시험에 나오는 것만 공부한다!

2024 시나공

기출문제집

정보처리기사

필기

11쪽 네트워크 장해

길벗알앤디 지음
(강윤석, 김용갑, 김우경, 김종일)

길벗

5과목 정보시스템 구축 관리

21.3

핵심 315 구조적 방법론

2416602

구조적 방법론은 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리(Precess) 중심의 방법론이다.

- 1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법론이다.
- 쉬운 이해 및 검증이 가능한 프로그램 코드를 생성하는 것이 목적이다.
- 복잡한 문제를 다루기 위해 분할과 정복(Divide and Conquer) 원리를 적용한다.

23.5, 22.4

핵심 316 정보공학 방법론

2416603

정보공학 방법론은 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론이다.

- 정보 시스템 개발 주기를 이용하여 대규모 정보 시스템을 구축하는데 적합하다.
  - 개체 관계도 (ERD; Entity - Relationship Diagram)를 사용한다.

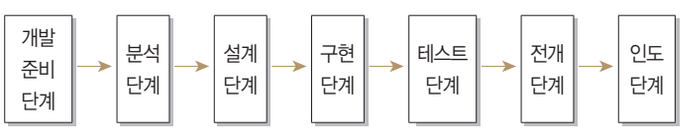
21.5, 21.3, 20.9

핵심 317 컴포넌트 기반 방법론

2416605

컴포넌트 기반(CBD; Component Based Design) 방법론은 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.

- 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.
- 새로운 기능을 추가하는 것이 간단하여 확장성이 보장된다.
- 유지 보수 비용을 최소화하고 생산성 및 품질을 향상시킬 수 있다.
- 컴포넌트 기반 방법론의 절차



23.5, 22.3

핵심 318 소프트웨어 재사용의 개요

2416701

소프트웨어 재사용(Software Reuse)은 이미 개발되어 인정받은 소프트웨어의 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 사용하는 것이다.

- 소프트웨어 개발의 품질과 생산성을 높이기 위한 방법으로, 기존에 개발된 소프트웨어와 경험, 지식 등을 새로운 소프트웨어에 적용한다.
- 재사용의 이점
  - 개발 시간과 비용을 단축시킨다.
  - 소프트웨어 품질을 향상시킨다.
  - 소프트웨어 개발의 생산성을 향상시킨다.
  - 프로젝트 실패의 위험을 감소시킨다.
  - 시스템 구축 방법에 대한 지식을 공유하게 된다.
  - 시스템 명세, 설계, 코드 등 문서를 공유하게 된다.

20.8

핵심 319 소프트웨어 재사용 방법

2416702

소프트웨어 재사용 방법에는 합성 중심 방법과 생성 중심 방법이 있다.

합성 중심 (Composition-Based)	전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법으로, 블록 구성 방법이라고도 함
생성 중심 (Generation-Based)	추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법으로, 패턴 구성 방법이라고도 함

23.7, 22.7, 22.3, 20.8

핵심 320 소프트웨어 재공학의 개요

2416703

소프트웨어 재공학(Software Reengineering)은 새로운 요구에 맞도록 기존 시스템 을 이용하여 보다 나은 시스템을 구축하고, 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것이다.

- 유지보수 비용이 소프트웨어 개발 비용의 대부분을 차지하는 문제를 염두에 두어 기존 소프트웨어의 데이터와 기능들의 개조 및 개선을 통해 유지보수성과 품질을 향상 시키려는 기술이다.

- 유지보수 생산성 향상을 통해 소프트웨어 위기를 해결하는 방법이다.
- 기존 소프트웨어의 기능을 개조하거나 개선하므로, 예방(Preventive) 유지보수 측면에서 소프트웨어 위기를 해결하는 방법이라고 할 수 있다.
- 소프트웨어 재공학도 자동화된 도구를 사용하여 소프트웨어를 분석하고 수정하는 과정을 포함한다.
- 소프트웨어의 수명이 연장되고, 소프트웨어 기술이 향상될 뿐만 아니라 소프트웨어의 개발 기간도 단축된다.
- 소프트웨어에서 발생할 수 있는 오류가 줄어들고, 비용이 절감된다.
- 주요 활동

분석 (Analysis)	기존 소프트웨어의 명세서를 확인하여 소프트웨어의 동작을 이해하고, 재공학할 대상을 선정하는 활동
재구성 (Restructuring)	• 기존 소프트웨어의 구조를 향상시키기 위하여 코드를 재구성하는 활동 • 소프트웨어의 기능과 외적인 동작은 바뀌지 않음
역공학 (Reverse Engineering)	• 기존 소프트웨어를 분석하여 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 활동 • 일반적인 개발 단계와는 반대 방향으로 기존 코드를 복구하거나, 기존 소프트웨어의 구성 요소와 그 관계를 파악하여 설계도를 추출함
이식(Migration)	기존 소프트웨어를 다른 운영체제나 하드웨어 환경에서 사용할 수 있도록 변환하는 활동

23.7, 23.5, 21.5, 21.3, 20.9, 20.8, 20.6

2416704

핵심 321 CASE의 개요

CASE(Computer Aided Software Engineering)는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

- 객체지향 시스템, 구조적 시스템 등 다양한 시스템에서 활용되는 자동화 도구(CASE Tool)이다.
- 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다.
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해 주는 통합된 도구를 제공해 주는 기술이다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구들을 지원하고, 개발자들은 이 도구를 사용하여 소프트웨어 개발의 표준화를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- CASE의 주요 기능 : 소프트웨어 생명 주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등
- CASE의 원천 기술 : 구조적 기법, 프로토타이핑, 자동 프로그래밍, 정보 저장소, 분산 처리

23.3, 22.7, 22.4, 22.3, 21.8, 21.3, 20.6

2417002

핵심 322 LOC(원시 코드 라인 수, source Line Of Code) 기법

LOC 기법은 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.

- 측정이 용이하고 이해하기 쉬워 가장 많이 사용된다.
- 예측치를 이용하여 생산성, 노력, 개발 기간 등의 비용을 산정한다.

$$\text{예측치} = \frac{a+4m+b}{6}$$ 단, a : 낙관치, b : 비관치, m : 기대치(중간치)

- 산정 공식
  - 노력(인월) = 개발 기간 × 투입 인원
    = LOC / 1인당 월평균 생산 코드 라인 수
  - 개발 비용 = 노력(인월) × 단위 비용(1인당 월평균 인건비)
  - 개발 기간 = 노력(인월) / 투입 인원
  - 생산성 = LOC / 노력(인월)

21.5, 20.9

핵심 323 수학적 산정 기법의 개요

2417101

수학적 산정 기법은 상향식 비용 산정 기법으로, 경험적 추정 모형, 실험적 추정 모형 이라고도 하며, 개발 비용 산정의 자동화를 목표로 한다.

- 비용을 자동으로 산정하기 위해 사용되는 공식은 과거 유사한 프로젝트를 기반으로하여 경험적으로 유도된 것이다.
- 수학적 산정 기법에는 COCOMO 모형, Putnam 모형, 기능 점수(FP) 모형 등이 있으며 각 모형에서는 지정된 공식을 사용하여 비용을 산정한다.

22.7, 22.4

핵심 324 COCOMO 모형 개요

2417102

COCOMO(COnstructive COst MOdel) 모형은 보헴(Boehm)이 제안한 것으로, 원시 프로그램의 규모인 LOC(원시 코드 라인 수)에 의한 비용 산정 기법이다.

- 개발할 소프트웨어의 규모(LOC)를 예측한 후 이를 소프트웨어 종류에 따라 다르게 책정되는 비용 산정 방정식에 대입하여 비용을 산정한다.
- 비교적 작은 규모의 프로젝트들을 통계 분석한 결과를 반영한 모델이므로 중소 규모 소프트웨어 프로젝트 비용 추정에 적합하다.
- 같은 규모의 프로그램이라도 그 성격에 따라 비용이 다르게 산정된다.
- 비용 산정 결과는 프로젝트를 완성하는 데 필요한 노력(Man-Month)으로 나타난다.

23.5, 22.7, 21.8, 21.5, 21.3, 20.8, 20.6

핵심 325 COCOMO의 소프트웨어 개발 유형

2417103

조직형 (Organic Mode)	• 기관 내부에서 개발된 중·소 규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형 • 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합함
반분리형 (Semi-Detached Mode)	• 조직형과 내장형의 중간형으로 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형 • 컴파일러, 인터프리터와 같은 유틸리티 개발에 적합함
내장형 (Embedded Mode)	• 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI)라인 이상의 소프트웨어를 개발하는 유형 • 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합함

20.6

핵심 326 Putnam 모형

2417104

Putnam 모형은 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.

- 푸트남(Putnam)이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.
- 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

20.8

핵심 327 기능 점수(FP) 모형

2417105

기능 점수(Function Point) 모형은 알브레히트(Albrecht)가 제안한 것으로, 소프트웨어의 기능을 증대시키는 요인별로 가중치를 부여하고, 요인별 가중치를 합산하여 총 기능 점수를 산출하며 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법이다.

- 소프트웨어 기능 증대 요인
  - 자료 입력(입력 양식)
  - 정보 출력(출력 보고서)
  - 명령어(사용자 질의수)
  - 데이터 파일
  - 필요한 외부 루틴과의 인터페이스

※ 자동화 추정 도구

- SLIM : Rayleigh-Norden 곡선과 Putnam 예측 모델을 기초로 하여 개발된 자동화 추정 도구
- ESTIMACS : 다양한 프로젝트와 개인별 요소를 수용하도록 FP 모형을 기초로 하여 개발된 자동화 추정 도구

22.4

핵심 328 PERT

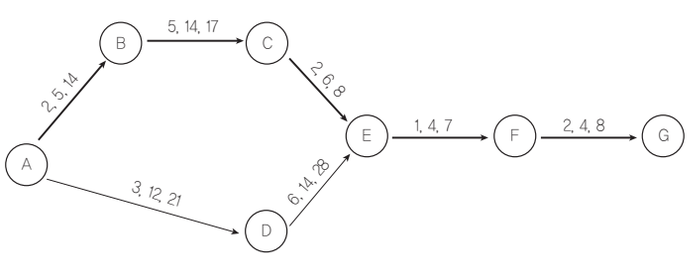
2417202

PERT(Program Evaluation and Review Technique, 프로그램 평가 및 검토 기술)는 프로젝트에 필요한 전체 작업의 상호 관계를 표시하는 네트워크로 각 작업별로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료 시기를 결정하는 방법이다.

- 과거에 경험이 없어서 소요 기간 예측이 어려운 소프트웨어에서 사용한다.
- 노드와 간선으로 구성되며 원 노드에는 작업을, 간선(화살표)에는 낙관치, 기대치, 비관치를 표시한다.
- 결정 경로, 작업에 대한 경계 시간, 작업 간의 상호 관련성 등을 알 수 있다.
- 다음과 같은 PERT 공식을 이용하여 작업 예측치를 계산한다.

$$\text{작업 예측치} = \frac{\text{비관치} + 4 \times \text{기대치} + \text{낙관치}}{6}$$

$$\text{평방 편차} = \left[\frac{(\text{비관치} - \text{낙관치})}{6}\right]^2$$



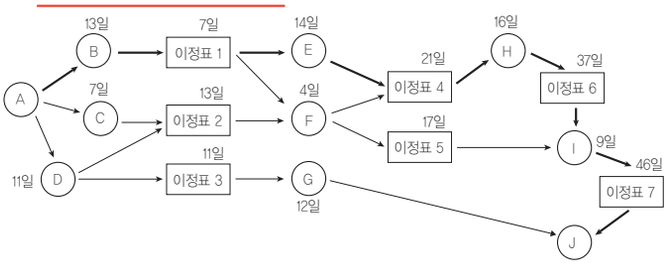
23.7, 22.7, 20.8

핵심 329 CPM

2417203

CPM(Critical Path Method, 임계 경로 기법)은 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는데 사용하는 기법이다.

- CPM은 노드와 간선으로 구성된 네트워크로 노드는 작업을, 간선은 작업 사이의 전후 의존 관계를 나타낸다.
- 원형 노드는 각 작업을 의미하며 각 작업 이름과 소요 기간을 표시하고, 박스 노드 는 이정표를 의미하며 박스 노드 위에는 예상 완료 시간을 표시한다.
- 간선을 나타내는 화살표의 흐름에 따라 각 작업이 진행되며, 전 작업이 완료된 후 다음 작업을 진행할 수 있다.
- 임계 경로는 최장 경로를 의미한다.



22.3

핵심 330 간트 차트

2417204

간트 차트는 프로젝트의 각 작업들이 언제 시작하고 언제 종료되는지에 대한 작업 일 정을 막대 도표를 이용하여 표시하는 프로젝트 일정표로, 시간선(Time-Line) 차트라고도 한다.

- 중간 목표 미달성 시 그 이유와 기간을 예측할 수 있게 한다.
- 사용자와의 문제점이나 예산의 초과 지출 등도 관리할 수 있게 한다.
- 자원 배치와 인원 계획에 유용하게 사용된다.
- 다양한 형태로 변경하여 사용할 수 있다.
- 작업 경로는 표시할 수 없으며, 계획의 변화에 대한 적응성이 약하다.
- 계획 수립 또는 수정 때 주관적 수치에 기울어지기 쉽다.
- 간트 차트는 이정표, 작업 일정, 작업 기간, 산출물로 구성되어 있다.
- 수평 막대의 길이는 각 작업(Task)의 기간을 나타낸다.

23.5, 23.2, 22.3

핵심 331 프로젝트 관리 (Project Management)

2417331

프로젝트 관리는 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템을 개발하기 위한 전반적인 활동이다.

관리 유형	주요 내용
일정 관리	작업 순서, 작업 기간 산정, 일정 개발, 일정 통제
비용 관리	비용 산정, 비용 예산 편성, 비용 통제
인력 관리	프로젝트 팀 편성, 자원 산정, 프로젝트 조직 정의, 프로젝트 팀 개발, 자원 통제, 프로젝트 팀 관리
위험 관리	위험 식별, 위험 평가, 위험 대처, 위험 통제
품질 관리	품질 계획, 품질 보증 수행, 품질 통제 수행

Risk Analysis

21.5

핵심 332 ISO/IEC 12207

2417402

ISO/IEC 12207은 ISO(International Organization for Standardization, 국제표준화기구)에서 만든 표준 소프트웨어 생명 주기 프로세스로, 소프트웨어의 개발, 운영, 유지보수 등을 체계적으로 관리하기 위한 소프트웨어 생명 주기 표준을 제공한다.

- ISO/IEC 12207은 기본 생명 주기 프로세스, 지원 생명 주기 프로세스, 조직 생명 주기 프로세스로 구분한다.

기본 생명 주기 프로세스	획득, 공급, 개발, 운영, 유지보수 프로세스
지원 생명 주기 프로세스	품질 보증, 검증, 확인, 활동 검토, 감사, 문서화, 형상 관리, 문제 해결 프로세스
조직 생명 주기 프로세스	관리, 기반 구조, 훈련, 개선 프로세스

23.2, 20.9, 20.6

핵심 333 CMMI(Capability Maturity Model Integration)

2417403

CMMI(능력 성숙도 통합 모델)는 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델로, 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발하였다.

- CMMI의 소프트웨어 프로세스 성숙도는 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분한다.

단계	프로세스	특징
초기 (Initial)	정의된 프로세스 없음	작업자 능력에 따라 성공 여부 결정
관리 (Managed)	규칙화된 프로세스	특정한 프로젝트 내의 프로세스 정의 및 수행
정의 (Defined)	표준화된 프로세스	조직의 표준 프로세스를 활용하여 업무 수행
정량적 관리 (Quantitatively Managed)	예측 가능한 프로세스	프로젝트를 정량적으로 관리 및 통제
최적화 (Optimizing)	지속적 개선 프로세스	프로세스 역량 향상을 위해 지속적인 프로세스 개선

21.5, 20.9, 20.8

핵심 334 SPICE(Software Process Improvement and Capability dEtermination)

2417404

SPICE(소프트웨어 처리 개선 및 능력 평가 기준)는 정보 시스템 분야에서 소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준으로, 공식 명칭은 ISO/IEC 15504이다.

- SPICE는 5개의 프로세스 범주와 40개의 세부 프로세스로 구성된다.
- SPICE는 프로세스 수행 능력 단계를 불완전, 수행, 관리, 확립, 예측, 최적화의 6단계로 구분한다.

Level

단계	특징
0 불완전 (Incomplete)	프로세스가 구현되지 않았거나 목적을 달성하지 못한 단계
1 수행 (Performed)	프로세스가 수행되고 목적이 달성된 단계
2 관리 (Managed)	정의된 자원의 한도 내에서 그 프로세스가 작업 산출물을 인도하는 단계
3 확립 (Established)	소프트웨어 공학 원칙에 기반하여 정의된 프로세스가 수행되는 단계
4 예측 (Predictable)	프로세스가 목적 달성을 위해 통제되고, 양적인 측정을 통해서 일관되게 수행되는 단계
5 최적화 (Optimizing)	프로세스 수행을 최적화하고, 지속적인 개선을 통해 업무 목적을 만족시키는 단계

22.3, 20.6

핵심 335 소프트웨어 개발 방법론 테일러링

2459942

소프트웨어 개발 방법론 테일러링은 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.

- 소프트웨어 개발 방법론 테일러링 작업 시 고려해야 할 사항에는 내부적 기준과 외부적 기준이 있다.

내부적 기준	• 목표 환경 : 시스템의 개발 환경과 유형이 서로 다른 경우 테일러링이 필요함 • 요구사항 : 프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우 테일러링이 필요함 • 프로젝트 규모 : 비용, 인력, 기간 등 프로젝트의 규모가 서로 다른 경우 테일러링이 필요함 • 보유 기술 : 프로세스, 개발 방법론, 산출물, 구성원의 능력 등이 서로 다른 경우 테일러링이 필요함
외부적 기준	• 법적 제약사항 : 프로젝트별로 적용될 IT Compliance가 서로 다른 경우 테일러링이 필요함 • 표준 품질 기준 : 금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우 테일러링이 필요함

23.7, 22.4, 21.8, 21.5, 20.9

핵심 336 소프트웨어 개발 프레임워크

2417601

프레임워크(Framework)는 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템이다.

- 선행 사업자의 기술에 의존하지 않은 표준화된 개발 기반으로 인해 사업자 종속성이 해소된다.
- 프레임워크의 주요 기능에는 예외 처리, 트랜잭션 처리, 메모리 공유, 데이터 소스 관리, 서비스 관리, 쿼리 서비스, 로깅 서비스, 사용자 인증 서비스 등이 있다.
- 프레임워크의 종류

스프링 프레임워크 (Spring Framework)	자바 플랫폼을 위한 오픈 소스 경량형 애플리케이션 프레임워크
전자정부 프레임워크	우리나라의 공공부문 정보화 사업 시 효율적인 정보 시스템의 구축을 지원하기 위해 필요한 기능 및 아키텍처를 제공하는 프레임워크
닷넷 프레임워크 (.NET Framework)	Windows 프로그램의 개발 및 실행 환경을 제공하는 프레임워크로, Microsoft 사에서 통합 인터넷 전략을 위해 개발함

22.4, 21.8, 20.9, 20.6

핵심 337 프레임워크의 특성

2417631

모듈화 (Modularity)	프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킴 프레임워크는 개발표준에 의한 모듈화로 인해 유지 보수가 용이함
재사용성 (Reusability)	프레임워크는 재사용 가능한 모듈들을 제공함으로써 예산 절감, 생산성 향상, 품질 보증이 가능함
확장성 (Extensibility)	프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
제어의 역흐름 (Inversion of Control)	개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킴

23.2, 22.4, 21.8

핵심 338 소프트웨어 정의 기술(SDE, SDx; Software-Defined Everything)

2417702

소프트웨어 정의 기술은 네트워크, 데이터 센터 등에서 소유한 자원을 가상화하여 개별 사용자에게 제공하고, 중앙에서는 통합적으로 제어가 가능한 기술이다.

- 관련 용어

용어	의미
소프트웨어 정의 네트워킹 (SDN; Software Defined Networking,)	• 네트워크를 컴퓨터처럼 모델링하여 여러 사용자가 각각의 소프트웨어들로 네트워킹을 가상화하여 제어하고 관리하는 네트워크 • 하드웨어에 의존하는 네트워크 체계에 비해 보다 효율적으로 네트워크를 제어, 관리할 수 있음 • 기존 네트워크에는 영향을 주지 않으면서 특정 서비스의 전송 경로 수정을 통하여 인터넷 상에서 발생하는 문제를 처리할 수 있음
소프트웨어 정의 데이터 센터 (SDDC; Software Defined Data Center)	데이터 센터의 모든 자원을 가상화하여 인력의 개입없이 소프트웨어 조작만으로 관리 및 제어되는 데이터 센터
소프트웨어 정의 스토리지 (SDS; Software-Defined Storage)	물리적인 데이터 스토리지(Data Storage)를 가상화하여 여러 스토리지를 하나처럼 관리하거나, 하나의 스토리지를 여러 스토리지로 나눠 사용할 수 있는 기술

23.7, 23.5, 23.2, 22.7, 22.4, 21.8

핵심 339 네트워크 관련 신기술

2417705

IoT(Internet of Things, 사물 인터넷)	정보 통신 기술을 기반으로 실세계(Physical World)와 가상 세계(Virtual World)의 다양한 사물들을 인터넷으로 서로 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 기술
메시 네트워크 (Mesh Network)	차세대 이동통신, 홈네트워킹, 공공 안전 등 특수 목적을 위한 새로운 방식의 네트워크 기술로, 대규모 디바이스의 네트워크 생성에 최적화되어 있음
피코넷 (PICONET)	여러 개의 독립된 통신장치가 블루투스 기술이나 UWB 통신 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술
파장 분할 다중화 (WDM, Wavelength Division Multiplexing)	• 광섬유를 이용한 통신 기술의 하나로, 파장이 서로 다른 복수의 신호를 보냄으로써 여러 대의 단말기가 동시에 통신 회선을 사용할 수 있도록 하는 것 • 파장이 다른 광선끼리는 서로 간섭을 일으키지 않는 성질을 이용한 기술
클라우드 기반 HSM (Cloud-based Hardware Security Module)	• 클라우드를 기반으로 암호화 키의 생성 · 저장 · 처리 등의 작업을 수행하는 보안기기를 가리키는 용어 • 클라우드에 인증서를 저장하므로 스마트폰과 같은 개별 기기에 인증서를 저장할 필요가 없음 • 암호화 키 생성이 하드웨어적으로 구현되기 때문에 소프트웨어적으로 구현된 암호 기술이 가지는 보안 취약점을 무시할 수 있음
파스-타(PaaS-TA)	• 소프트웨어 개발 환경을 제공하기 위해 개발한 개방형 클라우드 컴퓨팅 플랫폼 • 국내 IT 서비스 경쟁력 강화를 목표로 과학기술정보통신부와 한국정보 화진흥원이 연구개발(R&D)을 지원하였으며, 인프라 제어 및 관리 환경, 실행 환경, 개발 환경, 서비스 환경, 운영 환경으로 구성되어 있음
징(Zing)	• 10cm 이내 거리에서 3.5Gbps 속도의 데이터 전송이 가능한 초고속 근접무선통신(NFC) • 휴대용 스마트 기기, 노트북, 쇼핑몰 · 거리 등의 광고나 키오스크에 접목하여 사용할 수 있음
SSO (Single Sign On)	• 한 번의 로그인으로 개인이 가입한 모든 사이트를 이용할 수 있게 해주는 시스템 • 개인정보를 각 사이트마다 일일이 기록해야 하던 불편함을 해소할 수 있음 • 기업에서는 회원에 대한 통합관리가 가능해 마케팅을 극대화시킬 수 있음
스마트 그리드 (Smart Grid)	• 정보 기술을 전력에 접목해 효율성을 높인 시스템으로, 전력 IT라고도 부름 • 전력선을 기반으로 모든 통신, 정보, 관련 애플리케이션 인프라를 하나의 시스템으로 통합하여 관리함으로써 효율적인 에너지 관리가 가능함

23.2, 21.3, 20.8

핵심 340 네트워크(Network) 설치 구조

2459943

성형 (Star, 중앙 집중형)	• 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태 • 포인트 투 포인트(Point-to-Point) 방식으로 회선을 연결함
링형 (Ring, 루프형)	• 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트(Point-to-Point) 방식으로 연결시킨 형태 • 분산 및 집중 제어 모두 가능함 • 데이터는 단방향 또는 양방향으로 전송할 수 있으며, 단방향 링의 경우 컴퓨터, 단말장치, 통신 회선 중 어느 하나라도 고장나면 전체 통신망에 영향을 미침
버스형 (Bus)	• 한 개의 통신 회선에 여러 대의 단말장치가 연결되어 있는 형태 • 물리적 구조가 간단하고, 단말장치의 추가와 제거가 용이함 • 단말장치가 고장나더라도 통신망 전체에 영향을 주지 않기 때문에 신뢰성을 높일 수 있음
계층형 (Tree, 분산형)	중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태
망형 (Mesh)	• 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높음 • 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리함 • 보통 공중 데이터 통신망에서 사용되며, 통신 회선의 총 경로가 가장 김 • 모든 노드를 망형으로 연결하려면 노드의 수가 n개일 때, n(n-1)/2개의 회선이 필요하고 노드당 n-1개의 포트가 필요함

21.8

핵심 341 VLAN(Virtual Local Area Network)

2417831

VLAN은 LAN의 물리적인 배치와 상관없이 논리적으로 분리하는 기술로, 접속된 장비들의 성능 및 보안성을 향상시킬 수 있다.

P219 LOD

22.7, 21.5, 20.6

핵심 342 LAN의 표준안

2417808

IEEE 802의 주요 표준 규격

IEEE 802 위원회에서 지정한 LAN의 표준 규격은 다음과 같다.

표준 규격	내용
802.1	전체의 구성, OSI 참조 모델과의 관계, 통신망 관리 등에 관한 규약
802.2	논리 링크 제어(LLC) 계층에 관한 규약
802.3	CSMA/CD 방식의 매체 접근 제어 계층에 관한 규약
802.4	토큰 버스 방식의 매체 접근 제어 계층에 관한 규약
802.5	토큰 링 방식의 매체 접근 제어 계층에 관한 규약
802.6	도시형 통신망(MAN)에 관한 규약
802.9	종합 음성/데이터 네트워크에 관한 규약
802.11	무선 LAN에 관한 규약

802.11의 버전

802.11 (초기 버전)	2.4GHz 대역 전파와 CSMA/CA 기술을 사용해 최고 2Mbps까지의 전송 속도를 지원함
802.11a	5GHz 대역의 전파를 사용하며, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원함
802.11b	802.11 초기 버전의 개선안으로 등장하였으며, 초기 버전의 대역 전파와 기술을 사용해 최고 11Mbps의 전송 속도로 기존에 비해 5배 이상 빠르게 개선됨
802.11e	802.11의 부가 기능 표준으로, QoS 기능이 지원되도록 하기 위해 매체 접근 제어(MAC) 계층에 해당하는 부분을 수정함
802.11g	2.4GHz 대역의 전파를 사용하지만 5GHz 대역의 전파를 사용하는 802.11a와 동일한 최고 54Mbps까지의 전송 속도를 지원함
802.11i	802.11의 보안 기능 표준으로, 인증방식에 WPA/WPA2를 사용함
802.11n	2.4GHz 대역과 5GHz 대역을 사용하는 규격으로, 최고 600Mbps까지의 전송 속도를 지원함

21.5

핵심 343 CSMA/CA(Carrier Sense Multiple Access/Collision Avoidance)

2417832

CSMA/CA는 무선랜에서 데이터 전송 시 매체가 비어있음을 확인한 뒤 충돌을 피하기 위해 일정한 시간을 기다린 후 데이터를 전송하는 방법이다.

- 회선을 사용하지 않는 경우에도 확인 신호를 전송하여 동시 전송에 의한 충돌을 예방한다.

22.4, 21.5, 20.8, 20.6

핵심 344 경로 제어 프로토콜(Routing Protocol)

2418002

IGP(Interior Gateway Protocol, 내부 게이트웨이 프로토콜)	• 하나의 자율 시스템(AS) 내의 라우팅에 사용되는 프로토콜 • RIP(Routing Information Protocol) – 현재 가장 널리 사용되는 라우팅 프로토콜로 거리 벡터 라우팅 프로토콜이라고도 불리며, 최단 경로 탐색에 Bellman-Ford 알고리즘이 사용됨 – 소규모 동종의 네트워크(자율 시스템, AS) 내에서 효율적인 방법 – 최대 홉(Hop) 수를 15로 제한하므로 15 이상의 경우는 도달할 수 없는 네트워크를 의미하는데 이것은 대규모 네트워크에서는 RIP를 사용할 수 없음을 의미함 • OSPF(Open Shortest Path First protocol) – RIP의 단점을 해결하여 새로운 기능을 지원하는 인터넷 프로토콜로, 대규모 네트워크에서 많이 사용됨 – 인터넷 망에서 이용자가 최단 경로를 선정할 수 있도록 라우팅 정보에 노드 간의 거리 정보, 링크 상태 정보를 실시간으로 반영하여 최단 경로로 라우팅을 지원함 – 최단 경로 탐색에 다익스트라(Dijkstra) 알고리즘을 사용함 – 라우팅 정보에 변화가 생길 경우 변화된 정보만 네트워크 내의 모든 라우터에 알림 – 하나의 자율 시스템(AS)에서 동작하면서 내부 라우팅 프로토콜의 그룹에 도달함
EGP(Exterior Gateway Protocol, 외부 게이트웨이 프로토콜)	자율 시스템(AS) 간의 라우팅, 즉 게이트웨이 간의 라우팅에 사용되는 프로토콜
BGP(Border Gateway Protocol)	• 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 만들어짐 • 초기에 BGP 라우터들이 연결될 때에는 전체 경로 제어표(라우팅 테이블)를 교환하고, 이후에는 변화된 정보만을 교환함

20.9

핵심 345 흐름 제어(Flow Control)

2418004

흐름 제어란 네트워크 내의 원활한 흐름을 위해 송 · 수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능이다.

정지-대기 (Stop-and-Wait)	• 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식 • 한 번에 하나의 패킷만을 전송할 수 있음
슬라이딩 윈도우 (Sliding Window)	• 확인 신호, 즉 수신 통지를 이용하여 송신 데이터의 양을 조절하는 방식 • 수신 측의 확인 신호를 받지 않더라도 미리 정해진 패킷의 수만큼 연속적으로 전송하는 방식으로, 한 번에 여러 개의 패킷을 전송할 수 있어 전송 효율이 좋음 • 송신 측은 수신 측으로부터 확인 신호(ACK) 없이도 보낼 수 있는 패킷의 최대치를 미리 약속받는데, 이 패킷의 최대치가 윈도우 크기(Window Size)를 의미함 • 윈도우 크기(Window Size)는 상황에 따라 변한다. 즉, 수신 측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신 측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소함

23.7, 23.2, 22.3, 21.8, 20.9, 20.8

핵심 346 SW 관련 용어

2418101

매시업(Mashup)	웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술이다. 즉 다수의 정보원이 제공하는 콘텐츠를 조합하여 하나의 서비스로 제공하는 웹 사이트 또는 애플리케이션을 말함
서비스 지향 아키텍처 (SOA; Service Oriented Architecture)	• 기업의 소프트웨어 인프라인 정보시스템을 공유와 재사용이 가능한 서비스 단위나 컴포넌트 중심으로 구축하는 정보기술 아키텍처 • SOA 기반 애플리케이션 구성 계층 : 표현(Presentation) 계층, 업무 프로세스(Biz-Process) 계층, 서비스 중간(Service Intermediary) 계층, 애플리케이션(Application) 계층, 데이터 저장(Persistency) 계층
디지털 트윈 (Digital Twin)	• 현실속의 사물을 소프트웨어로 가상화한 모델로, 자동차, 항공, 에너지, 국방, 헬스케어 등 여러 분야에서 주목 받음 • 실제 물리적인 자산을 소프트웨어로 가상화함으로써 실제 자산의 특성에 대한 정확한 정보를 얻을 수 있고, 자산 최적화, 돌발사고 최소화, 생산성 증가 등 설계부터 제조, 서비스에 이르는 모든 과정의 효율성을 향상시킬 수 있음
텐서플로 (TensorFlow)	• 구글의 구글 브레인(Google Brain) 팀이 만든, 다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리 • C++언어로 제작되었고, 구글 검색, 음성 인식, 번역 등의 구글 서비스 전반에서 다양하게 사용되고 있음
도커(Docker)	• 컨테이너 기술을 자동화하여 쉽게 사용할 수 있게 하는 오픈소스 프로젝트 • 소프트웨어 컨테이너 안에 응용 프로그램들을 배치시키는 일을 자동화해주는 역할을 수행함
스크래피(Scrapy)	Python 기반의 웹 크롤링 프레임워크로, 코드 재사용성을 높이는 데 도움이 되며, 대규모의 크롤링 프로젝트에 적합함

23.7, 23.5, 23.2, 22.7, 22.4, 22.3, 21.8, 21.3

핵심 347 보안 관련 용어

2418201

서비스형 블록체인(BaaS; Blockchain as a Service)	• 블록체인(Blockchain) 앱의 개발 환경을 클라우드 기반으로 제공하는 서비스 • 블록체인 네트워크에 노드의 추가 및 제거가 용이하고, 블록체인 플랫폼마다 다른 브록체인 기술을 보다 편리하게 사용할 수 있게 함
OWASP(the Open Web Application Security Project)	• 웹 정보 노출이나 악성 코드, 스크립트, 보안이 취약한 부분을 연구하는 비영리 단체 • 보안 취약점 중 보안에 미치는 영향이 큰 것을 기준으로 선정한 10대 웹 애플리케이션 취약점을 3~4년에 한 번씩 발표하고 있음
TCP 래퍼(TCP Wrapper)	• 외부 컴퓨터의 접속 인가 여부를 점검하여 접속을 허용 및 거부하는 보안용 도구 • 네트워크에 접속하면 로그인한 다른 컴퓨터 사용자의 ID 및 로그를 조회하여 악용이 가능한 데, 이것을 방지하기 위한 방화벽 역할을 수행함
허니팟 (Honeypot)	• 비정상적인 접근을 탐지하기 위해 설치해 둔 시스템 • 침입자를 속여 실제 공격을 당하는 것처럼 보여줌으로써 추적 및 공격기법에 대한 정보를 수집함
DPI(Deep Packet Inspection)	OSI 7 Layer 전 계층의 프로토콜과 패킷 내부의 콘텐츠를 파악하여 침입 시도, 해킹 등을 탐지하고, 트래픽을 조정하기 위한 패킷 분석 기술

23.7, 22.3, 21.5

핵심 348 HW 관련 신기술

2459944

고가용성 (HA; High Availability)	• 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘을 의미함 • 가용성(Availability)을 극대화하는 방법으로는 클러스터, 이중화 등이 있음
RAID(Redundant Array of Inexpensive Disk)	여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장할 경우 그 블록들을 여러 디스크에서 동시에 읽거나 쓸 수 있으므로 디스크의 속도가 매우 향상되는데, 이 기술을 RAID라고 함
앤 스크린 (N-Screen)	N개의 서로 다른 단말기에서 동일한 콘텐츠를 자유롭게 이용할 수 있는 서비스를 말함
멤스(MEMS; Micro-Electro Mechanical Systems)	초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치
트러스트존 기술 (TrustZone Technology)	칩 설계회사인 ARM(Advanced RISC Machine)에서 개발한 기술로, 하나의 프로세서(Processor) 내에 일반 애플리케이션을 처리하는 일반 구역(Normal World)과 보안이 필요한 애플리케이션을 처리하는 보안 구역(Secure World)으로 분할하여 관리하는 하드웨어 기반의 보안 기술
멤리스터 (Memristor)	메모리(Memory)와 레지스터(Resister)의 합성어로, 전류의 방향과 양 등 기존의 경험을 모두 기억하는 특별한 소자

20.9

핵심 349 Secure OS의 개요

2418601

Secure OS는 기존의 운영체제(OS)에 내재된 보안 취약점을 해소하기 위해 보안 기능을 갖춘 커널을 이식하여 외부의 침입으로부터 시스템 자원을 보호하는 운영체제를 의미한다.

- 보호 방법을 구현하기 복잡한 것부터 차례로 분류하면 다음과 같다.
  - 암호적 분리(Cryptographic Separation) : 내부 정보를 암호화하는 방법
  - 논리적 분리(Logical Separation) : 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한하는 방법
  - 시간적 분리(Temporal Separation) : 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거하는 방법
  - 물리적 분리(Physical Separation) : 사용자별로 특정 장비만 사용하도록 제한하는 방법

21.5

핵심 350 Secure OS의 보안 기능

2418602

식별 및 인증	각 접근 주체에 대한 안전하고 고유한 식별 및 인증 기능
임의적 접근통제	• 소속 그룹 또는 개인에 따라 부여된 권한에 따라 접근을 통제하는 기능 • DAC(Discretionary Access Control) 또는 신분 기반 정책이라고도 함
강제적 접근통제	• 접속 단말 및 접속 방법, 권한, 요청 객체의 특성 등 여러 보안 속성이 고려된 규칙에 따라 강제적으로 접근을 통제하는 기능 • MAC(Mandatory Access Control) 또는 규칙 기반 정책이라고도 함
객체 재사용 보호	메모리에 기존 데이터가 남아있지 않도록 초기화하는 기능
완전한 조정	우회할 수 없도록 모든 접근 경로를 완전하게 통제하는 기능
신뢰 경로	비밀번호 변경 및 권한 설정 등과 같은 보안 작업을 위한 안전한 경로를 제공하는 기능
감사 및 감사기록 축소	• 모든 보안 관련 사건 및 작업을 기록(Log)한 후 보호하는 기능 • 막대한 양의 기록들을 분석하고 축소하는 기능

23.7, 23.2, 22.4, 21.5, 20.9, 20.8, 20.6

핵심 351 DB 관련 신기술

2459945

하둡 (Hadoop) 스쿱(Sqoop)	• 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼 • 일반 PC급 컴퓨터들로 가상화된 대형 스토리지를 형성하고 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 개발된 자바 소프트웨어 프레임워크로, 구글, 야후 등에 적용되고 있음
맵리듀스 (MapReduce)	대용량 데이터를 분산 처리하기 위한 목적으로 개발된 프로그래밍 모델로, 흩어져 있는 데이터를 연관성 있는 데이터 분류로 묶는 Map 작업을 수행한 후 중복 데이터를 제거하고 원하는 데이터를 추출하는 Reduce 직업을 수행함

타조 (Tajo)	오픈 소스 기반 분산 컴퓨팅 플랫폼인 아파치 하둡(Apache Hadoop) 기반의 분산 데이터 웨어하우스 프로젝트로, 우리나라가 주도하여 개발하고 있음
데이터 마이닝 (Data Mining)	• 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법 • 대량의 데이터를 분석하여 데이터 속에 내재되어 있는 변수 사이의 상호관계를 규명하여 패턴화함으로써 효율적인 데이터 추출이 가능함
OLAP(Online Analytical Processing)	• 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식을 말함 • OLAP 연산 : Roll-up, Drill-down, Drill-through, Drill-across, Pivoting, Slicing, Dicing

23.5, 21.3, 20.8

핵심 352 회복(Recovery)

2459946

- 회복은 트랜잭션들을 수행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업이다.
- 회복 기법

연기 갱신 기법 (Deferred Update)	• 트랜잭션이 성공적으로 완료될 때까지 데이터베이스에 대한 실질적인 갱신을 연기하는 방법 • 트랜잭션이 수행되는 동안 갱신된 내용은 일단 Log에 보관됨 • 트랜잭션의 부분 완료(성공적인 완료 직전) 시점에 Log에 보관한 갱신 내용을 실제 데이터베이스에 기록함 • 트랜잭션이 부분 완료되기 전에 장애가 발생하여 트랜잭션이 Rollback되면 트랜잭션이 실제 데이터베이스에 영향을 미치지 않았기 때문에 어떠한 갱신 내용도 취소(Undo)시킬 필요 없이 무시하면 됨 • Redo 작업만 가능함
즉각 갱신 기법 (Immediate Update)	• 트랜잭션이 데이터를 갱신하면 트랜잭션이 부분 완료되기 전이라도 즉시 실제 데이터베이스에 반영하는 방법 • 장애가 발생하여 회복 작업할 경우를 대비하여 갱신된 내용들은 Log에 보관시킴 • 회복 작업을 할 경우에는 Redo와 Undo 모두 사용 가능함
그림자 페이지 대체 기법 (Shadow Paging)	• 갱신 이전의 데이터베이스를 일정 크기의 페이지 단위로 구성하여 각 페이지마다 복사본인 그림자 페이지로 별도 보관해 놓고, 실제 페이지를 대상으로 트랜잭션에 의한 갱신 작업을 하다가 장애가 발생하여 트랜잭션 작업을 Rollback 시킬 때, 갱신된 이후의 실제 페이지 부분에 그림자 페이지를 대체하여 회복시키는 기법 • 로그, Undo 및 Redo 알고리즘이 필요 없음
검사점 기법 (Check Point)	트랜잭션 실행 중 특정 단계에서 재실행할 수 있도록 갱신 내용이나 시스템에 대한 상황 등에 관한 정보와 함께 검사점을 로그에 보관해 두고, 장애 발생 시 트랜잭션 전체를 철회하지 않고 검사점부터 회복 작업을 하여 회복시간을 절약하도록 하는 기법

23.5, 23.2, 21.8, 21.5, 21.3, 20.9, 20.8, 20.6

핵심 353 병행제어 (Concurrency Control)

2459947

병행제어란 다중 프로그램의 이점을 활용하여 동시에 여러 개의 트랜잭션을 병행수행할 때, 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호 작용을 제어하는 것이다.

- 병행제어 기법의 종류

로킹 (Locking)	• 주요 데이터의 액세스를 상호 배타적으로 하는 것 • 트랜잭션들이 어떤 로킹 단위를 액세스하기 전에 Lock(잠금)을 요청해서 Lock이 허락되어야만 그 로킹 단위를 액세스할 수 있도록 하는 기법
타임 스탬프 순서 (Time Stamp Ordering)	• 직렬성 순서를 결정하기 위해 트랜잭션 간의 처리 순서를 미리 선택하는 기법들 중에서 가장 보편적인 방법 • 트랜잭션과 트랜잭션이 읽거나 갱신한 데이터에 대해 트랜잭션이 실행을 시작하기 전에 시간표(Time Stamp)를 부여하여 부여된 시간에 따라 트랜잭션 작업을 수행하는 기법 • 교착상태가 발생하지 않음
최적 병행수행 (검증 기법, 확인 기법, 낙관적 기법)	병행수행하고자 하는 대부분의 트랜잭션이 판독 전용(Read Only) 트랜잭션일 경우, 트랜잭션 간의 충돌률이 매우 낮아서 병행제어 기법을 사용하지 않고 실행되어도 이 중의 많은 트랜잭션은 시스템의 상태를 일관성 있게 유지한다는 점을 이용한 기법
다중 버전 기법	• 타임 스탬프의 개념을 이용하는 기법으로, 다중 버전 타임 스탬프 기법이라고도 함 • 타임 스탬프 기법은 트랜잭션 및 데이터들이 이용될 때의 시간을 시간표로 관리하지만, 다중 버전 기법은 갱신될 때마다의 버전을 부여하여 관리함

※ 로킹 단위(Locking Granularity)

- 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 데이터베이스, 파일, 레코드, 필드 등이 로킹 단위가 될 수 있다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고, 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아진다.

23.2, 21.5, 21.3, 20.6

2459948

핵심 354 교착상태

교착상태(Dead Lock)는 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상을 의미한다.

- 교착상태 발생의 필요 충분 조건

상호 배제 (Mutual Exclusion)	한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
점유와 대기 (Hold and Wait)	최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
비선점 (Non-preemption)	다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함
환형 대기 (Circular Wait)	공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

- 교착상태의 해결 방법

예방 기법 (Prevention)	• 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행됨 • 자원의 낭비가 가장 심한 기법
회피 기법 (Avoidance)	• 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용됨 • 은행원 알고리즘(Banker's Algorithm) : E. J. Dijkstra가 제안한 것으로, 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래한 기법
발견 기법 (Detection)	• 시스템에 교착상태가 발생했는지 점검하여 교착상태에 있는 프로세스와 자원을 발견하는 것을 의미함 • 교착상태 발견 알고리즘과 자원 할당 그래프 등을 사용할 수 있음
회복 기법 (Recovery)	교착상태를 일으킨 프로세스를 종료하거나 교착상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것을 의미함

20.8

2419101

핵심 355 Secure SDLC의 개요

Secure SDLC는 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것을 의미한다.

- Secure SDLC는 소프트웨어의 유지 보수 단계에서 보안 이슈를 해결하기 위해 소모되는 많은 비용을 최소화하기 위해 등장하였다.
- Secure SDLC는 요구사항 분석, 설계, 구현, 테스트, 유지 보수 등 SDLC 전체 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.
- Secure SDLC의 대표적인 방법론

CLASP	• Secure Software 사에서 개발하였으며, SDLC의 초기 단계에서 보안을 강화하기 위해 개발된 방법론 • 활동 중심, 역할 기반의 프로세스로 구성되어 있으며, 현재 운용 중인 시스템에 적용하기에 적합함
SDL	• 마이크로소프트 사에서 안전한 소프트웨어 개발을 위해 기존의 SDLC를 개선한 방법론 • 전통적인 나선형 모델을 기반으로 함
Seven Touchpoints	• 소프트웨어 보안의 모범사례를 SDLC에 통합한 방법론 • 설계 및 개발 과정의 모든 산출물에 대해 위험 분석 및 테스트를 수행함 • SDLC의 각 단계에 관련된 7개의 보안 강화 활동을 수행함

23.5, 23.2, 22.7, 22.4, 21.3, 20.8, 20.6

2419131

핵심 356 보안 요소

보안 요소는 소프트웨어 개발에 있어 충족시켜야할 요소 및 요건을 의미한다.

- 보안 3대 요소에는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있으며, 그 외에도 인증(Authentication), 부인 방지(NonRepudiation) 등이 있다.

기밀성	• 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용됨 • 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없음
무결성	시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
가용성	인가받은 사용자는 언제라도 사용할 수 있음
인증	• 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말함 • 대표적 방법으로는 패스워드, 인증용 카드, 지문 검사 등이 있음
부인 방지	데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공함

22.7, 21.3

2459949

핵심 357 세션 하이재킹 (Session Hijacking)

세션 하이재킹은 서버에 접속하고 있는 클라이언트들의 세션 정보를 가로채는 공격 기법으로, 세션 가로채기라고도 한다.

- 정상적인 연결을 RST(Reset) 패킷을 통해 종료시킨 후 재연결 시 희생자가 아닌 공격자에게 연결하는 방식이다.
- 공격자는 서버와 상호 간의 동기화된 시퀀스 번호를 이용하여 인가되지 않은 시스템의 기능을 이용하거나 중요한 정보에 접근할 수 있게 된다.
- 탐지 방법에는 비동기화 상태 탐지, ACK Storm 탐지, 패킷의 유실 탐지, 예상치 못한 접속의 리셋 탐지가 있다.

23.7, 22.7, 22.3, 21.8, 20.9, 20.8

2419302

핵심 358 입력 데이터 검증 및 표현의 보안 약점

SQL 삽입 SQL Injection	• 웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고, 관리자 인증을 우회하는 보안 약점 • 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지할 수 있음
경로 조작 및 자원 삽입	• 데이터 입출력 경로를 조작하여 서버 자원을 수정 · 삭제할 수 있는 보안 약점 • 사용자 입력값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지할 수 있음
크로스사이트 스크립팅(XSS)	• 웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점 • HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 '〈', '〉', '&' 등의 문자를 다른 문자로 치환함으로써 방지할 수 있음
운영체제 명령어 삽입	• 외부 입력값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점 • 웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력값을 검증 없이 내부 명령어로 사용하지 않음으로써 방지할 수 있음
위험한 형식 파일 업로드	• 악의적인 명령어가 포함된 스크립트 파일을 업로드함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점 • 업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 등의 방법으로 방지할 수 있음
신뢰되지 않는 URL 주소로 자동접속 연결	• 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점 • 연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지할 수 있음
메모리 버퍼 오버플로	• 연속된 메모리 공간을 사용하는 프로그램에서 할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 할 때 발생하는 보안 약점 • 프로그램의 오동작을 유발시키거나, 악의적인 코드를 실행시켜 공격자가 프로그램을 통제할 수 있는 권한을 획득하게 함 • 메모리 버퍼를 사용할 경우 적절한 버퍼의 크기를 설정하고, 설정된 범위의 메모리 내에서 올바르게 읽거나 쓸 수 있도록 함으로써 방지할 수 있음

20.8

핵심 359 보안 기능의 보안 약점

2419402

적절한 인증 없이 중요기능 허용	• 보안검사를 우회하여 인증과정 없이 중요한 정보 또는 기능에 접근 및 변경이 가능함 • 중요정보나 기능을 수행하는 페이지에서는 재인증 기능을 수행하도록 하여 방지할 수 있음
부적절한 인가	• 접근제어 기능이 없는 실행경로를 통해 정보 또는 권한을 탈취할 수 있음 • 모든 실행경로에 대해 접근제어 검사를 수행하고, 사용자에게는 반드시 필요한 접근 권한만을 부여하여 방지할 수 있음
중요한 자원에 대한 잘못된 권한 설정	• 권한 설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용할 수 있음 • 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않은 사용자의 중요 자원에 대한 접근 여부를 검사함으로써 방지할 수 있음
취약한 암호화 알고리즘 사용	• 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요정보를 탈취할 수 있음 • 안전한 암호화 알고리즘을 이용하고, 업무관련 내용이나 개인정보 등에 대해서는 IT보안인증사무국이 안정성을 확인한 암호모듈을 이용함으로써 방지할 수 있음
중요정보 평문 저장 및 전송	• 암호화되지 않은 평문 데이터를 탈취하여 중요한 정보를 획득할 수 있음 • 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, HTTPS 또는 SSL과 같은 보안 채널을 이용함으로써 방지할 수 있음
하드코드된 비밀번호	• 소스코드 유출 시 내부에 하드코드된 패스워드를 이용하여 관리자 권한을 탈취할 수 있음 • 패스워드는 암호화하여 별도의 파일에 저장하고, 디폴트 패스워드나 디폴드 키의 사용을 피함으로써 방지할 수 있음

21.5, 20.6

핵심 360 스택 가드(Stack Guard)

2419731

- 널 포인터 역참조와 같이 주소가 저장되는 스택에서 발생하는 보안 약점을 막는 기술 중 하나이다.
- 메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장한 후 그 값이 변경되었을 경우 오버플로우 상태로 판단하여 프로그램 실행을 중단함으로써 잘못된 복귀 주소의 호출을 막는 기술이다.

20.9, 20.6

핵심 361 접근 지정자(접근 제어자)

2419831

접근 지정자는 프로그래밍 언어에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어이다

(접근 가능 : ○, 접근 불가능 : ×).

한정자	클래스 내부	패키지 내부	하위 클래스	패키지 외부
Public	○	○	○	○
Protected	○	○	○	×
Default	○	○	×	×
Private	○	×	×	×

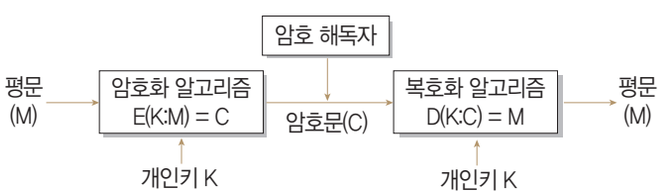
23.5, 22.4, 21.5, 21.3, 20.8

핵심 362 개인키 암호화(Private Key Encryption) 기법

2420002

개인키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화한다.

- 데이터베이스 사용자는 평문의 정보 M을 암호화 알고리즘 E와 개인키(Private Key) K를 이용하여 암호문 C로 바꾸어 저장시켜 놓으면 사용자는 그 데이터베이스에 접근하기 위해 복호화 알고리즘 D와 개인키 K를 이용하여 다시 평문의 정보 M으로 바꾸어 이용하는 방법이다.



- 개인키 암호화 기법에서 암호화 대상이 n개일 때 사용되는 키의 개수는 $\frac{n(n-1)}{2}$이다.
- 개인키 암호화 기법은 대칭 암호 기법 또는 단일키 암호화 기법이라고도 한다.
- 개인키 암호화 기법은 한 번에 하나의 데이터 블록을 암호화 하는 블록 암호화 방식과, 평문과 동일한 길이의 스트림을 생성하여 비트 단위로 암호화 하는 스트림 암호화 방식으로 분류된다.
- 종류
  - 블록 암호화 방식 : DES, SEED, AES, ARIA
  - 스트림 암호화 방식 : LFSR, RC4

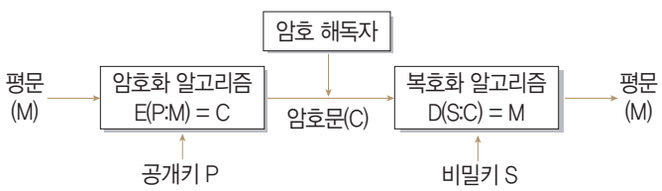
23.2, 22.4, 21.3, 20.9

핵심 363 공개키 암호화(Public Key Encryption) 기법

2420003

공개키 암호화 기법은 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다.

- 데이터베이스 사용자는 평문의 정보 M을 암호화 알고리즘 E와 공개키(Public Key) P를 이용하여 암호문 C로 바꾸어 저장시켜 놓고, 이를 복호화하기 위해서는 비밀키와 복호화 알고리즘에 권한이 있는 사용자만이 복호화 알고리즘 D와 비밀키(Secret Key) S를 이용하여 다시 평문의 정보 M으로 바꿀 수 있는 기법이다.



- 공개키 암호화 기법에서 암호화 대상이 n개일 때 사용되는 키의 개수는 2n이다.
- 공개키 암호화 기법은 비대칭 암호 기법이라고도 하며, 대표적으로는 RSA(Rivest Shamir Adleman) 기법이 있다.

대칭암호화 기법 - 암호화키와 복호화 키가 동일하다.

23.7, 23.5, 22.7, 22.3, 21.8, 20.8, 20.6

핵심 364 양방향 알고리즘 종류

2420032

개인키 암호화 방식과 공개키 암호화 방식에서 사용되는 주요 암호화 알고리즘에는 SEED, ARIA 등이 있다.

SEED	• 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류됨
ARIA (Academy, Research Institute, Agency)	• 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘 • ARIA는 학계(Academy), 연구기관(Research Institute), 정부(Agency)의 영문 앞 글자로 구성됨 • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류됨
DES(Data Encryption Standard)	• 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘 • DES를 3번 적용하여 보안을 더욱 강화한 3DES(Triple DES)도 있음 • 블록 크기는 64비트이며, 키 길이는 56비트
AES (Advanced Encryption Standard)	• 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘 • DES의 한계를 느낀 NIST에서 공모한 후 발표하였다. • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류됨
RSA (Rivest Shamir Adleman)	• 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘 • 큰 숫자를 소인수분해 하기 어렵다는 것에 기반하여 만들어짐 • 공개키와 비밀키를 사용하는데, 여기서 키란 메시지를 열고 잠그는 상수(Constant)를 의미함

ECC - 이산대수, 타원곡선

Robin - 소인수분해

23.7, 23.2, 22.7, 21.5, 21.3

핵심 365 해시(Hash) 일방향 함수

2420004

해시는 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.

- 해시 알고리즘을 해시 함수라고 부르며, 해시 함수로 변환된 값이나 키를 해시값 또는 해시키라고 부른다.
- 데이터의 암호화, 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- 해시 함수의 종류에는 SHA 시리즈, MD5, N-NASH, SNEFRU 등이 있다.

SHA 시리즈	• 1993년 미국 국가안보국(NSA)이 처음 설계했으며, 미국 국립표준기술연구소(NIST)에 의해 발표됨 • 초기 개발된 SHA-0 이후 SHA-1이 발표되었고, 다시 SHA-2라고 불리는 SHA-224, SHA-256, SHA-384, SHA-512가 발표됨
MD5	• 1991년 R.Rivest가 MD4를 대체하기 위해 고안한 암호화 해시 함수 • 블록 크기는 512비트이며, 키 길이는 128비트
N-NASH	• 1989년 일본의 전신전화주식회사(NTT)에서 발표한 암호화 해시 함수 • 블록 크기와 키 길이가 모두 128비트
SNEFRU	• 1990년 R.C.Merkle가 발표한 해시 함수 • 32비트 프로세서에서 구현을 용이하게 할 목적으로 개발됨 • 블록 크기는 512비트이며, 키 길이에 따라 128과 256으로 분류됨

21.8

핵심 366 솔트(Salt)

2420033

- 둘 이상의 계정에 대해 패스워드를 'qwer1234'라고 지정하고, 같은 암호화 알고리즘을 적용하게 되면 결과도 마찬가지로 동일하게 나타난다. 이 경우 공격자가 나타난다면 하나의 암호만 해제해도 둘 이상의 계정을 얻게 되므로, 이를 방지하고자 암호화를 수행하기에 앞서 원문에 무작위의 값을 덧붙이는 과정을 수행한다. 이때 덧붙이는 무작위의 값을 솔트(Salt)라고 한다.
- 솔트(Salt)를 사용하면 같은 패스워드에 대해 암호화를 수행하더라도 서로 다른 결과가 나타나게 되어 더 안전하게 암호화된 데이터를 관리할 수 있게 된다.

20.8

핵심 367 DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격

2420107

DDoS 공격은 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 분산 서비스 공격용 툴을 설치하여 에이전트(Agent)로 만든 후 DDoS 공격에 이용한다.

- 공격의 범위를 확대하기 위해 일부 호스트에 다수의 에이전트를 관리할 수 있는 핸들러(Handler) 프로그램을 설치하여 마스터(Master)로 지정한 후 공격에 이용하기도 한다.

분산 서비스 공격용 툴

에이전트(Agent)의 역할을 수행하도록 설계된 프로그램으로 데몬(Daemon)이라고 부르며, 다음과 같은 종류가 있다.

- Trin00 : 가장 초기 형태의 데몬으로, 주로 UDP Flooding 공격을 수행함
- TFN(Tribe Flooding Network) : UDP Flooding 뿐만 아니라 TCP SYN Flood 공격, ICMP 응답 요청, 스머핑 공격 등을 수행함
- TFN2K : TFN의 확장판
- Stacheldraht : 이전 툴들의 기능을 유지하면서, 공격자, 마스터, 에이전트가 쉽게 노출되지 않도록 암호화된 통신을 수행하며, 툴이 자동으로 업데이트되도록 설계됨

23.7, 23.5, 23.2, 22.4, 22.3, 21.8, 21.3, 20.6

핵심 368 네트워크 침해 공격 관련 용어

2459950

Ping of Death (죽음의 핑)	Ping 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법
SMURFING (스머핑)	IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법
스미싱 (Smishing)	문자 메시지(SMS)를 이용해 사용자의 개인 신용정보를 빼내는 공격 방법
피싱 (Phishing)	개인 정보(Private Data)와 낚시(Fishing)의 합성어로, 이메일이나 메신저 등을 통해 공기관이나 금융기관을 사칭하여 개인 정보를 빼내는 기법
Ping Flood	특정 사이트에 매우 많은 ICMP 메시지를 보내 이에 대한 응답(Respond)으로 시스템 자원을 모두 사용하게 해 시스템이 정상적으로 동작하지 못하도록 하는 공격 방법
Evil Twin Attack	실제 존재하는 동일한 이름의 무선 WiFi 신호를 송출하여 로그온한 사람들의 계정 정보나 신용 정보 등을 빼내는 기법
스위치 재밍 (Switch Jamming)	위조된 매체 접근 제어(MAC) 주소를 지속적으로 네트워크로 흘려보내, 스위치 MAC 주소 테이블의 저장 기능을 혼란시켜 더미 허브(Dummy Hub)처럼 작동하게 하는 공격 방법

SYN Flooding : 3-way-handshake

Land : 출발 IP와 목적지 IP를 동일하게하는 공격

p 281 Land

22.3

핵심 369 블루투스(Bluetooth) 관련 공격

2459951

블루버그 (BlueBug)	블루투스 장비 사이의 취약한 연결 관리를 악용한 공격으로, 휴대폰을 원격 조정하거나 통화를 감청할 수 있음
블루스나프 (BlueSnarf)	블루투스의 취약점을 활용하여 장비의 파일에 접근하는 공격으로, 인증없이 간편하게 정보를 교환할 수 있는 OPP(Object Push Profile)를 사용하여 정보를 열람함
블루프린팅 (BluePrinting)	공격 대상이 될 블루투스 장비를 검색하는 활동
블루재킹 (BlueJacking)	블루투스를 이용해 스팸처럼 메시지를 익명으로 퍼뜨리는 공격

23.7, 23.5, 23.2, 22.7, 22.4, 21.8, 20.6 2420109

핵심 370 정보 보안 침해 공격 관련 용어

웜 (Worm)	네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로, 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 웜 공격의 한 형태
제로 데이 공격 (Zero Day Attack)	보안 취약점이 발견되었을 때 발견된 취약점의 존재 자체가 널리 공표되기도 전에 해당 취약점을 통하여 이루어지는 보안 공격으로, 공격의 신속성을 의미함
키로거 공격 (Key Logger Attack)	컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격
랜섬웨어 (Ransomware)	인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 함
백도어 (Back Door, Trap Door)	• 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 함 • 백도어 탐지 방법 : 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등

파밍(Pharming) : 중간에서 도메인을 탈취하여 이용자를 가짜 사이트로

P284 트로이목마

23.7, 23.5, 23.2, 22.4 2459952

핵심 371 인증(認證, Authentication)

인증은 다중 사용자 컴퓨터 시스템이나 네트워크 시스템에서 로그인을 요청한 사용자의 정보를 확인하고 접근 권한을 검증하는 보안 절차이다.

- 인증에는 네트워크를 통해 컴퓨터에 접속하는 사용자의 등록 여부를 확인하는 것과 전송된 메시지의 위 · 변조 여부를 확인하는 것이 있다.
- 인증의 주요 유형
  - 지식 기반 인증(Something You Know) : 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것 (패스워드 사용)
  - 소유 기반 인증(Something You Have) : 사용자가 소유하고 있는 것을 기반으로 인증을 수행하는 것
  - 생체 기반 인증(Something You Are) : 사용자의 고유한 생체 정보를 기반으로 인증을 수행하는 것
  - 위치 기반 인증(Somewhere You Are) : 인증을 시도하는 위치의 적절성 확인하는 것

23.7, 23.5, 22.4 2459953

핵심 372 관리적/물리적/기술적 보안

효율적인 취약점 관리를 위해 관리적/물리적/기술적인 영역을 구분하여 보안 개념을 수립 및 설정하고, 발생하는 문제에 대응해야 한다.

관리적 보안	정보보호 정책, 정보보호 조직, 정보자산 분류, 정보보호 교육 및 훈련, 인적 보안, 업무 연속성 관리 등의 정의
물리적 보안	건물 및 사무실 출입 통제 지침, 전산실 관리 지침, 정보 시스템 보호 설치 및 관리 지침, 재해 복구 센터 운영 등의 정의
기술적 보안	사용자 인증, 접근 제어, PC, 서버, 네트워크, 응용 프로그램, 데이터(DB) 등의 보안 지침 정의

22.3 2420403

핵심 373 리눅스의 커널 로그

데몬	파일명	내용
kernel	/dev/console	커널에 관련된 내용을 관리자에게 알리기 위해 파일로 저장하지 않고 지정된 장치에 표시함
	var/log/wtmp	• 성공한 로그인/로그아웃에 대한 로그를 기록함 • 시스템의 시작/종료 시간에 대한 로그를 기록함
	var/run/utmp	현재 로그인한 사용자의 상태에 대한 로그를 기록함
	var/log/btmp	실패한 로그인에 대한 로그를 기록함
	var/log/lastlog	마지막으로 성공한 로그인에 대한 로그를 기록함

22.7, 21.8

핵심 374 침입 탐지 시스템(IDS; Intrusion Detection System)

2420503

침입 탐지 시스템은 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다.

- 방화벽과 같은 침입 차단 시스템만으로는 내부 사용자의 불법적인 행동과 외부 해킹에 100% 완벽하게 대처할 수는 없다.
- 문제가 발생한 경우 모든 내 · 외부 정보의 흐름을 실시간으로 차단하기 위해 해커 침입 패턴에 대한 추적과 유해 정보 감시가 필요하다.
- 오용 탐지(Misuse Detection) : 미리 입력해 둔 공격 패턴이 감지되면 이를 알려줌
- 이상 탐지(Anomaly Detection) : 평균적인 시스템의 상태를 기준으로 비정상적인 행위나 자원의 사용이 감지되면 이를 알려줌
- 침입 탐지 시스템의 종류
  - HIDS(Host-Based Intrusion Detection)
    - ▶ 시스템의 내부를 감시하고 분석하는데 중점을 둔 침입 탐지 시스템이다.
    - ▶ 내부 시스템의 변화를 실시간으로 감시하여 누가 접근해서 어떤 작업을 수행했는지 기록하고 추적한다.
    - ▶ 종류 : OSSEC, md5deep, AIDE, Samhain 등
  - NIDS(Network-Based Intrusion Detection System)
    - ▶ 외부로부터의 침입을 감시하고 분석하는데 중점을 둔 침입 탐지 시스템이다.
    - ▶ 네트워크 트래픽을 감시하여 서비스 거부 공격, 포트 스캔 등의 악의적인 시도를 탐지한다.
    - ▶ 종류 : Snort, Zeek 등
- 침입 탐지 시스템의 위치
  - 패킷이 라우터로 들어오기 전 : 네트워크에 시도되는 모든 공격을 탐지할 수 있음
  - 라우터 뒤 : 라우터에 의해 패킷 필터링을 통과한 공격을 탐지할 수 있음
  - 방화벽 뒤 : 내부에서 외부로 향하는 공격을 탐지할 수 있음
  - 내부 네트워크 : 내부에서 내부 네트워크의 해킹 공격을 탐지할 수 있음
  - DMZ : DMZ는 외부 인터넷에 서비스를 제공하는 서버가 위치하는 네트워크로, 강력한 외부 공격이나 내부 공격으로부터 중요 데이터를 보호하거나 서버의 서비스 중단을 방지할 수 있음

20.9

핵심 375 VPN(Virtual Private Network, 가상 사설 통신망)

2420507

VPN은 가상 사설 네트워크로서 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션이다.

21.5

핵심 376 SSH(Secure SHell, 시큐어 셸)

2420509

SSH는 다른 컴퓨터에 로그인, 원격 명령 실행, 파일 복사 등을 수행할 수 있도록 다양한 기능을 지원하는 프로토콜 또는 이를 이용한 응용 프로그램이다.

- 데이터 암호화와 강력한 인증 방법으로 보안성이 낮은 네트워크에서도 안전하게 통신할 수 있다.
- 키(key)를 통한 인증 방법을 사용하려면 사전에 클라이언트의 공개키를 서버에 등록해야 한다.
- 기본적으로는 22번 포트를 사용한다.