in volatile storage (i.e., storage that is lost in the event of a crash), can exactly once semantics be guaranteed? Explain.

    d. [10 points] Suppose now the state of the session layer is maintained in stable storage (guaranteed to survive crashes). Can exactly once semantics be guaranteed? Explain.

3. Consider the following sequence of events in NFSv4:

    i. Client A requests an exclusive lock on all of file X and the client receives the server's response of "success."

    ii. Client A requests that the lock be "downgraded" to a shared lock. The server responds "success," but the response is lost.

    iii. Client B requests a shared lock on all of the same file. The client receives the server's response of "success."

    iv. The server crashes.

    v. The server restarts and starts its grace period.

    vi. Client B quickly recovers its shared lock.

    a. [20 points] The server is still in its grace period as client A attempts to reestablish its lock state. What does it do and what are the responses? (Hint: we may have a problem here.)

    b. [20 points] What might be done to fix this problem with the protocol?