Chapter 17

Multiple Choice Questions

1. What is the difference between mechanisms and policies?
A. Mechanisms determine what will be done, while policies decide when it will be done
B. Mechanisms determine how something will be done, while policies decide what will be done
C. Mechanisms determine how something will be done, while policies decide why something will be done
D. Mechanisms determine what will be done, while policies decide how it will be done

Ans: B
Feedback: 17.1
Difficulty: Easy

2. A protection domain is a collection of access rights, each of which is ___________________

A. a pair <object-name, list-of-users>
B. a pair <object-name, rights-set>
C. a triplet <object-name, user, rights-set>
D. a triplet <object-name, process_id, rights-set>

Ans: B
Feedback: 17.4.1
Difficulty: Medium

3. The ability to copy an access right from one domain to another may be realized as follows

A. A right R is copied from domain A to domain B and R is removed from domain A. The right R could be copied from domain B to another domain.
B. A right R is copied from domain A to domain B, but the right R could not be copied from domain A to another domain.
C. A right R is copied from domain A to domain B, but the right R could not be copied from domain B to another domain.
D. none of the above

Ans: A
Feedback: 17.5
Difficulty: Medium

4. UNIX operating system associates a protection domain with the ____.
A. task
B. tread
C. process
D. user

Ans: D
Section: 17.4.2
Difficulty: Medium

5. The owner right allows ___________

A. addition of new rights only
B. addition of new rights and removal of some rights
C. removal of some rights only
D. none of the above

Ans: B
Feedback: 17.5
Difficulty: Easy

6. Which of the following is an advantage of compiler-based enforcement of access control?
A. Protection schemes are programmed as opposed to simply declared.
B. Protection requirements are dependent of the facilities provided by a particular operating system.
C. The means for enforcement needs to be provided by the designer of the subsystem.
D. Access privileges are closely related to the linguistic concept of a data type.

Ans: D
Section:17.12.1
Difficulty: Hard

7. Which of the following is true of the Java programming language in relation to protection?
A. When a class is loaded, the JVM assigns the class to a protection domain that gives the permissions of that class.
B. It does not support the dynamic loading of untrusted classes over a network.
C. It does not support the execution of mutually distrusting classes within the same JVM.
D. Methods in the calling sequence are not responsible for requests to access a protected resource.

Ans: A
Section: 17.12.2
Difficulty: Medium

8. A capability list for a domain is ____________________

A. a list of operations together with the list of processes allowed to run the operations on those objects.
B. a list of objects together with the list of processes allowed to access those objects.
C. a list of objects together with the operations allowed on those objects.
D. a list of triplet <object, process, rights>.

Ans: C
Section: 17.6.3
Difficulty: Medium

9. Object means __________

A. hardware object or software object
B. process or threat
C. software object only
D. process only

Ans: A
Section: 17.4
Difficulty: Medium

10. What capability is not used by Linux?

A. permitted
B. mapped
C. effective
D. inherited

Ans: B
Section: 17.10.1
Difficulty: Medium

11. ________________ is not a protection mechanism.

A. System Integrity Protection
B. Intrusion Prevention
C. System-Call Filtering
D. Sandboxing

Ans: B
Section: 17.11
Difficulty: Easy

Essay Questions

1. What are the main reasons for implementing a protection subsystem?
Ans: The most obvious is the need to prevent the mischievous, intentional violation of an access restriction by a user. Of more general importance, however, is the need to ensure that each process in a system uses system resources only in ways consistent with stated policies.

Feedback: 17.1
Difficulty: Easy

2. What does compartmentalization mean?
Ans: Compartmentalization is the process of protecting each individual system component through the use of specific permissions and access restrictions.

Feedback: 17.2
Difficulty: Medium

3. Explain the need-to-known principle.
Ans: The need-to-know principle means that at any time, a process should be able to access only those objects that it currently requires to complete its task. This rule is useful in limiting the amount of damage a faulty process or an attacker can cause in the system.

Feedback: 17.2
Difficulty: Medium

4. Describe domain switching.
Ans: The association between processes and domains can be dynamic. Then domain switching is used to implement such a mechanism enabling the process to switch from one domain to another.

Feedback: 17.4.1
Difficulty: Medium

5. What are the main drawbacks of the implementation of the access matrix as a global table?
Ans: The table is usually large and thus cannot be kept in main memory, so additional I/O is needed. Virtual memory techniques are often used for managing this table. In addition, it is difficult to take advantage of special groupings of objects or domains. For example, if everyone can read a particular object, this object must have a separate entry in every domain.

Feedback: 17.6.1
Difficulty: Medium

6. How does a lock-key mechanism work?
Ans: Each object has a list of unique bit patterns called locks. Similarly, each domain has a list of unique bit patterns called keys. A process executing in a domain can access an object only if that domain has a key that matches one of the locks of the object.

Feedback: 17.6.4
Difficulty: Medium

7. Describe the idea of the sandboxing.
Ans: Sandboxing involves running processes in environments that limit what they can do. In a basic system, a process runs with the credentials of the user that started it and has access to all things that the user can access. If run with system privileges such as root, the process can literally do anything on the system. It is almost always the case that a process does not need full user or system privileges.

Feedback: 17.11.3
Difficulty: Medium

8. Describe the idea of SIP (System Integrity Protection).
Ans: SIP restricts access to system files and resources in such a way that even the root user cannot tamper with them. SIP uses extended attributes on files to mark them as restricted and further protects system binaries so that they cannot be debugged or scrutinized, much less tampered with. Most importantly, only code-signed kernel extensions are permitted, and SIP can further be configured to allow only code-signed binaries as well.

Feedback: 17.11.1
Difficulty: Hard

9. Explain a confinement problem.

Ans. The confinement problem is a problem of guaranteeing that no information initially held in an object can migrate outside of its execution environment. The copy and owner rights provide us with a mechanism to limit the propagation of access rights. However, they do not give us the appropriate tools for preventing the propagation (or disclosure) of information.

Feedback: 17.5
Difficulty: Medium

10. Describe how the access matrix is implemented in MULTISC.

Ans. MULTISC uses a combination of access lists and capabilities. When a process first tries to access an object, the access list is searched. If access is denied, an exception condition occurs. Otherwise, a

capability is created and attached to the process. Additional references use the capability to demonstrate swiftly that access is allowed. After the last access, the capability is destroyed.

Feedback: 17.6.5
Difficulty: Hard

11. How does Linux use *system-call filtering*?

Ans. A code can be added to the kernel to perform an inspection at the system-call gate, restricting a caller to a subset of system calls deemed safe or required for that caller’s function. Specific system-call profiles can be constructed for individual processes. The Linux mechanism SECCOMP-BPF uses the Berkeley Packet Filter language to load a custom profile through Linux’s proprietary prctl system call. This filtering can be effectively enforced if called from within a run-time library when it initializes or from within the loader itself before it transfers control to the program’s entry point.

Feedback: 17.11.2
Difficulty: Hard

12. What protection mechanism is used to ensure that operating-system distributions and patches have not be changed?
Ans. It is code signing, which is the digital signing of programs and executables to confirm that they have not been changed since the author created them. It uses a cryptographic hash to test for integrity and authenticity. Code signing is used for operating-system distributions, patches, and third-party tools alike. Some operating systems, including iOS, Windows, and macOS, refuse to run programs that fail their code-signing check

Feedback: 17.11.4
Difficulty: Hard

True/False Questions

1. The kernel should not run with a higher level of privileges than user processes.

Ans: F
Feedback: 17.3
Difficulty: Easy

2. Rings of protection separate functions into domains and order them hierarchically.

Ans: T
Feedback: 17.4
Difficulty: Easy

3. Domains cannot share access rights

Ans: F
Feedback: 17.4.1
Difficulty: Easy

4. Android cannot provide the same level of protection as UNIX, because it is not able to separate users.

Ans: F
Feedback: 17.4.3
Difficulty: Medium

5. The default set of access rights are used if no entry in the access list is found.

Ans: T
Feedback: 17.6.2
Difficulty: Easy

6. In a dynamic protection system, sometimes access rights to objects shared by different users need to be revoked.

Ans: T
Feedback: 17.7
Difficulty: Medium

7. Role-based access control (RBAC) increases the security risk associated with superusers.

Ans: F
Feedback: 17.8
Difficulty: Hard

8. root user can modify mandatory access control (MAC)

Ans: F
Feedback: 17.9
Difficulty: Medium

9. Apple's systems employs capability-based protection in the form of entitlements.

Ans: T
Feedback: 17.10.2
Difficulty: Medium