Week 2 Homework due on Friday September 28, 6:59 PM

Group 5

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

Exercise 1

Prove Theorem 4-1 of Bishop's. Show all elements of your proof.

Theorem 4-1. Let m_1 and m_2 be secure protection mechanisms for a program p and policy c. Then $m_1 \cup m_2$ is also a secure protection mechanism for p and c. Furthermore, $m_1 \cup m_2 \approx m_1$ and $m_1 \cup m_2 \approx m_2$.

Answer:

Theorem 4-1 states that if m_1 and m_2 are secure protection mechanisms for a program p and they conform to policy c, then the combination or union of these two mechanisms is also a secure mechanism for program p and still conform to policy c. We will prove it by the below definitions.

For m_1 and m_2 to be secure for p and conform to c, then by definition 4-20, m_1 is as precise as m_2 and for all inputs $(i_1, \ldots, i_n)$ both mechanisms will yield $p(i_1, \ldots, i_n)$.

From definition 4-21, the union of m_1 and m_2 which we will represent by $m_3 = m_1 \cup m_2$ states that:

$m_3(i_1, \ldots, i_n) = p(i_1, \ldots, i_n)$ when $m_1(i_1, \ldots, i_n) = p(i_1, \ldots, i_n)$ OR $m_2(i_1, \ldots, i_n) = p(i_1, \ldots, i_n)$, otherwise, the combined mechanism will return the same value as m_1 which is a secure protection mechanism.

Therefore, the above definitions prove the validity of Theorem 4-1 and that the union (or combination) of two secure protection mechanisms which comply with control policy c will still be a secure mechanism for the program. Furthermore, the union of the two mechanisms are as precise as each of mechanism when it is standalone.

Exercise 2

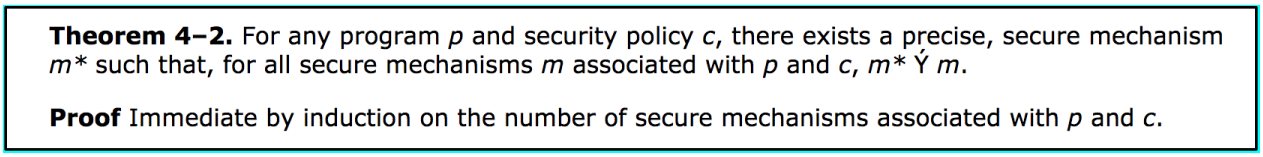
Expand the proof of Theorem 4-2 of Bishop's to show the statement, and the proof, of the induction.

Theorem 4-2. For any program p and security policy c, there exists a precise, secure mechanism m* such that, for all secure mechanisms m associated with p and c, m* ≈_m.

Answer:

From Theorem 4-2, the ideal m* is a protection mechanism which is secure for program p and conform precisely to the control policy c. As stated in definition 4-20, two distinct protection mechanisms are precise to each other if the similar inputs yield similar outputs for program p while conforming precisely with policy c. We will show by induction, that any mechanism in the system which is secure for p and conform to c will be as similarly precise with m*.

According to Bishop, the induction process would prove that a number of secure mechanisms associated with p and c will be similarly precise as m* as shown in Figure 1.



Theorem 4–2. For any program *p* and security policy *c*, there exists a precise, secure mechanism *m** such that, for all secure mechanisms *m* associated with *p* and *c*, *m** Ý *m*.

Proof Immediate by induction on the number of secure mechanisms associated with *p* and *c*.

Figure 1: Extracted from Computer Security: Art and Science by Matt Bishop - Chapter 4.7 Security and Precision

The proof by mathematical induction is as follows:

1) Base step: n = 1 where n represents there is only one precise, secure mechanism associated with p and c. Obviously, $m^* \approx m_1$ as m* equals m_1.
2) Assuming $m^* \approx m$ where m represents (m_1, …, m_k) for k number of precise and secure mechanisms associated with program p and confidentiality policy of c.
3) For the k+1 mechanisms, (m_1, ….., m_k) $\cup$ $m_{k+1} \approx m^*$ according to the Theorem 4-1(proved in exercise 1) for the number of k+1 mechanisms associated with p and c.
   By mathematical induction, Theorem 4-2 has been proved to be valid.

Exercise 3 3a) Mention why writing to the virus prevention region is dangerous.

In the DG/UX system, why is the virus prevention region below the user region? Why is the administrative region above the user region?

Answer:

In the DG/UX system, the virus prevention region is below the user region. This is to prevent user processes or programs from writing to them (where site executables are). However, users can execute the programs and therefore read and execute are allowed ("read" usually includes "execute")[1] but not write. The Bell-LaPadula security model does not allow write down.

In the DG/UX system, the administrative region is above the user region. This is to prevent user processes and programs from reading information at the administrative level. The information is sensitive and contains logs, MAC label, Authorization and Authentication database and others

[1] Computer Security: Art and Science by Matt Bishop, Chapter 5.2. The Bell-LaPadula Model

which users should not be allowed to read. The Bell-LaPadula security model does not allow read up for lower level subjects.

Exercise 4

Declassification effectively violates the *-property of the Bell-LaPadula Model. Would raising the classification of an object violate any properties of the model? Why or why not?

Answer:
The principle of tranquility states that subjects and objects may not change their security levels once they have been instantiated[2]. Declassification violates the *-property of the Bell-LaPadula Model which states that S can write O if and only if $l_s \leq l_o$ and S has discretionary write over O and in other words it is "no writes down". A declassification makes information available to subjects who did not have access to it before and rendering a "high" subject unable to write on the "low" object. However, the principle of weak tranquility does provide flexibility by allowing "trusted entities" to declassify information by removing sensitive elements. For example, the UK government declassified two (of the many) papers written by Alan Turing in breaking the Enigma code after more than 70 years since the end of the second world war[3].

Raising the classification of an object is not a violation as the Bell-LaPadula model as the model does not state how to classify the information but instead describes how to manage the information based on its classification. Raising the classification enforces the control of "no reading up" for the lower level subjects. However, the information which has been previously accessed by subjects of that security level is no longer "confidential" despite raising its classification as the information has already been accessed. Therefore, raising the classification of an object does not violate the Bell-LaPedula model as the confidentiality of the object is not valid.

[2] Computer Security: Art and Science by Matt Bishop, Chapter 5.3. Tranquility
[3] Turing's Nazi Enigma Code-Breaking Secrets Have Been Declassified - https://gizmodo.com/5904253/turings-nazi-enigma-code-breaking-secrets-have-been-declassified

Exercise 5

Prove Theorem 6-1 of Bishop's for the strict integrity policy of Biba's model.

Theorem 6-1. If there is an information transfer path from object $o_1 \in O$ to object $o_{n+1} \in O$, then enforcement of the low-water-mark policy requires that $i(o_{n+1}) \leq i(o_1)$ for all $n > 1$.

Answer:

The notion of an information transfer path is defined as[4]:

Definition 6-1: An information transfer path is a sequence of objects $o_1, \ldots, o_{n+1}$ and a corresponding sequence of subjects $s_1, \ldots, s_n$ such that s_i reads o_i and s_i writes o_{i+1} for all i, $1 \leq i \leq n$.

If there is an information transfer path from object $o_1 \in O$ to object $o_{n+1} \in O$, we assume that every read and write action should be conducted as per the sequence in *Definition 6-1*. According to the rules of Strict Integrity Policy (class note page 33) in Biba's model:

Step1:

For $s_n \in S$ can read $o_n \in O$ if and only if $i(s_n) \leq i(o_n)$ which is also known as "no reads down". The property states that a subject can read an object only if the integrity level of the subject is less than the integrity level of the object.

For $s_n \in S$ can write $o_{n+1} \in O$ if and only if $i(o_{n+1}) \leq i(s_n)$ which is also known as "no writes up". This property states that a subject can write to an object only if the object's integrity level is less than or equal to the subject's level.

According to definition 6-1, information transfer path rules that s_i reads o_i and s_i writes o_{i+1}, the above-inequality can be derived from the definition of the above Strict Integrity Policy.
Combining the above two inequality statements, and by transitivity therefore, $i(o_{n+1}) \leq i(o_n)$.

Step2:

Applying the Strict Integrity Policy again,

For $s_{n-1} \in S$ can read $o_{n-1} \in O$ if and only if $i(s_{n-1}) \leq i(o_{n-1})$ known as "no reads down".
For $s_{n-1} \in S$ can write $o_n \in O$ if and only if $i(o_n) \leq i(s_{n-1})$ known as "no writes up".

Again, by transitivity, therefore subject s_{n-1}, $i(o_n) \leq i(o_{n-1})$.

[4] Computer Security: Art and Science (Matt Bishop) – Chapter 6.2. Biba Integrity Model

Step3:
By using mathematical induction: for all n > 1 the integrity requirements should be $i(o_{n+1}) \leq i(o_n) \leq \ldots \leq i(o_1)$. Therefore, the Theorem 6–1 has been proved!

Exercise 6

Explain why the system controllers in Lipner's model need a clearance of (SL, {D, PC, PD, SD, T}).

Answer:

The Lipner's model leverages two components: Bell-LaPadula security model and Biba integrity model. As such, the Lipner's model consists of two security clearance: AM, SL (ordered from highest to lowest) and five categorie: D, PC, PD, SD, T. In Lipner's model, the system controllers have clearance as follows: (SL, {D, PC, PD, SD, T}).

According to the rules of Bell-LaPadula security model, subject S can read an object O if and only S dom O (or no reads up) while subject S can write to O if and only if O dom S (or no writes down).

For read right, system controllers have SL security level in all the categories[5] because they have the responsibility to monitor, control or govern the processes and data in the system. Therefore, system controllers have to have read access to all objects, including development code, production code/data, tools and system programs.

System controllers have no write access to development code, production code/data, tools and system programs, preventing above-mentioned objects from modification by the system controllers. However, the system controllers have downgrade privilege because they need to install code after it is certified for production so that other parties cannot write to it[6].

[5] Week-2 notes: page 37
[6] Computer Security: Art and Science (Matt Bishop) – 6.3. Lipner's Integrity Matrix Model