Question 3:

A computer system provides protection using the Bell-LaPadula policy. How would a virus spread if

(a) the virus were place on the system at system low (the compartment dominated by all other compartments)? Justify your answer.

Computer viruses spread via written activities. Bell-LaPadula model suggests that a process can write any item in the category (compartment) that rules the category (compartment) where the process is present. Every category (compartment) manages the system at its lowest level, so a system-low process can write to any challenge in any category (compartment). Or, to put it another way, a computer virus might spread even if it was given a low system priority.

(b) the virus were place on the system at system high (the compartment that dominates all other compartments)? Justify your answer.

Under the Bell-LaPadula Model, a process can write to any object in a category (compartment) that is managed by the category (compartment) that contains the process. Given that every compartment in the system is within the jurisdiction of the category (compartment) at system high, the machine virus cannot propagate to other compartments. Nevertheless, it could be disseminated among objects that belong to the same class at the same level (compartment) (system high in this instance).

Question 4:

Classify the following vulnerabilities using the RISOS model. Assume that the classification is for the implementation level. Justify your answer.

(a) The presence of the “wiz” command in the sendmail program (see section 20.2.8).

To identify, confirm, or provide authorization to the attacker, sendmail programs with the "wiz" instruction are insufficient. Guessing the permissions of the mail-sending server allows anyone to execute commands as that user. It permits an inaccurate identification of the user.

(b) The failure to handle the IFS shell variable by loadmodule (see section 20.2.8).

As the source code is already available, the notion has to be tested by disassembling the load module executable. All you need to do is change the IFS value to include "/," clear PATH and IFS, and then rename the little program that was originally in the ld. No changes to the source code are required. Make your way to the bin to start the load module. An effective UID of 0 for the process is printed. With success, the exam was passed. Consequently, the generalization's basic error is revealed. An environment in which a vested program operates should not be appropriately sanitized by asking whether subprograms can be trusted because of the problem with subprocesses and inheriting environment variables with their values. This is an illustration of a problem with the parameter validation process.

(c) The failure to select an Administrator password that was difficult to guess (see section 20.2.9).

Using the RISOS architecture, the inability to choose an Administrator password that is hard to guess can be categorized as an Elevation of Privilege security risk. This grade is due to the possibility of an attacker guessing the password, gaining elevated access, and using system resources without authorization. The administrator essentially offered attackers the opportunity to increase their privileges and perform damaging activities on the system by choosing a weak or simple-to-guess password. Consequently, the implementation level Elevation of Privilege vulnerability is caused by the incapacity to choose an Administrator password that was challenging to guess.