Answer all questions. Please do not waste any words: answers are graded on content, not style. Where code is required, any kind of pseudocode is fine as long as its meaning is clear. Each question is worth 25 points. You get 25 points for writing your name on every page, for a total of 200 points.

Part 1: Protection

In a classic 1966 paper, Jack Dennis and Earl Van Horn laid out fundamental hardware structures for “protection of computing entities from unauthorized access” in “multiprogrammed computing systems”. Modern hardware platforms have adopted a simplified form of these structures to support “modern classical” operating systems, i.e., today’s descendants of Unix, Mach, and NT. After almost 40 years these platform structures are now evolving in directions not contemplated by Dennis and Van Horn. In connection with the Flicker and SubVirt papers we discussed two recent hardware extensions to broaden support for protection: Trusted Platform Module (TPM) and transparent hypervisor support (e.g., Intel’s VMX Root operation). The four questions in this part of the exam ask about basic protection features and these new platform extensions.

1. What are the key hardware features for protection assumed by the “modern classical” operating systems to protect the trusted OS reference monitor (kernel) from untrusted application code? You do not need to describe virtual address space maps.

2. A key feature of TPM is “remote attested measurement” in which a TPM issues a network message containing a “proof” of the contents of a range of physical memory at a point in time. Briefly outline the structure of such an attestation and how the receiver might check it. This question is not really about TPM: I am looking only for evidence that you understand the basic cryptosystem elements underlying attestation.

3. Briefly outline the purpose or goals or rationale for the new platform features: who/what are they trying to protect, and from whom/what? (a) Trusted Platform Module (b) Transparent hypervisor support

4. Assess the costs or risks of these new platform features using the following questions as a guide.

(a) To what extent must the “modern classical” operating systems change to take advantage of these new hardware features?

(b) The first rule for any kind of security work is “do no harm”. To what extent do the new features have potential to weaken security assurances for users of these systems?

Part 2: Servers and Concurrency

5. Draw rough sketches (“back of napkin” or “cartoon graphs”) showing the performance behavior typical of servers across a range of load levels. Briefly describe the graphs and their important features or transition points.

(a) response time

(b) throughput
6. A “typical” server such as Apache uses a basic structure often referred to as a “thread pool” or “work queue” or “multiple process” model using “forking” or “preforking”.

(a) Outline how this basic structure uses threads or processes. You can answer this question by writing pseudocode showing how the server partitions the work of handling requests across threads or processes.

(b) Outline the advantages and disadvantages of this basic structure.

7. The developers of the Staged Event-Driven Architecture (SEDA) claim that their approach is “robust to huge variations in load”, in contrast to the basic server structure of Question 6.

(a) Summarize what this claim means in terms of the graphs of Question 5.

(b) What should a server application do if the network delivers more requests to the server than it can handle? How would a SEDA server determine that it is overloaded? How might a SEDA server adapt to overload?

(c) SEDA presumes that the server application cannot change the resources allocated to it by the operating system. Is this a reasonable assumption? Discuss.

Have a great summer!