Homework to be submitted by 21 Sep 22:00 hour

Group 5

Wong Ann Yi (1004000)
Liu Bowen (1004028)
Tan Chin Leong Leonard (1004041)

Exercise 1

Let a password checking program *auth* be a mechanism defined as *auth*(*u, p, d*) = 1 if the pair (*u, p*) ∈ *d*, and *auth(u, p, d)* = 0 otherwise. (u = username, p = password, d = database.) Is it a secure mechanism in the sense of Defintion 4-19 in Bishop's book *c(u, p, d) = (u, p)*? Why or why not?

Answer:

It is secure mechanism because this policy prevents unauthorized disclosure of information.

As a function, *auth*: $U \times P \times D \rightarrow \{1, 0\}$, where *U* is the set of potential user names, *P* is the set of potential passwords and *D* is the databases, *1* and *0* represent true and false, respectively. Then for $u \in U$, $p \in P$, and $d \in D$, *auth*(*u*, *p*, *d*) = *1* if and only if the pair (*u*, *p*) ∈ *d*. Otherwise, the observer will be denied of any information.

Exercise 2

Given the security levels TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED (ordered from highest to lowest), and the categories A, B, and C, specify what type of access (read, write, or both) is allowed in each of the following situations. Assume that discretionary access controls allow anyone access unless otherwise specified.

a) Paul, cleared for (TOP SECRET, {A, C}), wants to access a document classified (SECRET, {A, C}).

Answer:

Using Bell LaPadula rules, our answers are as follows:
Read rights:
Paul can read document iff L(Paul) dom L(document)

As security level: SECRET < TOP SECRET and the categories are: $\{A, C\} \subseteq \{A, C\}$ therefore Paul can read document. This is also known as the "no reads up" rule

Write rights:
Paul can write document iff L(document) dom L(Paul)
As security level: TOP SECRET > SECRET therefore Paul cannot write document. This is also known as the "no writes down" rule.

b) Anna, cleared for (CONFIDENTIAL, {C}), wants to access a document classified (CONFIDENTIAL, {B}).

Answer:

Read rights:
Under Bell LaPadula rules, Anna can read document iff L(Anna) dom L(document)
As security level: CONFIDENTIAL = CONFIDENTIAL but the categories: $\{B\}\not\subset\{C\}$ therefore Anna cannot read document.

Write rights:
Anna can write document iff L(document) dom L(Anna)
As security level: CONFIDENTIAL = CONFIDENTIAL but categories: $\{B\}\not\subset\{C\}$ therefore Anna cannot write document.

c) Jesse, cleared for (SECRET, {B, C}), wants to access a document classified (SECRET, {B, C}).

Answer:

Read rights:

Using Bell LaPadula rules, Jesse can read document iff L(Jesse) dom L(document)
As security level: SECRET = SECRET and categories: $\{B,C\}\subseteq\{B,C\}$ therefore Jesse can read document classified (SECRET, {B, C}).

Write rights:
Jesse can write document iff L(document) dom L(Jesse)
As security level: SECRET = SECRET and categories: $\{B,C\}\subseteq\{B,C\}$.therefore Jesse can write document classified (SECRET, {B, C}).

d) Robin, who has no clearances (and so works at the UNCLASSIFIED level), wants to access a document classified (CONFIDENTIAL, {B}).
Answer:

Using Bell LaPadula rules,

Read rights:

Robin can read document iff L(Robin) dom L(document)

As security level: UNCLASSIFIED ≤ CONFIDENTIAL, therefore Robin cannot read document classified (CONFIDENTIAL, {B}).

Write rights:
Robin can write document iff L(document) dom L(Robin)

As security level: UNCLASSIFIED ≤ CONFIDENTIAL and Φ ⊆ {B} therefore Robin can write document classified (CONFIDENTIAL, {B}).

The integrity level of subjects will remain unchanged if we elevate the object levels to be above the subject's level instead of lowering the subject's integrity level.

Exercise 3

Give an example that demonstrates the integrity level of subjects decreases in Biba's low-water-mark policy. Under what conditions will the integrity level remain unchanged?
Answer:

The example will use the read function under the Biba's low-water-mark policy.

Assuming that there is a set of integrity level {ISP (System Program), IO(Operational), ISL(System Low)} (from the highest to lowest) in system. Assuming that there are two integrity categories {ID (Development), IP(Production)}.

The subject-to-integrity and object-to-integrity tables we designed are as follow:

Subject	Integrity levels (for read function only)
Ordinary users	(ISL, { IP })
Programmers	(IO, { ID})
System manager	(ISP, { IP, ID })

Object	Integrity levels
Development code	(ISL, { ID })
Production code	(IO, { IP })
System programs	(ISP, { IP, ID })

As shown in tables, After system manager read production code, the integrity level of system manager will be altered into IO as i'(s) = min(i(System manager), i(Production code)). Similarly, the integrity level of system manager will become ISL after he reads the development code, which results in integrity level of subjects decreases in Biba's low-water-mark policy. This example illustrates that the integrity level will decrease when using Biba's rule.

If the integrity level remains unchanged, the subject S with the integrity level i(s) can only read the objects whose integrity level are no less than i(s). In our table above, the programmers can only read production code or system programs.

Exercise 4

Suppose a system used the same labels for integrity levels and categories as for security levels and categories. Under what conditions could one subject read an object? Write to an object?

In the Bell-LaPaula security model, the labels for security levels of the subjects are: Top Secret, Secret, Confidential and Unclassified. The categories of the objects are also labeled the same.

In order for a subject (s) to read an object (o), the condition must be: iff L(s) dom L(o) and s has the permission to read o. In addition, the "no reads up" rule applies.

In order for the subject to write an object, the condition must be: iff L(o) dom L(s) and s has the permission to write o. In addition, the "no writes down" rule applies.

In the Biba's integrity model, the labels for the integrity level are: Highly trusted, Medium trusted, Low trust. The categories of the objects are also labeled the same.

In order for a subject (s) to read an object (o). the condition must be: s can read o if and only if $i(s) \leqslant i(o)$ and follows a "no reads down" rule.

In order for the subject to write an object, the condition must be: s can write to o if and only if $i(o) \leqslant i(s)$ and follows a "no writes up" rule.

The rules are shown in table:

	Bell-LaPaula security	Biba integrity
subject s read object o	iff L(s) dom L(o)	$i(s) \leqslant i(o)$
subject s write object o	iff L(o) dom L(s)	$i(o) \leq i(s)$

1) Not secure as there is information leakage. The attacker can create a list of incorrect username and password pairs whenever the program returns 0 as output. Also, the password checker checks validity of username first, followed by password and finally the existence of the username and password pair in the database. Hence if username is invalid, the program will output 0 immediately. If username is correct and password is wrong, the program will output 0 after a delay. Therefore, an attacker can realise whether the username or password is in the database from the time taken for the output to appear.

3) The integrity level of subjects will remain unchanged if we elevate the object levels to be above the subject's level instead of lowering the subject's integrity level.