

I. Introduction

1. Security Engineering

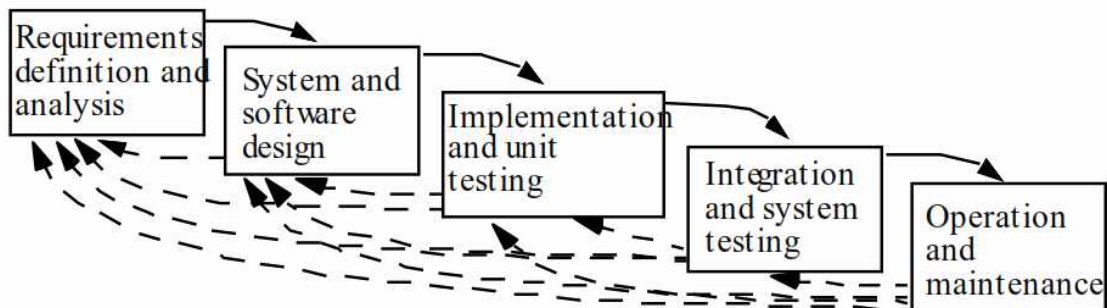
- 보안의 관점으로 dependable 한 시스템을 만드는 일련의 과정
(요구사항 분석 - 설계 - 제작 - 테스트 - 릴리즈)
- Dependability = Security(intended failure) + Reliability(accidental failure)
- 목표 : 가능한 많은 보안 취약점을 찾아, 제거하는 것
→ 보안결점을 빨리 찾아야 Cost가 내려감

2. Policy - Assurance - Mechanism의 관계

Policy	Assurance	Mechanism
· Security Policy : CIA 관점에서 우리가 무엇을 해야하고, 하지말아야 할지 또 그것을 어떻게 이룰것인지 요구사항을 기술한 문서 → 비인가 사용으로부터 식별된 Asset을 보호하는 것을 기술하는 것	· 구현된 Mechanism이 Policy에 맞게 구현되어 있는지 Justification 하는 것	· Policy를 구현한 것

↓ (Assurance의 종류)

Stage	내 용
Policy Assurance	· 정책이, 일관적이고, 완전하고, 기술적으로 타당한지 Justification 하는 것
Design Assurance	· Design 이 Security Policy에 적합한지 확인
Implementation Assurance	· Implementation이 Security Policy 에 적합한지 확인
Operational Assurance	· 시스템 설치, 협상관리, 동작등 릴리즈 이후 Security Policy를 따르는지 확인



3. Risk의 식별

- Risk = Expectation Asset Loss × Vulnerability × Threat
 - Asset : 식별되고 가치 있는 것들
 - Vulnerability : 의도적이든 비의도적이든 자산을 손상할 수 있는 시스템의 약점
 - Threat : 취약점을 이용해서, 자산에 손상을 줄 수 있는 Adversary의 행동

• ALE(Average Loss Expectation = Loss 확률 × 잠재적 총 Loss

• Risk 분석 방법

구 분	내 용
정량적 분석	<ul style="list-style-type: none"> · 수학적 이론에 기반 · Risk는 input 의 질에 달려 있음 · 언제나 모든 경우에 해당되는 것은 아님
정성적 분석	<ul style="list-style-type: none"> · 전문가의 판단에 맡김 · 더 많은 곳에 적용이 가능하다.

• DREAD Model (Risk 분석 모델)

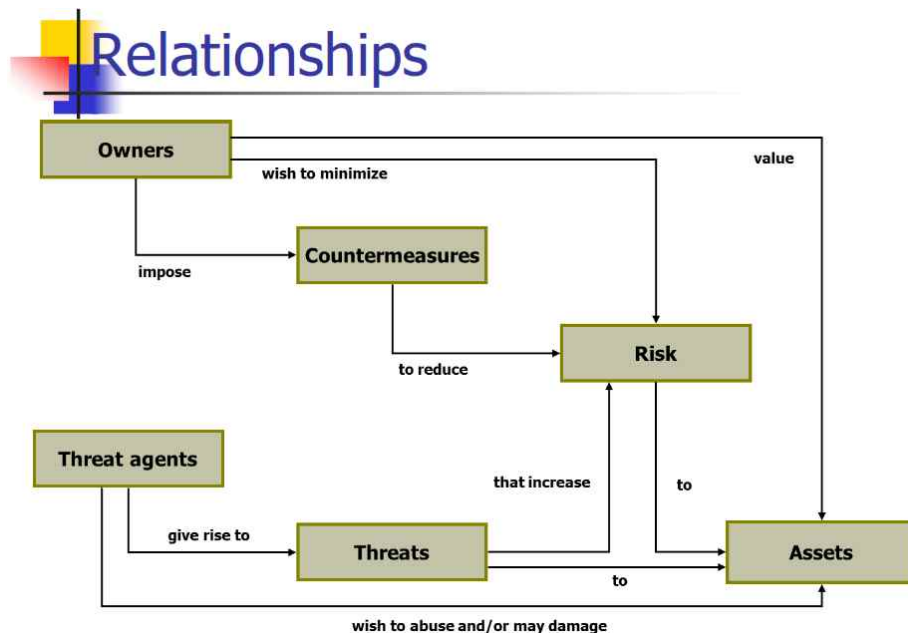
구 분	내 용
Damage potential	· 내가 얼마만큼 당할지 : exploit 이 성공하면 결과는?
Reproducibility	· 얼마의 주기로 당하는지, 어떤 환경에서 당하는지 확인
Exploitability	· 해커의 Skill이 얼마만큼 전문적인지?
Affected User	· 얼마나 많은 User가 피해를 볼 것인지?
Discoverability	· 이 취약점을 누구든 알 수 있는지?

MS's DREAD Risk Analysis Model

	High (3)	Medium (2)	Low (1)
Damage potential	Attacker can retrieve extremely sensitive data and corrupt or destroy data	Attacker can retrieve sensitive data but do little else	Attacker can only retrieve data that has little or no potential for harm
Reproducibility	Works every time; does not require a timing window	Timing-dependent; works only within a time window	Rarely works
Exploitability	Bart Simpson could do it	Attacker must be somewhat knowledgeable and skilled	Attacker must be VERY knowledgeable and skilled
Affected users	Most or all users	Some users	Few if any users
Discoverability	Attacker can easily discover the vulnerability	Attacker might discover the vulnerability	Attacker will have to dig to discover the vulnerability

- Risk를 줄이는 방법
 - Accept Risk : Risk가 감당할 만큼 작은 경우
 - Avoid Risk : 취약점을 제거 함
 - Limit Risk : 위험이 작아지도록 통제
 - Transfer Risk : 보험가입처럼 위험을 다른 곳에 전가함
- Risk 의 Countermeasure는 지속적으로 변한다 → 환경, 경영방식이 변하기 때문
하지만 최소한으로 해야하는 것이 있다.

※ Baseline Protection : 전형적인 Case들에 대한 추천 대응책



- Threat을 분석하는 방법
 - 1) Threat List 작성
 - 2) Threat Tree(Attack Tree 작성)
 - 3) MS 의 STRIDE 모델 (Threat Category)을 이용

구 분	내 용
Spoofing	· 남인척해서 System에 접근
Tampering	· 조작하는 것
Repudiation	· 이벤트를 발생하고 안했다고 하는 것
Information Disclosure	· 정보 노출
Denial of Service	· 서비스 거부
Elevation of Privilege	· 권한상승 : SNS를 통해서 PW를 알아내는 것

II. Definition

1. Security란 무엇인가?

- 자산을 Prevention, Protection, Reaction(recovery/restore asset)을 통해서 지켜내는 것
 - Prevention : 자산을 위협으로부터 지켜내는 것을 측정
 - Detection : 언제, 어떻게, 누구에 의해서 자산이 위협에 처하는지 측정
 - Reaction : 자산을 recover 하거나 위협으로부터 벗어나게 하는 것
- ==> Prevention 에 투자할수록 prevention이 잘 되도록 detection 부분에 투자를 해야한다.

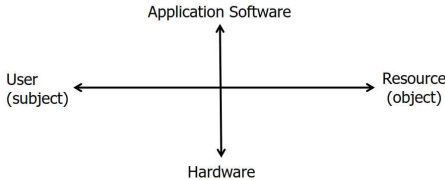

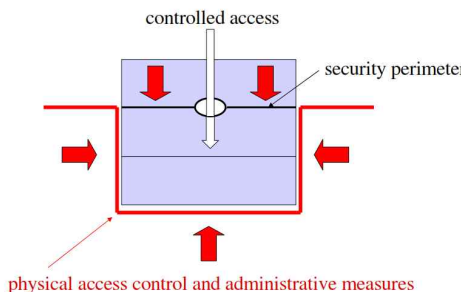
2. 보안의 3가지 목표

구 분	내 용
Confidentiality	· 비인가자에게 정보의 노출을 방지하는 것
Integrity	· 비인가된 정보의 변경을 방지
Availability	· 인가된 사용자가 원할 때 언제든지 시스템 서비스를 이용

3. 보안의 3가지 목표의 Others

구 분	내 용
Accountability	· 책임추궁성, Audit log는 책임을 확인하기 위해서 반드시 보호되어야 함
Non-repudiation	· 시작, 배달, 목적에 대한 부인방지를 해야함.
Dependability	· Reliability + Security
Survivability	· 중대한 결함에 대한 System Failure를 다룸
Authenticity	· 누구랑 얘기하는 것을 아는 것

4. Fundamental Design Parameters

구 분	내 용
1. 어디에 보호중점을 두어야 하는가?	 <p>data, operation, or users?</p>
2. 어떤 레이어에 Security Control을 놓을 것인가? (The MAN-MACHINE SCALE) The Man-Machine Scale	
3. Simplicity 에 둘 것인가 Security에 둘 것인가? Complexity vs Assurance??	<p>· Assurance의 요건</p> <ul style="list-style-type: none"> - 충분성(Sufficiency) : security function은 정확히 시스템에서 운영되는가? - 정확성(Correctness) : security function은 임의로 지나칠 수 없다. <p>· 이 점은 항상 컴퓨터 Security의 딜레마가 된다.</p>
4. 보안통제를 중앙에서 할 것인가? 개별적으로 할 것인가? Contralized vs Decentralized??	
5. 만약에 해커가 Security Boundary를 지나 Layer Below 로 침해를 한다면 어떻게 막을 것인가?	<p>· Security Parameter = Security Boundary</p> <p>Access to the Layer Below</p> 

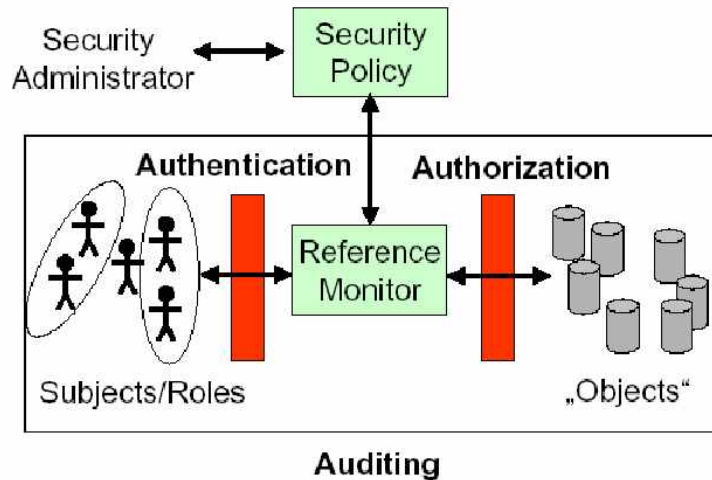
III. Identification & Authentication

1. Authentication vs Authorization

- Authentication : Identity를 증명하는 방법(=verification)
- Authorization : 인가된 사용자에게 레벨에 맞는 자원을 접근토록 하는 것

2. Reference Monitor : Access Control Concept

- Security Kernel : Reference Monitor를 구현해 놓은 것
- TCB : Security Kernel 외에 보안 메커니즘을 구현한 것



3. Identification : ID를 찾는 것 (1:N)

4. Verification : ID를 가지고 있는 사람이 그 사람이 맞는지 확인하는 것 (1:1)

- 방법 : Some we know, have, are, behave...

2 가지 이상하면 multifactor authentication

5. Verification을 구현하는 방법 2가지

- Password
 - 구현이 쉽다. 하지만 노출의 문제가 존재한다 → 암호화 및 Access Control 로 가능
 - 기본적인 Password defence 방법
 - 로그인 횟수 제한, PW checker(generator) 사용, PW aging(timer) 사용
 - 가능한 공격
 - 가) Guessing Attack
 - 전수조사 : 입력횟수 제한하거나, Hash화 저장한다.

- 사전공격 : Hash 저장내용도 Dictionary에 저장이 가능함. 그러므로
 - large salt values : salt(nonce)를 덧붙여 hash 화
→ 동시에 multiple user에 대한 공격을 방지
 - key stretching algorithm : salt 붙인 hash를 여러 번 수행
- ⇒ 결국에는 Dictionary attack에서 찾는 시간을 줄인다.

나) Replay Attack

- Man-in-the-middle attack 같이, 지난 번에 썰던 것 다시 쓸 수 있다.
- Timer(Time 방식, Counter 방식)를 사용하여, 다시 못쓰게 한다.

다) Sniffing(Snooping)

- 몰래 정보를 엿본다. 암호화해서 해결 가능

• Biometrics

- 사람마다 가지고 있는 신체적 특징이나 행동을 이용한다., 단 error rate 존재
- 두 가지 Error
 - FRR : 인가자 맞는데 틀리다고 하는 것(Type I)
 - FAR : 비인가자 맞는데 맞다고 하는 것(Type II)
 - EER(Equal Error Rate) : $FRR = FAR$
- 갖추어야 할 Design Property
 - 보편성, 유일성, 지속성, 수집용이성, 주변환경에 대한 적응성, 수용성(거부감이 없어야 한다.)

IV. Authorization

1. Access Control Security Model

- Security Policy를 formalizing 해 놓은 것
- Lattice 모형
 - 정의 : Partial Order + Great Lower Bound + Least Upper Bound
 - Patial Order 란?

Def :

1. (reflexive closure of R on A)

R_r = the smallest reflexive relation containing R .

$$R_r = R \cup \{ (a, a) \mid a \in A, (a, a) \notin R \}$$

2. (symmetric closure of R on A)

R_s = the smallest symmetric relation containing R .

$$R_s = R \cup \{ (b, a) \mid (a, b) \in R \text{ and } (b, a) \notin R \}$$

3. (transitive closure of R on A) (後面再詳細說明)

R_t = the smallest transitive relation containing R .

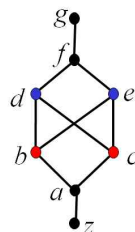
$$R_t = R \cup \{ (a, c) \mid (a, b) \in R_t \text{ and } (b, c) \in R_t \text{ but } (a, c) \notin R_t \} \text{ (repeat)}$$

- Great Lower Bound + Least Upper Bound

Let A be a subset of a poset (S, \preceq) . An element x is called the **least upper bound** of A if x is an upper bound of A and $x \preceq z$ whenever z is an upper bound of A .

Let A be a subset of a poset (S, \preceq) . An element x is called the **greatest lower bound** of A if x is a lower bound of A and $y \preceq x$ whenever y is a lower bound of A .

Ex



$$A_1 = \{d, e\}, A_2 = \{b, c\}$$

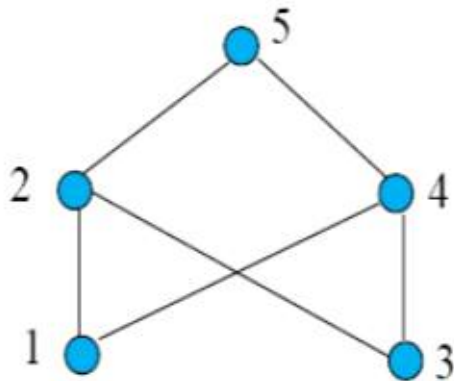
least upper bound of $A_1 = f$

A_1 has no greatest lower bound

A_2 has no least upper bound

greatest lower bound of $A_2 = a$

- Lattice 문제를 풀어보자

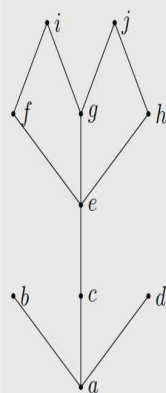


Consider the elements 1 and 3.

- Upper bounds of 1 are 1, 2, 4 and 5.
- Upper bounds of 3 are 3, 2, 4 and 5.
- 2, 4 and 5 are upper bounds for the pair 1 and 3..
- There is no lub since
 - 2 is not related to 4
 - 4 is not related to 2
 - 2 and 4 are both related to 5.
- There is no glb either.

The poset is not a lattice.

Example



Minimal/Maximal elements?

- Minimal & Minimum Element: a .
- Maximal Elements: b, d, i, j .

Bounds, glb, lub of $\{c, e\}$?

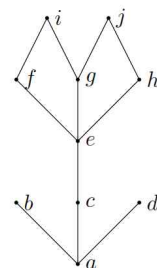
- Lower Bounds: $\{a, c\}$, thus glb is c .
- Upper Bounds: $\{e, f, g, h, i, j\}$ thus lub is e

Bounds, glb, lub of $\{b, i\}$?

- Lower Bounds: $\{a\}$, thus glb is a .
- Upper Bounds: \emptyset , thus lub DNE.

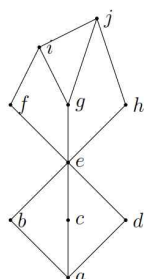
(2012-2 중간 기출!)

Is the example from before a lattice?



No, since the pair (b, c) do not have a least upper bound.

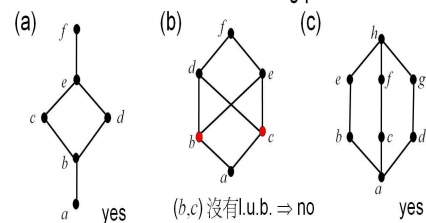
What if we modified it as follows?



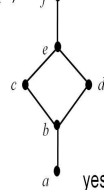
Yes, it is now a lattice, since for any pair, there is a lub & glb

Example 21.

Determine whether the following posets are lattices.

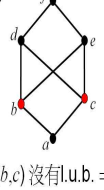


(a)



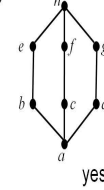
yes

(b)



(b, c) 沒有 l.u.b. \Rightarrow no

(c)



yes

2. DAC vs MAC

구분	DAC	MAC
특징	<ul style="list-style-type: none"> · 자원의 접근을 사용자 계정에 기반 (Need to know) · 사용자는 자원과 관련된 ACL이 변경되어 권한을 부여받는다. · Data Owner가 누가 그 데이터에 접근할 것인지 결정한다. 	<ul style="list-style-type: none"> · 자원에 대한 접근권한을 관리자가 부여한다. · 자원에 대한 접근은 사용자의 보안등급과 주어진 동안에 대상의 보안레벨에 기반 · Multi-level Security : 보안등급을 계층화 해서 나누어 놓은 것 <ul style="list-style-type: none"> → 최초 Formalizing : BLP Model (No read up, No write down)
제한사항	<ul style="list-style-type: none"> · Identity 도용시 자료 유출이 가능하다. (A가 B, C에 권한을 주면, B, C가 다른 이에게 자료를 줄 수 있다.) 	<ul style="list-style-type: none"> · 객체 단위의 세밀한 권한 설정 불가 · Security Level 개수 : $m \cdot (2^n)$ <ul style="list-style-type: none"> - Security level : m - Categories : n

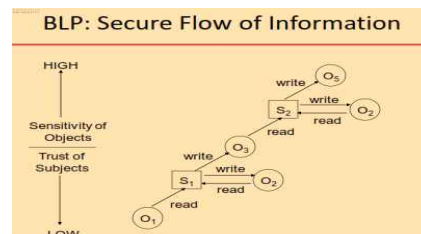
- Separation of Privilege : A system should not grant permission based on single condition.
- Least Privilege : A subject should be given only those privileges that it needs in order to complete its task

3. Bell-Lapadulla Model

- 최초의 MLS을 Formalizing한 모델 → 최초 수학적 증명
- Only Confidentiality 만 언급함
 - : No information flow from high to low security level for confidentiality

■ The State Set

- A state : (b, M, f) , includes
 - Access operations currently in use b
 - List of tuples (s, o, a) , $s \in S$, $o \in O$, $a \in A$.
 - Access permission matrix
 - $M = (M_{s,o})_{s \in S, o \in O}$, where $M_{s,o} \subset A$
 - Clearance and classification $f = (f_S, f_C, f_O)$
 - $f_S : S \rightarrow L$ maximal security level of a subject
 - $f_C : S \rightarrow L$ current security level of a subject ($f_C \leq f_S$)
 - $f_O : O \rightarrow L$ classification of an object



- 2 가지 Property

Simple Security Property (No Read Up)	<ul style="list-style-type: none"> ● A state (b, M, f) satisfies the SS-property if <ul style="list-style-type: none"> ● $\forall (s, o, a) \in b$, such that $a \in \{\text{read}, \text{write}\}$ ● $f_O(o) \leq f_S(s)$ ● I.e. a subject can only observe objects of lower classification <p>Subject는 단지 자신보다 동급이거나 낮은 레벨의 Object를 읽을 수 있다.</p>
* - Property (No Write Down)	<ul style="list-style-type: none"> ■ *-Property (Star-Property) ● A state (b, M, f) satisfies the *-property if <ul style="list-style-type: none"> ● $\forall (s, o, a) \in b$, such that $a \in \{\text{append}, \text{write}\}$ ● $f_C(s) \leq f_O(o)$ ● and <ul style="list-style-type: none"> ● if $\exists (s, o, a) \in b$ where $a \in \{\text{append}, \text{write}\}$, ● then $\forall o', a' \in \{\text{read}, \text{write}\}$, such that $(s, o', a') \in b$ ● $f_O(o') \leq f_O(o)$ <p>Subject는 자신보다 동급이거나 높은 레벨의 Object에 쓸 수 있다. 그리고, Writing 중에는 그 Object level 보다 높은 레벨을 읽을 수 없다.</p>

장점 for BLP	단점 for BLP
<ul style="list-style-type: none"> - 최초로 Formalizing 된 것이다. - 여기서 언급된 state machine model 로 다른 모델을 언급 가능하다 : Biba - 	<ul style="list-style-type: none"> - 기밀성 위주 초점(Not Integrity) - Change access right에 대한 언급이 없다. - Covert Channel이 존재 : Information Flow는 Security Mechanism 으로 컨트롤이 안됨

• McLean 이 주장하는 것 : Subject와 Object의 Security level을 모두 다운을 시키자. 그러면 모든 사람은 모든 Object에 접근할 수 있는 것 아니냐? 가장 낮은 등급에 있는 사람이 Object 등급을 낮추어서 비밀을 Overwrite 할 수 있다. Covert Channel이 존재한다.!!!

※ Covert Channel : 시스템 설계자가 알지 못하는 방법으로 정보가 누출되는 것.

그래서 암호화 하기도 한다.!!

→ Bell : 아니!!! System 전제조건에 그런 것 없었다.

전제조건이 Tranquility를 유지하는 것이다.

※ Tranquility : Security level과 Access Right는 절대 변하지 않는다!!!!

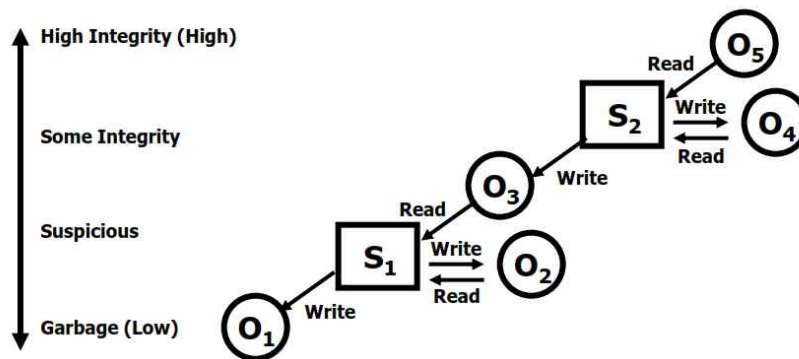
4. Biba Model

- Only Concern Integrity
- 정보의 Flow는 아래 방향으로만 가능하다

- ▶ Similar to Bell-LaPadula in several ways
 - Based on multilevel security with a partial ordering
 - Based on subjects and objects
- ▶ Subjects and objects mapped to integrity levels forming a lattice
 - $f_S: S \rightarrow L$ subject integrity level
 - $f_O: O \rightarrow L$ object integrity level

- 2가지 Property

Simple Integrity Property (No Write Up)	<ul style="list-style-type: none"> ◦ Corresponds to ss-property in Bell-LaPadula ◦ If subject s can modify object o, then $f_O(o) \leq f_S(s)$. ◦ no write-up Subject는 자신보다 동급이거나 낮은레벨의 Object에만 변경 가능하다
Integrity * - Property (No Read Down)	<ul style="list-style-type: none"> ◦ Corresponds to *-property in Bell-LaPadula ◦ A subject s can read an object o only if $f_S(s) \leq f_O(o)$ ◦ No read down Subject는 자신보다 동급이거나 높은레벨의 Object만 읽을 수 있다.



<2012-2 중간 기출><2012-2 기말고사 그대로 출제>

Exercise 3 (Access Control (14 points))

In a hospital we have 4 kind of users: Doctor, Nurse, Secretary, Patient. In this hierarchy, Doctors are superior to Nurse, Secretary and Patient, Nurse and Secretary are superior to Patient, but Nurse is neither superior nor inferior to Secretary. Medical information have several security levels for files, in decreasing order: Operating room, Emergency and Personal.

1. (3 points) Draw a lattice of all the security clearances.

Suppose that **Dave** the surgeon has clearance (Doctor, Operating room), **Nancy** the nurse has clearance (Nurse, Emergency), **Shari** the secretary has clearance (Secretary, Emergency) and **Paul** the patient has clearance (Patient, Personal). A **Receipt** containing payment information has clearance (Secretary, Personal), a **Prescription** for antibiotics has clearance (Doctor, Emergency), the **List** of medical tools necessary for an operation has clearance (Nurse, Operating room) and the **File** containing the home address of patients in the hospital has clearance (Secretary, Emergency).

2. (3 points) Place all these actors and documents (in bold in the paragraph above) on the preceeding lattice.

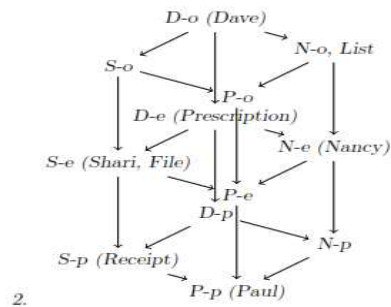
In the Bell-Lapadula model, say if the following actions are allowed, explaining each time why that is:

3. (1 point) Dave writes on the List.
4. (1 point) Nancy reads the File.
5. (1 point) Paul writes on the Prescription
6. (1 point) Shari reads the receipt.

In the BIBA model, say if the following actions are allowed, explaining each time why that is:

7. (1 point) Dave writes on the List
8. (1 point) Nancy reads the File
9. (1 point) Dave writes on the File
10. (1 point) Shari reads the prescription.

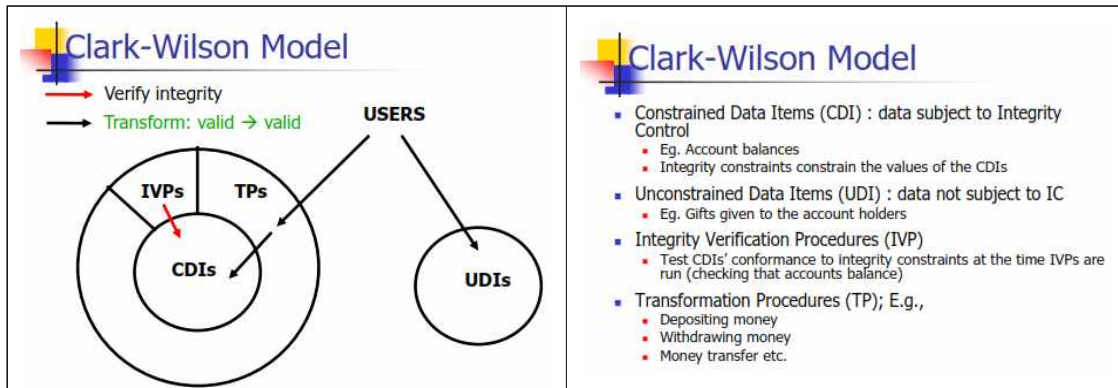
Solution :



3. no. (write down disallowed)
4. no. (unrelated nodes, nothing allowed)
5. yes. (write up, allowed)
6. yes. (read down, allowed)
7. yes. (write down allowed)
8. no. (unrelated nodes, nothing allowed)
9. yes. (write down allowed)
10. yes. (read up allowed)

5. Clake-Willson Model

- 최초의 상업적 모델임(그 동안에는 군사적인 것만 강조)
- Well Formed Transaction & Separation of duties 을 통해서 Integrity 달성됨.
- User는 직접 Data에 접근하는 것이 아니라 Program을 통해서 접근 가능



6. Harrison-Ruzo-Ullman Model(HRU Model)

- Access Right가 변경되는 것을 초점으로 모델링
- BLP는 access right를 바꾸거나 subject 또는 Object를 만들거나 없애는 정책이 없음
- 접근권한이 변화하면 Safety를 확인함

Six primitive operations for manipulating subjects, objects, and the access matrix :

- enter r into $M_{s,o}$
- delete r from $M_{s,o}$
- create subject s
- delete subject s
- create object o
- delete object o

7. Chinese Wall Model

- Consulting 할때 각 회사간 민감한 정보의 노출을 방지하기 위함
- 동적으로 Access Control이 변화함, Free Choice와 강제적 접근제어의 조합임. 처음에는 어느것이던지 접근가능하다. 하지만 접근 후에는 그 object 외 다른 정보를 접근 못하도록 한다. Chinese Wall 이 생김

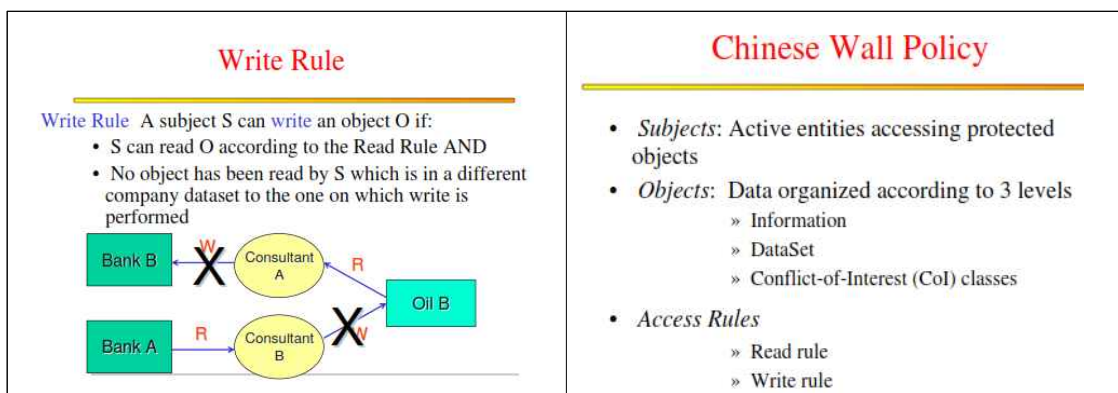
• 2가지 Property

- SS-Property : object가 요구할때만 Access가 가능하다.

Subject가 Conflict of Interest 에 정보노출을 방지함

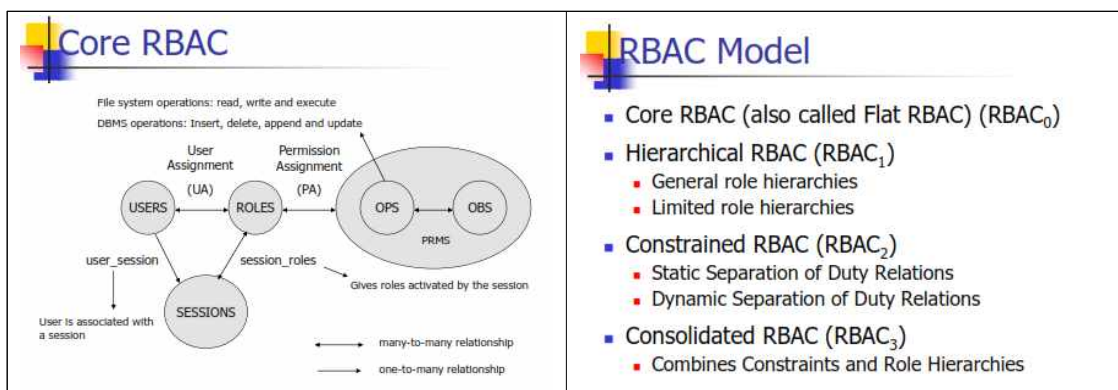
- *-Property : 다른 회사의 dataset과 접근이 가능한 object에 read할 권한이 없을 때에만 subject는 그 회사의 object의 접근이 가능하다.

회사 Data set에서 검열받지 않은 정보가 새 나가는 것을 방지



7. Role-based Access Control(RBAC)

- Subject-Object 외에 Role이라는 개념을 도입
- RBAC의 개념



- Hierarchical RBAC : Role에 계층을 만들어 운용
 - General role hierarchy : 상급자는 하급자의 모든 역할을 담당(Multiple inheritance)
 - Limited role hierarchy : 한 두명에 제한된 인원에게만 역할을 넘김
- Constrained RBAC
 - Static Separation of Duty : 상호배제 효과(감사자, 은행원 관계)..

Role이 충돌되는 경우에는 둘이 양립할 수 없다.

- Dynamic Separation of Duty : 충돌되는 역할(Role)을 동시에 수행할 수 없다.
감사하는 기간에는 은행원이 하는 역할을 할 수 없다.
- Consolidated RBAC : Hierarchical + Constrained RBAC

8. Information Flow Security Model

- Access Control 보다는 Information Flow로 중점을 맞춘다
- Covert Channel에 의한 Information Leaking을 막을 수 있다.
- 정보의 양을 측정하는 방법
 - Entrophy-based : 정보의 양을 엔트로피의 변화로 확인한다.
 - Lattice-based : requirements를 Formulizing 사용한다.
- ※ Security policy가 Formulized 되면 좋은 이유 : Automatic Checking 가능
 - Lattice 모델을 사용함
 - 모든 정보를 통제 가능하지만, 실제 구현 불가
 - 어떤 시스템의 Information Flow이 Secure한 것은 Undecidable Problem임
 - 두 가지로 나뉨 : Static(시스템 그 자체의 정보흐름 평가)


Dynamic(시스템이 실행되고 있는 상태에서 평가)

→ Execution Monitor(EM)을 통하여 확인한다.

위반사항이 발견되면 Target' s execution을 종료한다.

그러나 예측은 불가능하다.

9. Access Control Structure

Access Control Matrix	 <p>Capabilities: $S_1: \{(R_1, rW), (R_2, rWX)\}$ $S_2: \dots$</p> <p>Access Control lists: $R_1: \{(S_1, rW), (S_3, rWX)\}$ $R_2: \dots$</p>
Rule based Access Control	<ul style="list-style-type: none"> • MAC처럼 구분되어 있음 • 어떤 룰을 미리 규정해 놓고 Subject와 Object 간에 일어나는 일을 확인, 통제한다 • Firewall에서 미리 Rule을 규정하는 것
Constrained User Interface	<ul style="list-style-type: none"> • User의 입력을 제한한다
Context Dependent Access Control	<ul style="list-style-type: none"> • Object에 대한 접근은 Object의 Contents에 달려 있다. E-mail이나 DB에서 특정 단어 검색 같은 것

10. Reference Monitor : Security Policy Concept

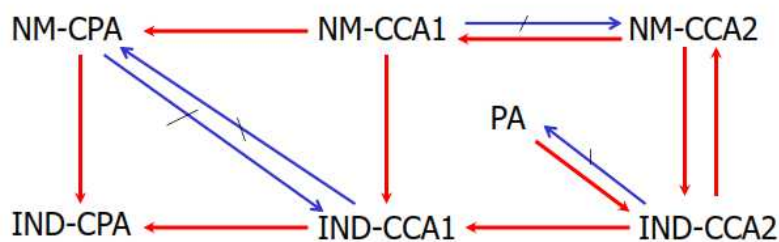
- EM(Execution Monitor)에서 구현함
 - Line 단위 체크를 한다.
 - Debug, audit, tracing 등...
 - 전제조건
 - 프로그램은 한 줄에 실행명령에만 모니터링
 - 보기 쉬워야 한다
 - 정책에 대한 집행이 일어났는지 확인해야함
 - Safety Property와 연관되어 있음
 - 1) Safety Property : 처음부터 종료할 때까지 나쁜 일은 없다.
 - 2) Liveness Property : 최종적으로 좋은 일이 발생할 것이다.
 - Schneider' s Formulization : EM으로 점검할 수 있는 일과 아닌 일을 구분하였음
 - Access Control Model을 이용가능하다.
 - Information Flow를 구현하기 힘들다
 - Availability / Liveness Property를 구현하는 것은 힘들다

V. Security Design (Cryptography)

1. Security의 종류

- Provable Security : 정해진 security goal 과 attack model 등 전제조건하에서만 안전함을 수학적으로 증명하는 것. 시스템의 한계를 명확히 알 수 있다. 여기서 Attack model 은 어떤 시스템을 분석하는데 분석환경(전제조건)을 define하는 것이다.

※ 전제조건이 더 타이트 하거나, Security Goal이 더 강력할수록 더 강력하다.



A \rightarrow B: proven that meeting notion A implies meeting B

A $\not\rightarrow$ B: proven that meeting notion A implies **not** meeting B

NOTE: A implies B iff there is a path from A to B

- Heuristic Security : 수학적으로 증명하지는 않았지만, 현재까지는 안전하다. 그러므로, 공격 시나리오에 대한 DB확보가 관건이다.

- Ad-hoc Security : 이제까지 한번도 깨진 적이 없다. 성능측정이 불가능하다. 안전성 분석 자체를 해본 적이 없다.

2. Provable Security에서 필요로 하는 것 : Security Goal 과 Attack Model

3. Kerchoff 의 원리

- 암호화, 복호화 알고리즘은 공개되어야 하며, 안전성은 키의 안전성에만 의존한다.

※ 공개를 안하면 보안시스템을 살수록 비밀을 지켜야 하며, 관리해야하는 비밀의 개수만 늘어난다.

4. 4가지 공격 유형 (Attack Model) : 공격자가 어떤 공격을 시도할 것인지 기술

구분	내용
Ciphertext Only Attack	<ul style="list-style-type: none"> · C를 다수 가지고 특정 Cn을 해독한다. · Plaintext가 노출이 되면 안됨, 한번 암호화 하면 죽을 때 가지 공개가 안된다. 문서의 보존연한 없다. 관리해야하는 C 가 늘어난다.
Known Plaintext Attack	<ul style="list-style-type: none"> · (m,C)의 쌍을 다수 가지고 Cn을 해독한다. · 평문이 공개되어도 안전해야하마. 문서 보존기간이 풀린 개념
Chosen Plaintext Attack	<ul style="list-style-type: none"> · m을 선택해서 Encryption Oracle을 이용해서 C를 만들어내고 Cn을 해독하는 것 · 2차대전에 미드웨이 해전에서 MA를 몰라 일본에 거짓 정보를 흘린다. 미드웨이 물 부족...이라고. 그러니까, 일본은 MA 물 부족이라고 송전, 그래서 MA가 미드웨이라고 확신... 미드웨이를 고르는 식으로 한다.
Chosen Ciphertext Attack	<ul style="list-style-type: none"> · C를 선택해서 Decryption Oracle을 이용해서 m을 만들어내고 Cn을 해독

5. Security Goal : 달성하려는 보안목표를 명확히 해야

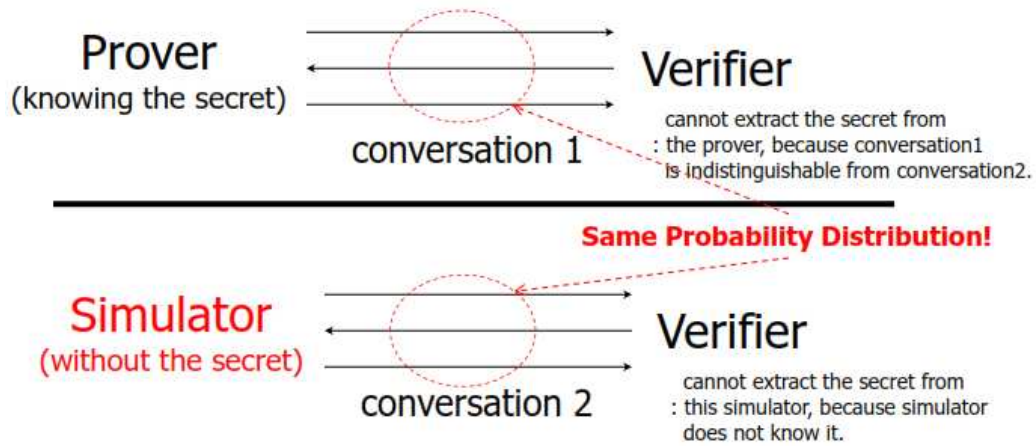
- **One-Way (OW)** : Hard to invert the encryption function
- **Semantically Secure (IND)** : Hard to obtain any partial information of a plaintext from the ciphertext
- **Non-Malleability (NM)** : For any non-trivial relation R , $E(M) \rightarrow E(R(M))$ is hard

구분	내용
One-way	<ul style="list-style-type: none"> · 공격자가 암호문으로부터 평문(전체)를 해독하는 것이 불가능 · 부분 해독이 가능하다. 대통령 5시... 이런 것 조합 가능 하지만 OW 관점에서는 상관 없다.
Semantic Secure (Indistinguishability)	<ul style="list-style-type: none"> · 어떠한 암호문이 1비트의 정보도 노출이 안된 것임 (No information leakage) · 같은 평문에는 항상 다른 암호문이 나와야 한다. · 해결책 : random padding을 붙여야 한다. 그럼 모두 값이 바뀔 <p>※ Goldwasser Micali 가 증명 한 내용</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> <p>■ Random Padding → Polynomial Security (IND-CPA)</p> <p>■ Ciphertext leaks ← Semantic Security</p> <p>NO information!</p> </div> <div style="text-align: center;"> </div> <div style="margin-left: 10px;"> <p>Random한 Padding을 붙이면, Message Space가 작아도 안전하다</p> <p>Message Space가 작아도 안전하면 어떠한 부분 정보도 노출되지 않는 Schema을 만들 수 있다.</p> </div> </div> <p><주> Polynomial Security : Message Space 가 작아도 안전하다. Semantic Security : 어떠한 부분 정보도 노출되지 않는다.</p> <p>→ Polynomial Security = Semantic Security</p>
Non-Malleability	<ul style="list-style-type: none"> · 반드시 암호문으로 하는 것이 해독밖에 없는가? · 실제 해독은 안하지만 암호문의 변경을 통해 목적 달성 · 해결책 : 고정 padding을 붙인다 → IND와 상충된다. 그래서 RSA-OAEP 같은 것을 사용한다. Random padding 과 Fixed padding을 동시 붙인다.

6. ZKIP(Zero Knowledge Interactive Proof)

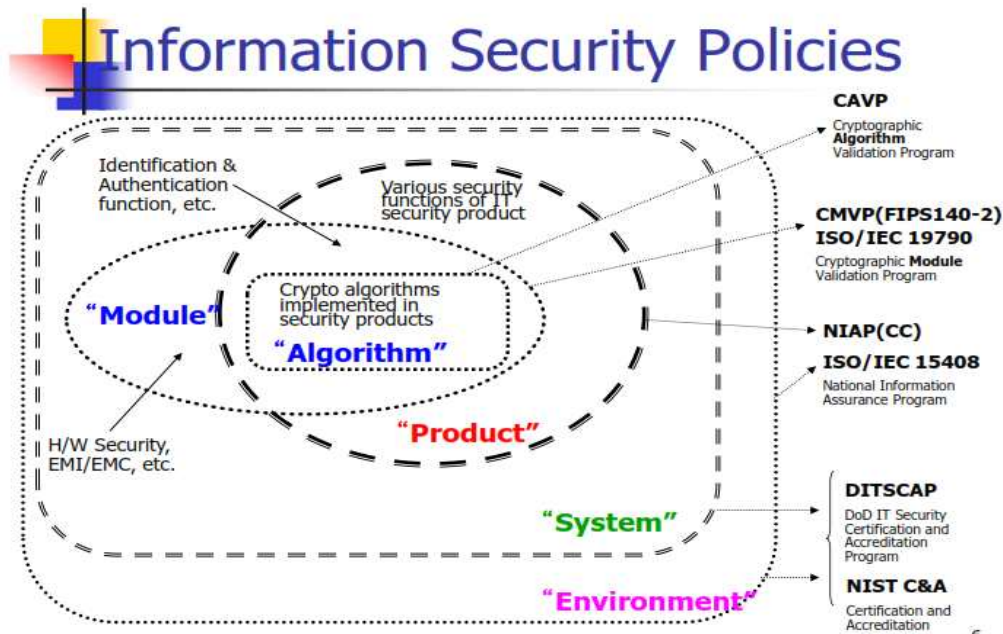
- Completeness : Prover 가 사실을 알고 있다면, Verifier는 이를 수용한다.
 - Soundness : Verifier가 인정하면, Prover가 가지고 있는 statements는 사실이다.
verifier는 Prover 여러번 반복시켜 한번이라도 거짓이 있는지 확인한다. 거짓을 알고 있는 Prover가 계속적으로 참일 확률은 무시할만하다.
 - Zero Knowledge : Proof 동안에는 어떠한 정보도 노출이 되지 않는다.
- ※ Indistinguishable 하다는 의미 : 비밀을 모르는 Simulator가 Verifier와 Communication중에 가지는 Fake Transcript 와 진짜 비밀을 아는 Prover가 Verifier와 Communication중에 가지는 Real Transcript의 값의 확률분포가 같아서 누가 누군지 구분을 못한다. 그러므로, Indistinguishable하다.

- Zero knowledge proofs are simulatable
(conversation distributions are **indistinguishable**)



VI. Security Evaluation

1. Information Security Policies



***** Information Security Point *****

- CAVP : 암호화 관련, 암호 알고리즘 중심(미국 표준 → ISO)
- CMVP : 암호 모듈 중심으로 테스트 및 제대로 구현되었는지 Validate 한다.
CAVP는 CMVP를 위해 사전에 실시해야 한다. Vender 가 제품을 개발하면 국가에서 지정한 Testing LAP에서 기본적인 TEST를 실시하고, NIST나 CSEC에서 Validate 한다(Certificate 제공).
그리고 User는 이를 구매한다.
- CC : 공통평가 기준, 정보보호 기능이 들어가 있는 모든 IT 제품을 평가한다.
- CC 제한사항 : 실험실 환경에서 테스트 하는 것으로 현장과 환경이 맞지 않으면 보안사고가 발생하고, 단일제품별로 평가하는 개념으로 시스템 전체에 대한 평가하는 것이 아니다.

***** Information Assurance Point *****

- DITSCAP : System 중심으로 보안성 평가, 순정부품으로 만든 자동차는 안전할까?
- NIST C&A : Environments 중심의 보안성 평가, 제품자체는 안전성이 있지만, 전쟁같은 복잡한 환경에서도 안전할까?

1-1. ISMS(Information Security Management System) : 정보보호 관리체계 인증제도

- 정보보호 관리체계 인증제도 : ISMS ← BS7799(영국) → ISO27001

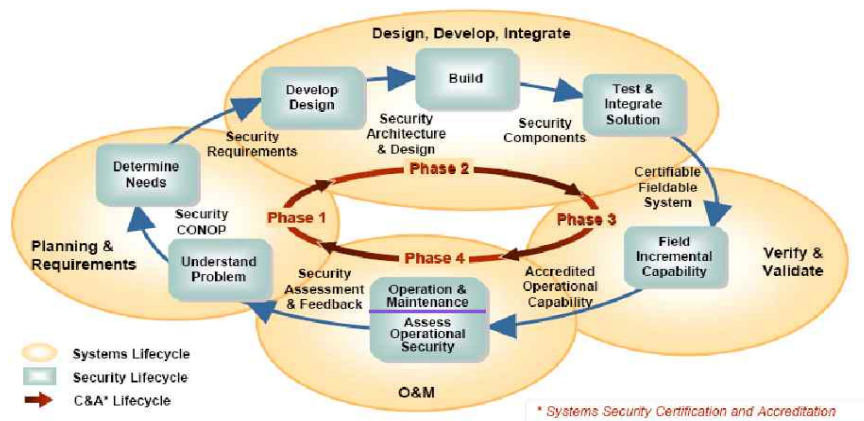
※ 조직이 얼마나 정보를 잘 관리하고 있느냐?

1-2. C&A?? = Assurance의 의미임.

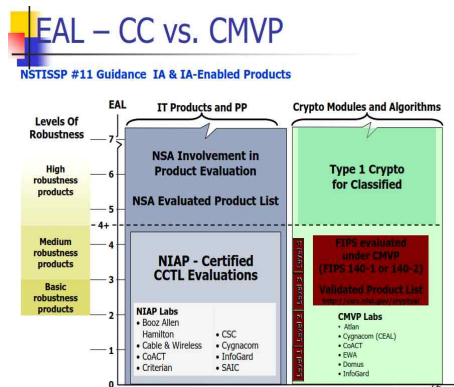
- Certification(인증) : 주어진 환경에서 시스템이 안전하게 동작하는 것 인정
- Accreditation(인정) : 여러 Lifecycle 동안 잘 유지되고 있는 것을 보장

※ 설치 전에는 Certification에 집중, 설치 후에는 Accreditation을 받아야 된다

Integrating security engineering into the systems engineering lifecycle enables successful information assurance implementation



※ C&A?? = Information Assurance의 의미임.



==> 미국은 Robustness Level로 분류 한국은 공공, 민간으로 분류... 허접해...

평가 협정대상은 EAL 1 ~ 4단계인데 그 이유는 5단계 이상은 미국에서 비밀등급에 해당한다. 즉, NSA가 개발한 비공개 알고리즘(Type D)을 써야한다.

2. IT Security Evaluation은 어떻게 할까?

- 2가지 기준이 있음

- SW Engineering에서 정의한 Process 를 잘따랐는지?

- SW 개발시의 Documentation을 Verification 함
 - : Checking design errors,
Evaluate specifications and traceability of security function
- 그러면 어느 정도까지 확인을 해야하는가? Traceability 가 만족될 때까지 해야한다. 의문점이 없어야 함. CC에는 그 해당항목이 만족될 정도 까지만 하면된다. 외국용 CC는 문서 전체 전수조사를 하지만, 국내용 CC는 문서 내용 일부만 발췌해서 한다. (국내용은 국외에서 인정 안 함)
- Process는 어떤 것으로 검증하는가?
 - CMM(능력성숙도 모델) : SW 개발프로세스를 확인한다.
 - SSE-CMM(System Security Engineering Capability Maturity Model)
 - : CMM의 Security 버전을 사용함

- Functional Testing + Vulnerability Analysis를 한다.

- All known vulnerabilities을 알고 있는지? 현재까지 공격에 안전한가?
- DB화 된 Vulnerabilities를 가지고 테스트를 한다. 만약 테스트 도중 중간에 새로운 Attack이 나오면 그것도 해보아야 함.

3. Recent History

- Early 1980년대 : Military는 일찍부터 Evaluation Process를 정의하였다.

- 정의 방법은 Functional requirements, Assurance requirements, Level of Trust 의 조합임. ==> 왜 Security Level 이 아니고, Level of Trust 일까? 이것은 안전도가 아닌 신뢰도를 말한다. 보안등급하고 차이가 있다. Assurance Requirement와 더 관련 있다. 보안기능이 많고 적음이 아니라, 그 제품이 보안 기능을 얼마나 제대로 구현했는가 문제이다.

- TC-SEC(Orange book)

- DoD에서 만든 첫 번째 기준서 (1985)
- Evaluation Level을 C1, C2, B1, B2, B3, A1으로 분류
- 다른 기준하고 다른 점

- 각각 클래스 마다 보안기능 요구조건 및 구현기능 제시
- 정형화된 기술의 제시(BLP)
- 레퍼런스 모니터(EM)의 개념 제시

→ 복잡한 보안기능보다는 간단하지만 확실한 보안기능만 있으면 됨.

안되는 것과 되는 것이 무엇인지 명확히 구분 가능하다.

- Rainbow Series : TCSEC 이후 여러 가지 기준서가 나왔음. 각기 색을 가지고 있음
- ※ 정보보호의 모든 평가 제도는 평가기관에 제출해서 평가받는 개념이 아니고 개발자 스스로 안전하다는 것을 입증하는 것!!

4. CC(Common Criteria)

- 1999년 ISO/IEC 15408 국제 표준으로 제정
- International Standard Meta-language for describing IT Security requirements
- CCRA : 상호 협정서 같은 것
 - 평가 기술력을 인정해 주겠다.
 - 내가 다른 나라로부터 평가 기술력을 검증받았다는 것
 - 임의로 감독 3나라를 선택해서 그 나라의 CC를 제대로 평가하는지 확인한다.
- 국제 / 국내용 CC 제도 차이점
 - 국제용 : CCRA 수준으로 CCRA 회원국들 사이에서 인정 가능함
 - 국내용 : 평가보증등급에서 요구하는 평가항목 중 보안기능과 취약성 검사를 제외한 일부항목을 표본추출하여 평가. 국내에서만 인정됨
- CAP(인증서 발행국) 과 CCP (인증서 수용국이 있음)
 - 인증서 발행국이 “갑” 이다.
- CC 평가 인증단계
 - 평가 신청준비 : 제출물에 대한 약식검토
 - 평가 계약단계 : 제출물 설명회(안정성 입증자료 준비/소개)
 - 평가단계 : 평가진행중 동일항목에서 2번이상 문제발생시 계약취소.. 보완용청서 수정
 - 인증단계 : 국정원 인증위원회 개최 후 승인시 인증서 발급
- 평가 기준

단계	내용
EAL 7	· Formally Verified Design and Tested · 개발문서에 대한 정형화 기술
EAL 6	· Semi-formally Verified design and Tested · 전체소스 코드
EAL 5	· Semi-formally Designed and Tested · 개발문서에 대한 전체 기술, 보안기능 전체 소스코드 · 5단계 부터는 국가 암호기술이 들어가 공개 못함(국가 기밀)
EAL 4	· Methodically Designed and Tested, Reviewed · 상세설계서, 보안정책서, 일부소스코드, 상세시험서 · 4단계까지만 상호 인정 without 암호 기술
EAL 3	· Methodically Tested and Checked · 생명주기지원, 개발보안, 오용분석서
EAL 2	· Structurally Tested · 기본설계서, 기능시험서, 취약성분석서
EAL 1	· Functionally Tested · 기능명세서, 설명서

※ 상위 레벨을 받기 위해서는 하위 레벨에서 요구하는 모든 문서가 있어야 함.

· 용어 정리

- TOE(Target of Evaluation) : Evaluation에 필요한 IT 제품이나,
시스템 그와 연관된 Documentation 들 같은 것이다.
- TOE Resource : TOE에서 사용하는 것들
- TOE Security Policy (TSP) : TOE에서 어떻게 asset을 관리, 보호,
분배할 것인가 적어놓은 Rule of Set
- TOE Security Function(TSF) : TSP가 적용되는 HW,SW,Firmware 등
모든 것들의 Set
- PP(보호프로파일) : Customer가 CC에 의해 Define된 Security Requirements
 - * Implementation과는 다르다
 - * PP represents “I WANT”
 - * 특정 제품에 독립적이다.
- ST(보안목표 명세서) : 실제 product가 무엇인지 적어놓은 Documents
 - * Specific to an implementation
 - * ST represents “I provide”
 - * 제안서에는 특정제품 언급
- RFP(Request For Proposal) : 작성시 요구조건
 - 기능 : 기능이 있는지, 기능이 있다면 제대로 구현되어 있는지?
 - 성능 : 어떤 성능이 있는지.. 그런데 CC 평가 대상이 아님
 - 환경 : 주어진 전제조건은 이렇다는 것

※ 보안제품 사고가 많이 발생하는데 원인?
환경 조건이 달라서... 문서를 잘 읽어봐라... 내용을 반드시 이해하도록...
- CC Evaluation Process

· 제안서 심사 : (보안)기능이 충분한지 확인 · ST 평가 : 기능이 충분히 발휘되었는지 확인	Sufficiency 강조	
· 검수 : 제대로 구현되어 있는지 확인 · TOE 평가 : 제품이 제안서에 서있는 그대로 되었는지 확인하는 것	Correctness 강조	

PP (Protection Profile) (1/3)		ST (Security Target)	
1	PP Introduction (보호프로파일 소개) 1.1 PP Identification (보호프로파일 식별) 1.2 PP Overview (보호프로파일 개요)	1	ST Introduction (보안목표명세서 소개) 1.1 ST Identification (보안목표명세서 식별) 1.2 ST Overview (보안목표명세서 개요) 1.3 GC Conformance (공통평가기준 적합성)
2	TOE Description (TOE 설명)	2	TOE Description (TOE 설명)
3	TOE Security Environment (TOE 보안환경) 3.1 Assumptions (가정사항) 3.2 Threats (위협) 3.3 Organisational Security Policy (조직의 보안정책)	3	TOE Security Environment (TOE 보안환경) 3.1 Assumptions (가정사항) 3.2 Threats (위협) 3.3 Organisational Security Policy (조직의 보안정책)
4	Security Objectives (보안목적) 4.1 Security Objectives for the TOE (TOE 보안목적) 4.2 Security Objectives for the Environment (환경에 대한 보안목적)	4	Security Objectives (보안목적) 4.1 Security Objectives for the TOE (TOE 보안목적) 4.2 Security Objectives for the Environment (환경에 대한 보안목적)
5	IT Security Requirements (IT 보안요구사항) 5.1 TOE Security Functional Requirements (TOE 보안기능요구사항) 5.2 TOE Security Assurance Requirements (TOE 보증요구사항) 5.3 Security Requirements for the IT Environments (IT 환경에 대한 보안요구사항)	5	IT Security Requirements (IT 보안요구사항) 5.1 TOE Security Functional Requirements (TOE 보안기능요구사항) 5.2 TOE Security Assurance Requirements (TOE 보증요구사항) 5.3 Security Requirements for the IT Environments (IT 환경에 대한 보안요구사항)
6	PP Application Notes (보호프로파일 응용 시 주의사항)	6	TOE Summary Specification (TOE 요약명세서) 6.1 TOE Security Functions (TOE 보안기능) 6.2 Assurance Measures (보증수단)
7	Rationale (이론적 근거) 7.1 Security Objectives Rationale (보안목적의 이론적 근거) 7.2 Security Requirements Rationale (보안요구사항의 이론적 근거)	7	PP Claims (보호프로파일 수용) 7.1 PP Reference (보호프로파일 참조) 7.2 PP Tailoring (보호프로파일 재정의) 7.3 PP Additions (보호프로파일 추가사항)
		8	Rationale (이론적 근거) 8.1 Security Objectives Rationale (보안목적의 이론적 근거) 8.2 Security Requirements Rationale (보안요구사항의 이론적 근거) 8.3 TOE Summary Specification Rationale (TOE 요약명세서의 이론적 근거) 8.4 PP Claims Rationale (보호프로파일 수용의 이론적 근거)

※ 왜 같은 작업(1~5)이 여러 번 반복되는가? 혹시 잘못된 것이 있는지 PP에서 판단된 위험이 빠졌는지 확인하기 위해서 임. 그래서 PP, ST 모두 확인해야 한다.

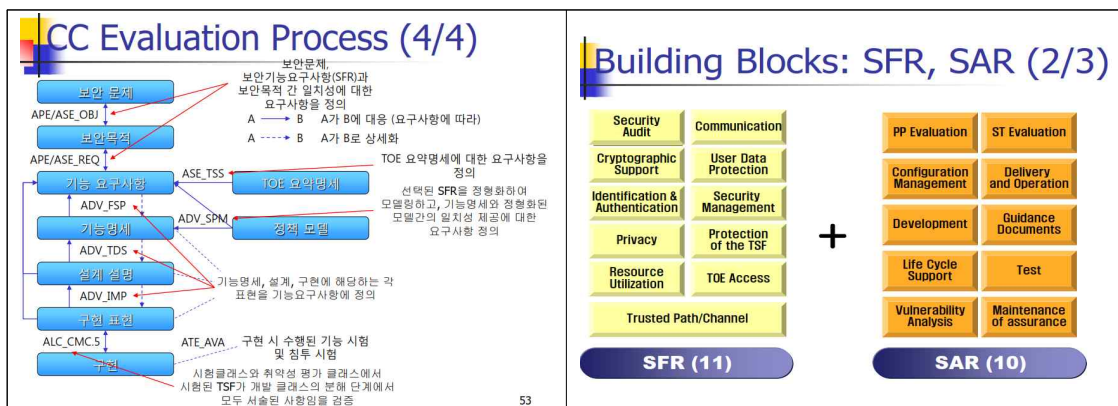
• Evaluation Process

- SFR:Security Functional Requirements : 보안기능 관련 내용

/ TOE안에 어떤 보안기능들이 있는가 확인

- SAR:Security Assurance Requirements : SFR에서 TOE가 정확히 통제되었고,

결정된 보안기능을 어떻게 개발해야 하는지 기술



※ 그 밖에 국내용 제도를 배우기는 하는데. 외국인 영강인 관계로 시험문제 절대 나올리 없음....^^!! 한국을 욕하게 되는 것임...

※ <http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/exam/>

실제 구글에서 computer security 란 과목으로 많은 대학에서 가르치고 있는데,

여기 시험 문제들이 그대로 출제 되고 있음.... filetype:pdf 로 하고 시험문제를 찾으려면 많은 도움을 얻을 수 있음!!!

Computer Security

Draft Exam with Answers. 2009.

Please note that the questions written here are a draft of the final exam. There may be typos in the questions that were corrected in the final version (available on the web page).

Question 1. Access control

(a) In access control systems, what is a *capability*? [10%]

A capability is a piece of data possession of which proves authorisation to access a resource.

(b) (i) Explain an advantage of access control lists over capability lists. [5%]

Access control lists make it easy for an administrator to see who has access to a given resource. (This is harder with capabilities.)

(ii) Explain an advantage of capability lists over access control lists. [5%]

Capabilities may be transferred offline between users. This is generally not possible with access control lists.

(c) A hospital patient record system provides login accounts for nurses. It is desired to implement the following policy:

- (i) When a nurse registers a new patient, the nurse is granted access to the patient's records for a period of 90 days.
- (ii) A nurse possessing the right to access a patient record can give that right to another user (this facilitates staff shift changes). This may be done offline.

To implement this policy, the system works as follows. When a nurse registers a new patient, a capability to access the patient record for the following 90 days is generated. The nurse stores it on a USB stick, and may copy it onto other USB sticks to give to other users. When a user attempts to access patient records, she is prompted to upload the relevant capability. The capability has the following format:

patient-id, issue-date, hmac(K, (patient-id, issue-date))

where $\text{hmac}(K, \dots)$ denotes a suitable keyed hash function with key K . The key K is a secret key known only to the patient record system. Any user in possession of this capability is able to access the records of the patient with *patient-id*, provided the date is within 90 days after *issue-date*.

- (i) Suppose nurse A registers a patient and receives such a capability. A passes it to B, B passes it to C, and C passes it to D. Is D able to use the capability? [4%]

Yes, provided the capability is still within its validity period.

- (ii) In order to stop long-lived capabilities being distributed widely, the hospital decides to adopt the policy that the nurse that initially registers the patient will have access to the records for 90 days, as before, but if she passes the capability to any other user, the validity should be 10 days from the issue date. Explain a new format for capabilities which would support this new policy. [10%]

If we include the user name in the capability, then the system can be programmed to grant 90 days to the named user, and 10 days to any other user. Thus, the new format is:

patient-id, issue-date, user-id, hmac(K, (patient-id, issue-date, user-id))

4. (3 points) Naïve voting system: We consider that 1 and 0 are the two possible ballots for an elections. A server publishes his public RSA key (N, e) . Each voter encrypt his vote, 0 or 1, as $RSA_{(N,e)}(0)$ or $RSA_{(N,e)}(1)$ respectively. At the end of the election the server decrypt all received messages and counts the votes. Show how an attacker eavesdropping on the network can learn everybody's vote.

4. Naïve voting system: We consider that 1 and 0 are the two possible ballots for an elections. A server publishes his public RSA key (N, e) . Each voter encrypt his vote, 0 or 1, as $RSA_{(N,e)}(0)$ or $RSA_{(N,e)}(1)$ respectively. At the end of the election the server decrypt all received messages and counts the votes. Show how an attacker eavesdropping on the network can learn everybody's vote.

The attacker can compute $RSA_{(N,e)}(0)$ and compare it with all the votes it sees on the network to learn everybody's vote.

그리고 해결책으로 랜덤 패딩 붙여라 그런거였음... ㅋㅋㅋ