Security

1. What is the difference between security and privacy? Are they entirely the same or entirely different or neither? Explain.

2. Explain the three key principles of computer security?

3. What is a threat model? What factors should you consider when defining threat model?

4. What hardware mechanism does x86 ISA provide to ensure that Operating System's code and data are protected from user-level processes?

5. What is the role of privilege levels (defined by the ISA) in a computer system? How many privilege levels are defined in the x86 ISA? In which privilege level does the OS execute?

6. (a) In an traditional x86 CPU, what are the execution privilege levels and their meanings? (b) How does Intel VTx extend the traditional x86 execution privilege levels to support system virtual machines?

7. Assume that a machine uses x86 ISA and runs any mainstream monolithic OS. Explain the basic security mechanisms provided by
   A. The CPU execution hardware
   B. Memory management hardware and software (in the OS)

8. Explain the basic security mechanisms supported by (a) the CPU execution hardware, (b) Memory management hardware and software, (c) File system. Assume that the machine uses x86 ISA.

7. In x86, how does the MMU figure out whether a code currently executing on CPU has permissions to read/write to/execute a given address in memory?

8. What is authentication?

9. Describe different techniques to authenticate users.

10. What are some ways in which by which authentication mechanisms can be subverted?

11. What is sandboxing? List two sandboxing mechanisms.

12. Explain Discretionary, Mandatory, and Role-based access control mechanisms.

13. What is meant by "trust" in computer security?

14. Explain (a) trusted computing base (TCB) including why is it called “Trusted”, (b) Reference Monitor, and (c) relationship between TCB and reference monitor.

17. Explain the two key data access principles of multi-level security (MLS) systems (also called Mandatory Access Control).

18. Why is Mandatory access control called “mandatory”? What’s the alternative?

19. Give an example of a scenario where the software doesn’t trust the OS, hypervisor, and/or the hardware platform on which it runs? What can the software possibly do to “secure” itself in this situation?

20. Considering memory protection, explain how the operating system ensures that user-level processes don’t access kernel-level memory?