

ex6

In addition to mathematical and informal statements of policy, policies can be implicit (not stated).

a)  Why might this be done?  b)  Might it occur with informally stated policies?  c)  What problems can this cause?

Solution:

1. Policies may be implicit for a number of reasons.

The policy may be ambiguous, and the resolution of the ambiguity left to the reader; thus, the exact policy is not explicitly stated.

The policy may not cover all aspects of the system; those aspects not covered by the explicit policy would presumably be covered by the implicit policy.

The institution owning the computer may simply choose to tell users to use “common sense”; this is also an implicit policy.

b)  It is highly likely that informally stated policies will have many areas of ambiguity and not cover all contingencies. Hence these types of policies often lead to implicit policy components.

c)  The main problem with implicit policies is that: § not all users may know about them, or may have agreed to them.

The statement that “common sense is so unusual because it’s not common”  applies here.

Given that people cannot refer to an oracle, or source, for an implicit  policy but instead must gather opinions and make their own decisions, which may disagree with those of the system managers, a user may find herself violating the security policy without realizing it or intending to violate it.

ex7

Problem 7

a. Prevention is more important than detection and recovery.

Example: Prevention of Virus Infection in a computer is more important than its detection and recovery. If a computer is already infected with a virus, it may corrupt and delete data, which may not be recoverable.

b. Detection is more important than prevention and recovery. Example: Although prevention is always better than cure, Detection would be more important in cases when it is very hard to prevent a certain type of attack. This is true in Intrusion Detection Systems. For example, if a service is to be provided, there is always a threat of a Denial-of-Service attack. Such attacks should be detected first to prevent the unavailability of the service.

c. Recovery is more important than prevention and detection. Example: In a hard disk crash, recovery of the users files and other information is more important. All Hard disks are probable to crash after some time, so it is hard to prevent such crashes, but a recovery plan should be put in force before hand by using RAID arrays and weekly backups.

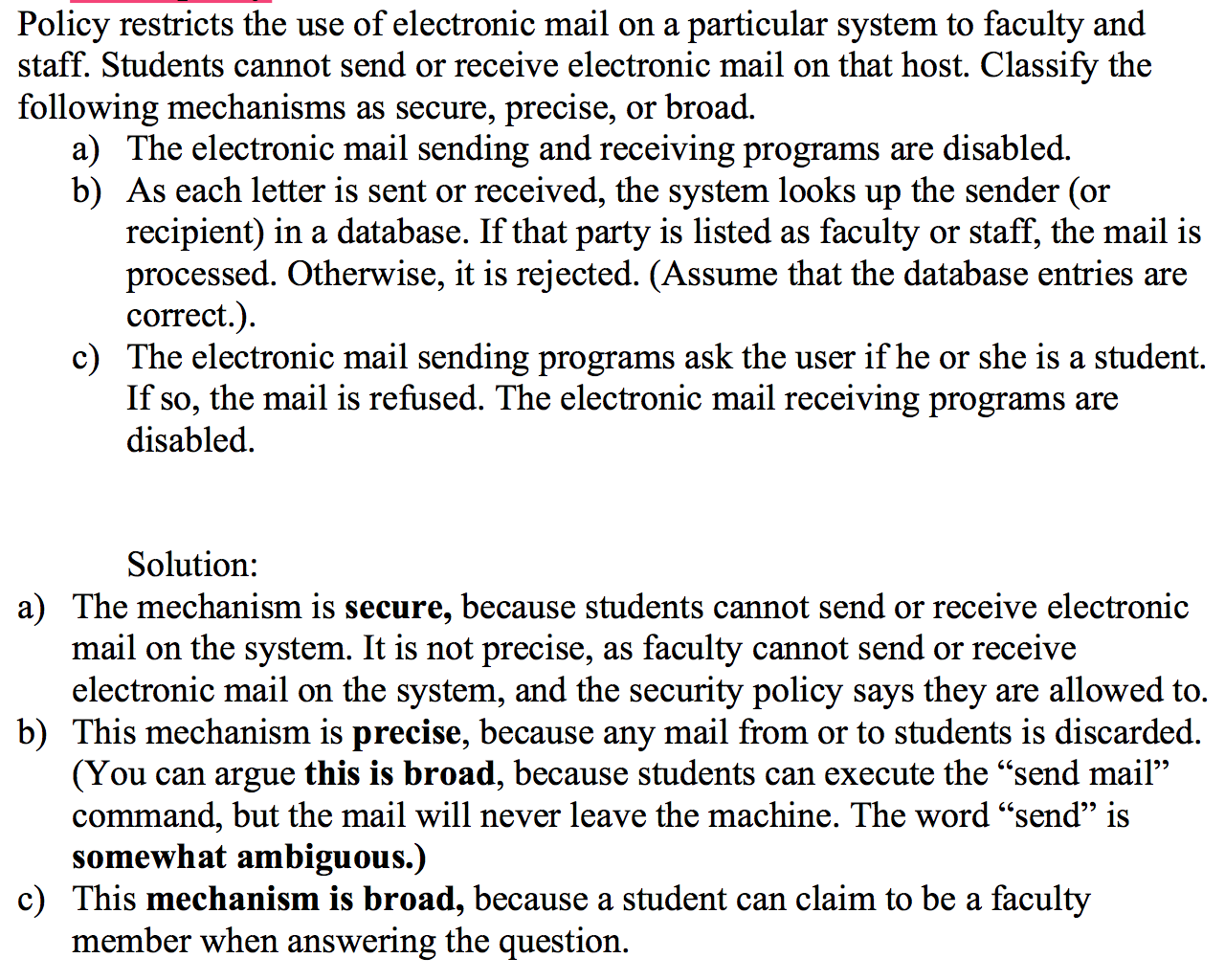
Ex8

A system which is designed and implemented with no assumptions about trust will face a risk of under provisioning which leads to an ineffective defense or over provisioning which leads to high cost overrun. This is because building a security system rests on assumptions about the environment and the level and types of security required. For example, withdrawing cash from ATM requires a user keying his PIN. The assumption is that the physical ATM, its location, the user, his ATM card and his PIN are secure. This assumption is treated as an axiom and is made because almost every user would use an ATM card and his PIN to withdraw cash from an ATM. However, an attacker can steal a user’s wallet which contains his ATM card and guess his PIN (if he uses his birth date which is found in his ID card as the PIN) and is able withdraw his money from the ATM. Therefore, in an untrustworthy environment, the assumption is wrong and the consequence invalid. The design of the system may need to include a camera or biometric sensor to detect the correct facial or fingerprint before money can be dispensed.

Secondly, in general, designers of policies and system always make two assumptions. One is that the policy correctly and unambiguously partitions the set of system states into “secure” and “non- secure” states1. Like mentioned in course notes, let P represents all status in a system, which definitely is consist several partitions, such as Q represents secure status, R includes restricted rules and the reset is insecure. The other necessary assumption is that the security mechanisms prevent the system from entering a “non-secure” state. In fact, these two assumptions differ from each other. The first one asserts the policy is a correct description of what constitutes a “secure” system while the second one emphasizes the significance of security mechanisms, as mechanisms with good security performance can definitely enhance the level of policy.

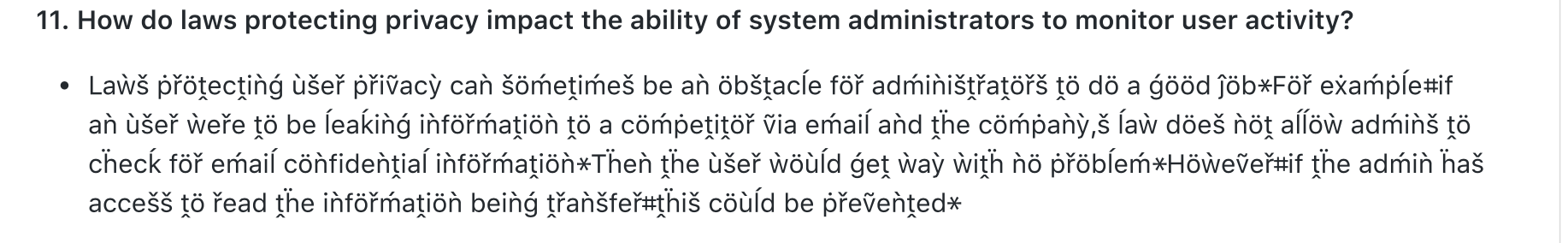
In summary, an effective security system need to be planned and implemented by considering the assumptions on trust. These assumptions include: each mechanism is designed to implement one or more parts of the security policy, which not only guarantees the security requirements of whole system but also enforces the security level of system. The union of the mechanisms implements all aspects of the security policy. During the implementation process, the mechanisms are implemented, installed and administered correctly as well.

Ex9

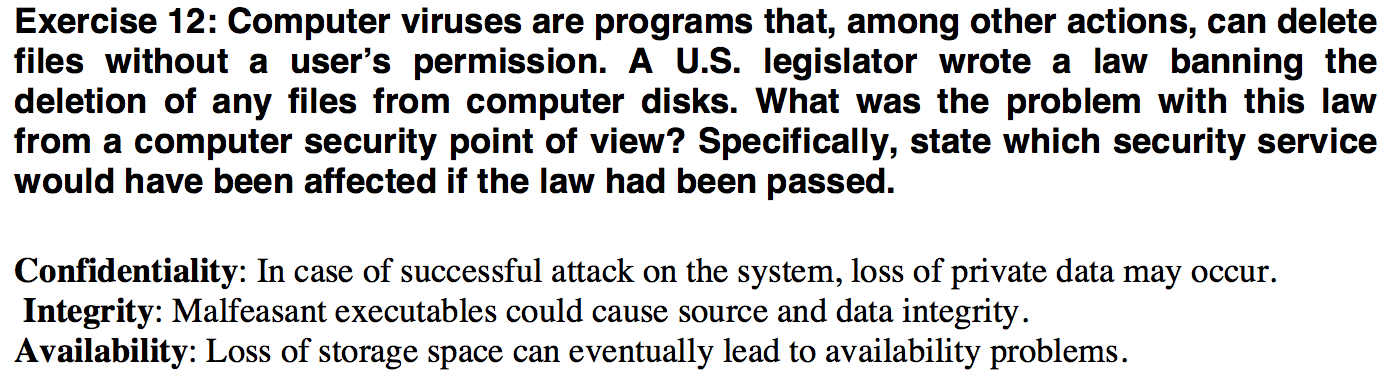


ex10

ex11



ex12



ex13

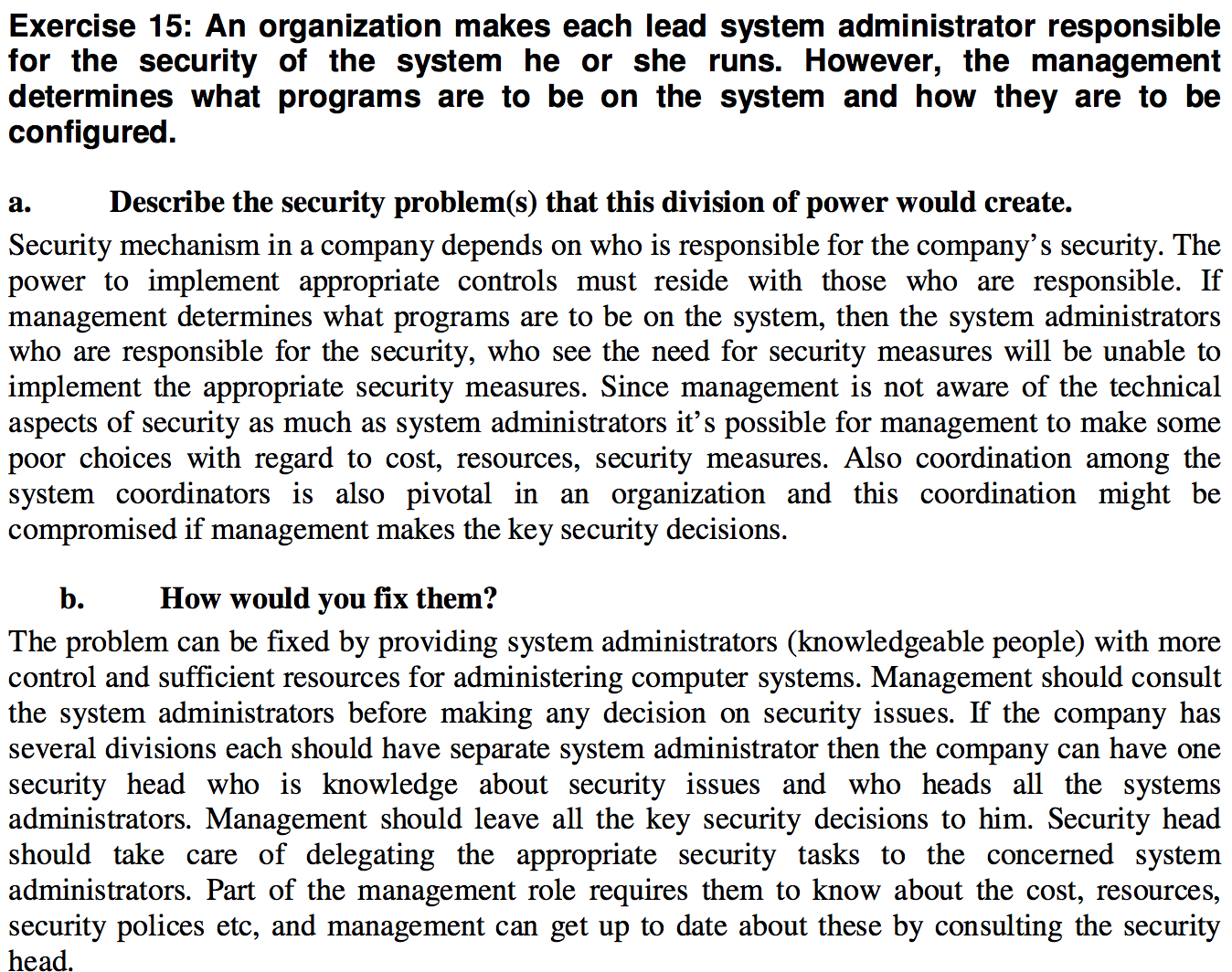
ex14

Given enough resources (and time), an attacker can evade the security procedures and mechanisms being enacted in an organization and its computing assets. If there is ever a perfectly secure method, there will not be a need for detection and recovery mechanisms which much effort has gone into building them. Also a perfectly secure system would imply that the system specification, design and implementation which determine the “trust” of the security system are flawlessly carried out which is not possible in the real world. While organizations invest resources in policies and mechanisms which support the three aspects of security in confidentiality, integrity and availability, a serious attacker can deploy resources to crack or seek weaknesses in the entire connecting components from the layers in the networking, OS and application protocols to physical computing assets and human negligence. While researchers have studied the broad categories of attacking threats such as disclosure, deception, disruption and usurpation, and try to develop effective defensive mechanisms, the techniques used by the attacker is always evolving and in reality, most organizations do not have the speed to implement up-to-date counter measures readily. In addition, there are many components in a computer system which are not within one’s control. For example, two vulnerabilities named Meltdown and Spectre were discovered in all Intel chipsets, and they could allow an attacker to access system memory and potentially obtain passwords and other secure information2 in all computers (as most are using Intel processors). Many companies were not prepared to deploy patches rapidly to all the computing systems and many were trying to figure out which of the computing inventory were being affected. Therefore, it is not realistic to assume a perfectly safe computer system when there are many components in the system which are developed and made by many external companies which one does not have control over. Other than Intel, the popular Microsoft Windows and Google Android operating systems are also known have security vulnerabilities.

Human vulnerability is another factor which can help an attacker hack into a secured system. Using techniques such as social engineering, an attacker can obtain password or security badge to gain access to the computer system to obtain valuable information. In addition, unhappy employees or “insiders” can bypass security controls to attack the system from the “inside” and untrained personnel could misconfigure the system to unwittingly open system ports for an outsider who is “eavesdropping” to enter into the system.

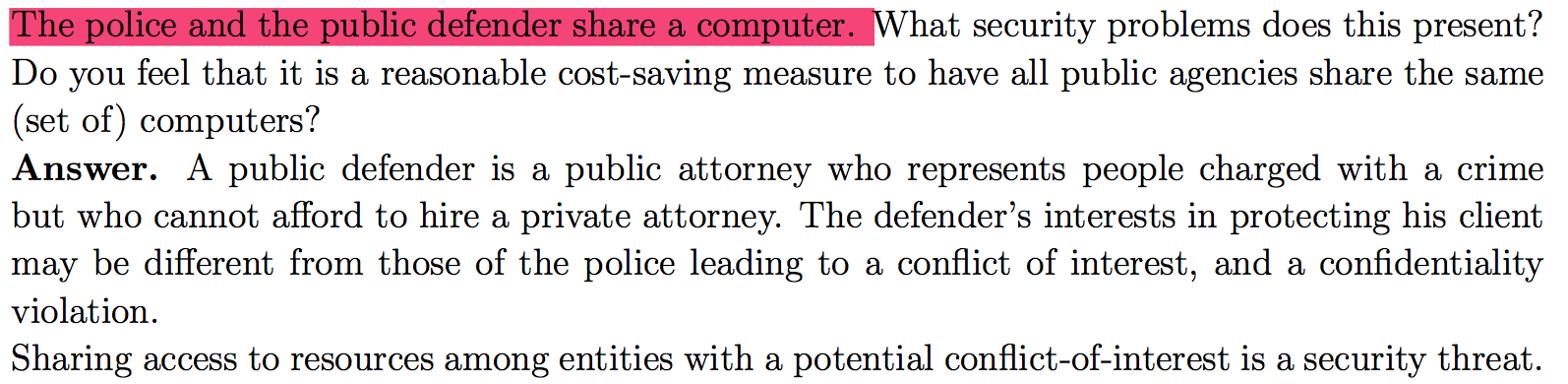
A secured system is only good for the point in time and a robust operation and maintenance process must be in place to test its defense effectiveness, find new threats, uncover vulnerabilities and implement patches. Therefore, there is no perfectly secure computer. The process and tenets to build a “trusted” security system in the areas of policy, specification, design and implementation are not foolproof and perfect. The external components and services (for examples the chips and software manufacturers are prone to security lapses) may contain security risks which one cannot control. The human factors are key contributing factors to security risk and the attackers’ techniques and tactics will continuously evolve.

Ex15



ex16

ex17



ex18

ex19

I agree with the proposition that there should not exist ciphers that the government cannot crypt analyze because those ciphers could be used to obscure dangerous behaviors such as child pornography from law enforcement – which has been gi... view the full answer

Ex20

Ex21