2012년 2학기 보안공학 중간고사 정리
IAS Lab 24기 정가람

Introduction

Formulation a security policy

- 소프트웨어 공학에서 요구사항 분석과 같음.
- software 엔지니어링에서는 요구사항 분석이 가장 중요하다.
- 보안 시스템에서 요구사항을 정리한 것이 security policy이다.
- security analysis와 policy를 세우는 것이 가장 중요하다.

Software Engineering

- "Systematic (체계적인)", "Quantifiable (정량화 할 수 있는)"
- 소프트웨어 생산성을 높이는 것이 중요하다.
- 팀 단위로 개발을 할 때는 소프트웨어 엔지니어링이 중요하다.

CMM: 개발조직의 프로세스 성숙도를 평가하는 모델

- 미국 같은 경우에는 CMM이 굉장히 중요하다.
- 특정 레벨 이상이어야지만 입찰을 시도할 수 있는 경우도 있다.

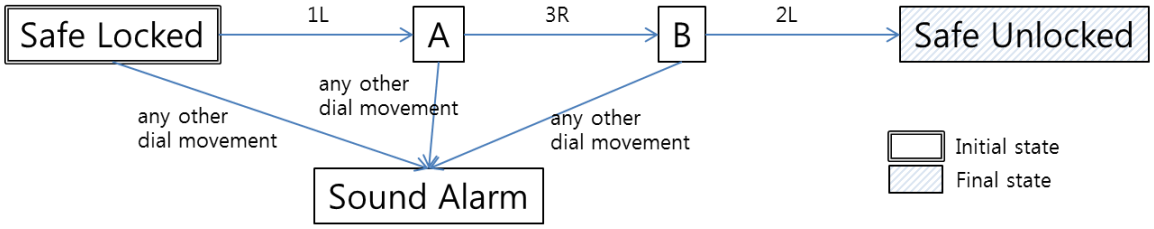
(Semi-) Formal Method

- Informal method: English (or other natural language)
- Semiformal methods: Gane & Sarsen/DeMacro/Yourdon, Entity-Relationship Diagrams

FSM Example

(M202, Open University, UK) A safe has a combination lock that can be in one of three positions, labeled 1, 2, and 3. The dial can be turned...

-> 이렇게 기술되어 있는 것을 아래와 같이 FSM으로 그릴 수 있어야 함!



S/W Engineering + Security = Security Engineering

Requirement Analysis Security Policy

-> Design -> Design ※ Design이 가장 잘 되어있는 분야는 암호학

-> Implementation -> Implementation
-> Testing – Product evaluation -> Testing – Product evaluation (=CC, CMVP)
- Process evaluation (=CMM) - Process evaluation (=SCMM: Security CMM)
※ Testing에는 프로덕트를 평가하는 과정과 프로세스를 평가하는 과정으로 나눠질 수 있다.
-> Selling

Motivation: "Security flaws are identified only at the later stages of the application lifecycle"
- 보안 정책이 잘못되어 향후 고치려고 하면, 100배 이상의 비용 차이가 날 수 있다. (잘못을 언제 발견하느냐에 따라서 다를 수 있다. 초기에 발견하면 적은 비용이 발생하지만 나중에, 특히 맨 마지막에 발견하게 되면 더 큰 비용이 든다.) 100배는 맨 마지막에 발견하였을 때.. (MS에서 그렇다고 함)
- Requirement analysis 단계에서 security flaws를 발견 -> cost = 1
- Selling 단계에서 security flaws를 발견 -> cost = 100

Security Engineering
- Object of security engineering: Find & Remove Security Weakness AS EARLY AS POSSIBLE
- From Natural disasters to Malicious Acts
(자연재해~해킹 위협까지)
- 보안공학의 목적은, 보안결함을 가급적 빨리 발견하자는 것이다. (AS EARLY AS POSSIBLE!)
- SDL(Security Development Lifecycle) @ Microsoft

	Public (공공)	Private (민간)	Military (군)
Government	NIS(정부), ㄴNCSC ㄴIT보안인증사무국(ITSCC) MOPAS(전자정부)	KCC(방송통신위원회)	Cyber command(사이버사령부)
Government Affiliated Agency	NSRI (예전에는 ETRI)	KISA ㄴKISC ETRI(사이버융합보안연구단)	NSRI(ADD에서 cyber 쪽만 떨어져 나옴) (예전에는 ADD. 군 무기체제를 모두 개발했었음)

Ross Anderson의 Security Engineering 정의(II-2p): "to make dependable system"
- Dependability = Reliability (Accidental Failures=Natural Disaster) + Security (International Failures=Malicious Acts)
- Reliability and Security are often strongly correlated in practice
- But malice is different from error!

Secure Design

Protocol security analysis

– formal model: 자동화된 방법. 시간이 적게 걸림. detail한 attack을 검출 못함

- computation model: 수작업. 시간이 많이 걸림. detail한 attack도 검출 가능

Brief History of Provable Security

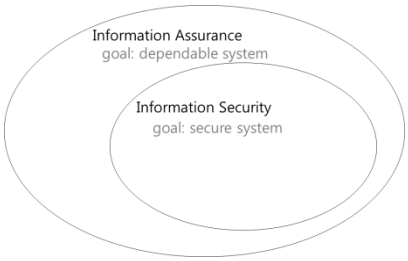
1976년을 기준으로 고전암호화 현대암호를 나눔. -> 이때를 기준으로 민간의 암호 기술이 정부의 기술력과 같아지거나, 앞질렀다고 보기 때문. 1976년 이후로 민간의 암호 연구가 활발해짐. 이때 RSA가 나옴.

※ Secure Design에 대한 이론은 암호학이 상당히 오래되었음. 가장 엄격함.

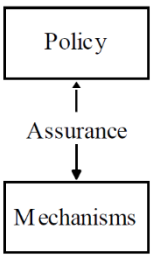
Secure Implementation/Operation

- CAVP, CMVP: 암호 알고리즘에 대한 평가 방법론
- CC: 네트워크, 시스템 보안에 대한 평가 방법론
- DITSCAP: Operation 보안에 대한 평가 방법론

Assurance



정보보증이 정보보호보다 더 상위 개념임!



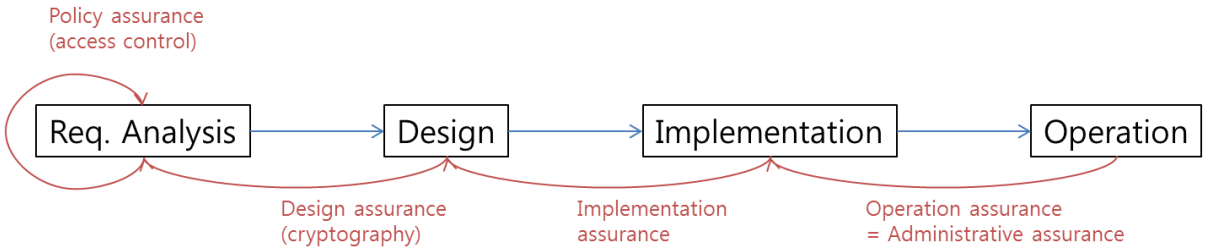
보안 요구사항을 아주 명쾌하게 서술한 것.

Mechanisms이 Policy를 준수하도록 하는 방법론(Process) = verification the process

Policy의 요구사항에 맞게 설계되고 구현된 결과물

※ Assurance 는 시스템의 신뢰성을 결정하는데 매우 중요함

Types of Assurance

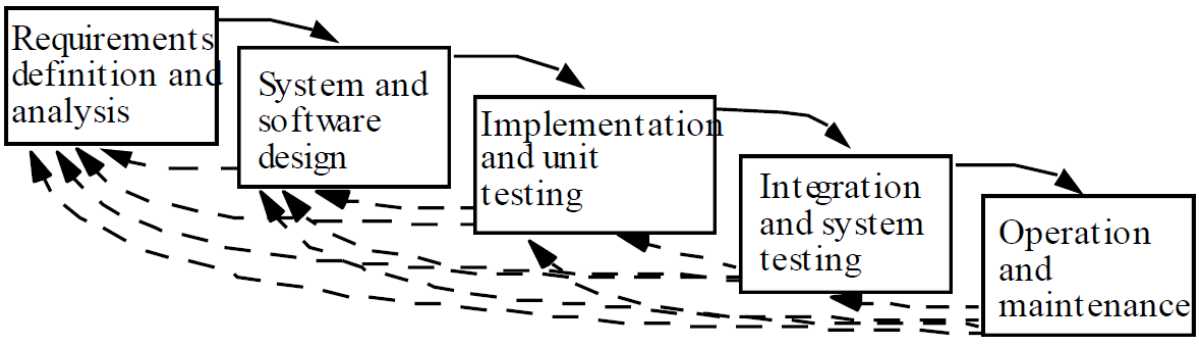


o Policy assurance: 보안 정책(Policy)이 완전하고 일관되고 명확하게 기술되었는지
o Design assurance: Design 결과물이 보안 정책(Policy)이 요구하는 바를 따르는지
o Implementation assurance: Implementation 결과물이 보안 정책(Policy)이 요구하는 바를 따르는지
o Operational assurance: 시스템을 설치, 설정, 운영할 때 보안정책(Policy)이 요구하는 바를 따르는지

Life Cycle

Conception ↓	Idea Proof of concept High-level requirements analysis – 이 주제에서 "secure"하다는 것은 무엇인가? 이 주제가 Security를 만족시킬 수 있는가? 회사는 Security를 만족하기 위해서 추가 리소스를 투입할 의지가 있는지?
Manufacture ↓	Develop detailed plans for each group involved Implement the plans to create entity
Deployment ↓	Delivery Installation and configuration
Fielded Product Life	Routine maintenance, patching Customer service, support organizations Retirement or decommission of product

Waterfall Life Cycle Model



Key point

- Assurance 는 시스템의 신뢰성을 결정하는데 매우 중요함
- Assurance level이 올라갈수록 훨씬 엄격한 수학적 알고리즘이 필요함 → 비용↑
- Assurance 는 System의 모든 life cycle에서 필요함

Foundations

Risk Score Calculation(위험도 계산)

- Risk(위험) = Expected Asset Loss(자산가치) * Vulnerabilities(취약성) * Threats(위협)
- ALE (Average Loss Expectancy) = probability of loss * total loss potential -> 뭔소리지?

※ How to measure "Expected Asset Loss?" → 보안경제학 책 보면 나와있음.

Assets (자산)

S/W, H/W,
데이터와 정보, 평판, 명성 → 측정하기 어려움
식별이 용이함, 가치평가가 어려움

Vulnerabilities (취약성)

시스템에서 사고나 혹은 비 의도적으로 자산(Asset)에 손상을 줄 수 있는 시스템의 부분.
- 희한하게 설정해놓은 계정, 알려진 결함이 있는 프로그램, 낮은 수준의 접근제어, 낮은 수준의 방화벽 설정, 충격을 입을 수 있는 것들

Threats (위협)

취약점을 이용해서 자산에 피해를 주려고 하는 적에 의한 시도

※ Identifying Threats

1. Threat lists → 위협을 나열하고
2. STRIDE(categorization) → 분류하고 (STRIDE는 위협 분류를 위해 MS에서 개발한 모델 이름임)
   - Spoofing 사기
   - Tampering 조작
   - Repudiation 거부
   - Information disclosure 정보 노출
   - Denial of Service 서비스 거부
   - Elevation of privilege 권한 상승
3. Optionally, draw threat trees → 공격 시나리오를 tree로 그리거나 문서로 정의하거나 함

Risk (위험)

위험을 계산하는 방법 2가지.

o Quantitative Risk Analysis: Risk가 정형화된 점수로 계산됨.

+ 수학적 이론에 근거하였기 때문에 신뢰도가 높음

- 결과값의 퀄리티가 입력 값의 퀄리티에 영향을 받음

- 적용 범위에 한계가 있음.

o Qualitative Risk Analysis: Risk가 계산하는 사람의 전문성에 의존. ex) DREAD model

+ 적용 범위가 넓음

- 보안 전문가의 판단력에 의존적임

※ DREAD model

Damage potential: 공격 당했을 때 입는 damage 양

Reproducibility: 공격을 수시로 가할 수 있는가? or 가끔 특정상황에서만 가할 수 있는가?

Exploitability: 공격 기술의 전문성

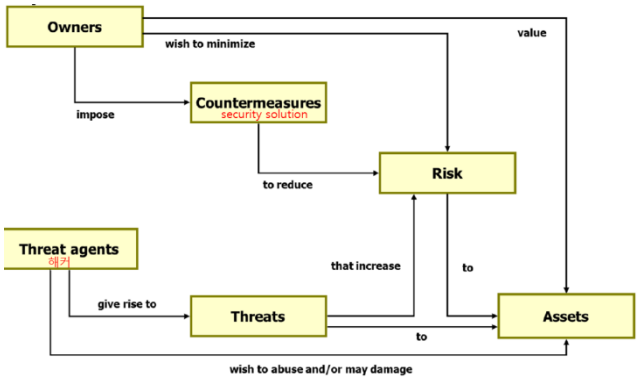
Affected users: 공격이 영향 받는 범위

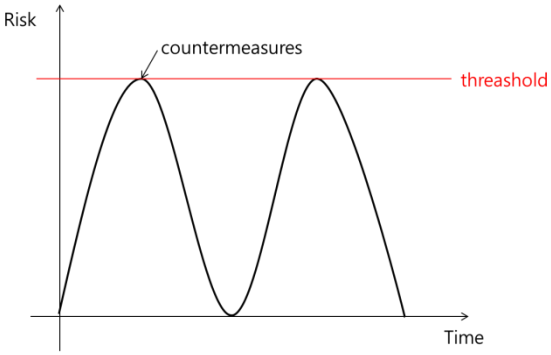
Discoverability: 이 취약점이 얼마나 발견되기 쉬운가?

Countermeasures/Risk Mitigation

- 위험 분석은 어떤 것을 조치해야 하는지를 보여줌
- 위험분석은 비용, 시간, 부정한 관리자 때문에 항상 가능하지 않음
- Baseline protection – 일반적인 경우에 보안 요구사항은 권장 요구사항을 가지고 있다

Asset, Risk Threats, Countermeasure의 관계★





용어정리

Security: 자산을 보호하는 것. 예방, 탐지, 대응(회복/자산복구) 을 수반한다.

Confidentiality: 정보가 승인 없이 공개되는 것을 방지함. privacy와 secrecy를 수반함

- Privacy: 개인적인 비밀. 예를 들면 나와 어떤 특정인 둘만의 비밀
- secrecy: 어떤 집단이나 단체의 비밀

Integrity: 정보가 승인 없이 변형되는 것을 방지함.

Availability:

- 서비스는 인가된 사용자가 필요로 할 때 언제든지 (과도한 지연 없이) 접근 가능하고 사용 가능하여야 함
- fault-tolerance (장애 허용 능력, 범위)가 필요함
- 장애(fault)는 우발적이거나 악의적임 (Byzantine?)
- DoS 공격은 악의적인 공격 중 하나

Accountability: 감사 정보는 선택적으로 유지되고 보호되어야 함.

Nonrepudiation: 부인방지.

Dependability: = Reliability (사고에 의한 장애) + Security (의도적인 장애)

Survivability:

Access Control & Security Policy

Security Policy = 우리가 무엇을 보호해야 하는지를 기술 ★

- State what should be protect
- A security policy is statement of what is, and what is not, allowed
- And how this should be achieved

Security Policy Example – MLS (Multilevel Secure)

- MLS Systems are widely used in government
- Basic Idea: 인가증을 가지고 있는 사람만 허용 (보안등급을 나누어서 관리)
- 60/70년대에 많이 사용됨
- 문제: 정책을 불투명하게 정의했음 = 정형화되지 않음

Formalizing the Security Policy

- Bell-Lapadula(1973) : 최초로 MLS를 formalizing함. 정책을 formalization 해서 기술하는 방법? → 정책들 간에 공존할 수 있는가 이런 거 증명함

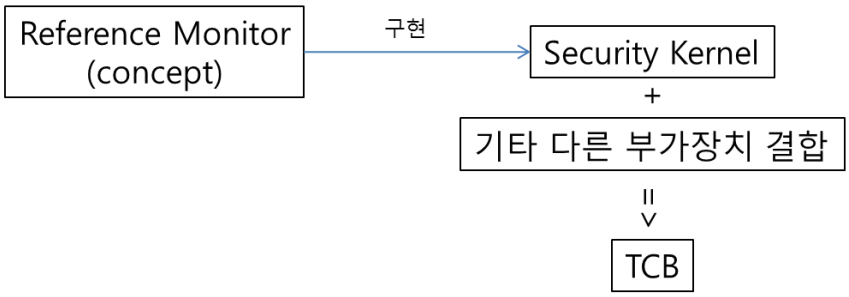
Authentication 과 Authorization ★

- Authentication: 신원파악. identification 하고는 다름. id가 주어지고, 신원을 파악하는데 사용.
  ※ Authentication 과 Identification 의 차이
  - Authentication: 사용자인증. 사용자가 ID랑 verifier를 서버로 전송해서 인증 수행. 1:1 matching
  - Identification: 개인식별. 사용자가 verifier만 서버로 전송해서 ID를 찾음. 1:N matching
- Authorization: 사용자의 권한 파악.

Reference Monitor★★ (교재 88p)

- Access Control Concept. 이걸 implementation 한 것이 방화벽이나, security kernel임.
- Security Kernel: Reference monitor를 implementation 한 것.

- TCB (Trust Computing Base): Security Kernel + Other Protection Mechanisms



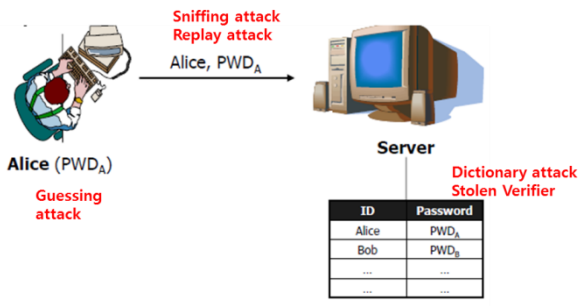
※ 교재 88p의 정의

- Reference Monitor: An access control concept that refers to an abstract machine that mediates all access to objects by subjects
- Security Kernel: The hardware, firmware, and software elements of a trusted computing based that implement the reference monitor concept. It must mediate all access, be protected from modification, and be verifiable as correct.
- Trusted Computing Base(TCB): The totality of protection mechanisms within a computer system including hardware, firmware, and software – the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of the TCB to correctly input by system administrative personnel of parameters (e.g. a user's clearance) related to the security policy.

Authentication Methods

- Multifactor authentication -> 여러 개의 인증 방법을 결합해서 사용함.
  - Something we have: OTP, 보안토큰, 인증서
  - Something we know: 패스워드, PIN 코드
  - Something we are: 지문, 생체정보

Password



o Guessing attack

- Brute force: error msg에 많은 정보를 담지 않는다.
- Dictionary: 사전에 나와 있는 단어를 한쪽부터 입력하면서 찾음. Shadow file 유출. 가장 좋은 방법은 해커가 hashed password에 접근하지 못하게 하는 것이지만, 이미 노출되었다고 한다면 'large salt values'를 사용. Multiple user에 대한 공격에는 효과적이지만 single user 대상 공격은 별로임. 'key stretching algorithms' (ex PBKDF2) 는 single user 대상 공격 시간을 지연 시

키긴 하지만 공격을 아예 막을 수는 없음.

※ Salt: 해시값에서 Plain text를 찾아내지 못하도록 추가된 공개된 난수. 내부의 모든 정보가 유출되었더라도 사전공격의 시간을 늦추기 위함. Single user를 타게팅한 공격은 막을 수 없음.

Desired Properties
Universality 보편성
Uniqueness 고유성
Permanence 지속성
Collectability 수집의 용이성

FRR: False Rejection Rate. Type I Error. 인증을 못함
FAR: False Acceptance Rate. Type II Error. 잘못 인증함
ERR: FAR과 FRR의 수치가 같아질 때의 오류 율을 의미. 시스템의 정확도를 결정하는데 중요함. 두개의 biometric 시스템을 비교하는데 도움이 됨.

Needs-to-know Principle
어떤 데이터를 봐야 할 사람만 접근 가능하게 한다. 보통 DAC를 가지고 구현함.
문제점: 권한을 부여받은 사람이 다른 사람에게 데이터를 copy해서 줄 수도 있는데 이런건 통제가 안됨. 강력한 통제가 필요한 곳에서는 쓸 수 있음.

Security Model (= Formal Specification of Security Policy): security policy를 정형화해서 기술한 것. DAC, MAC 등등

DAC: need-to-know principle. 필요한 사람만 보게 함. 데이터 소유자가 자기 임의대로 누가 이 데이터를 control 할 것인지를 결정. 유닉스 시스템에서 사용.
MAC: 강제적 접근제어. clearance level 부여. (비밀취급인가증). 모든 리소스에는 security level을 부여하고, 모든 사용자에는 clearance level을 부여해서 clearance level > security level 인 경우에만 데이터에 접근 가능하게 함.

Lattice
http://en.wikipedia.org/wiki/Lattice_(order)
Example:
- (N,|), the natural numbers also form a lattice under the operations of taking the greatest common divisor and least common multiple, with divisibility as the order relation: a ≤ b if a

divides b. 1 is bottom; 0 is top.

시큐리티 시스템을 디자인 할 때 고려해야 하는 요소

1. 데이터 혹은 operations 혹은 users. 이 중에서 어디에 포커를 맞출 것인가
2. 어떤 레이어에 보안을 만들 것인가
3. 보안을 단순하게 할 것인가 아니면 엄격한 보증을 할 것인가
4. 중앙화 할 것이냐 비중앙화할 것이냐
5. 공격자가 우리가 만들어 놓은 보안 레벨 이하를 통해 접근할 경우 이는 어떻게 막아야 하는가