66. 소프트웨어 개발 방법론 활용

소프트웨어 개발 생명주기 모형

소프트웨어 개발 생명주기(SDLC : Software Development Life Cycle)

소프트웨어 시스템의 개발, 가동, 운용, 유지보수, 파기의 전 공정을 체계화한 개념이다.
소프트웨어 시스템의 개발부터 생애를 마치기까지의 과정에 대한 작업 프로세스를 모델화한 것이다.
소프트웨어 생명주기 또는 소프트웨어 수명주기라고도 한다.
단계별 주요 활동과 산출물을 표현함으로써 프로젝트의 관리를 쉽게 해준다.
프로젝트의 비용 산정과 개발 계획을 수립할 수 있는 기본 골격이 된다.
용어를 표준화시키고 문서화가 충실한 프로젝트 관리를 가능하게 한다.

소프트웨어 개발 생명주기 모형(SDLC Model)

소프트웨어 개발 조직이 담당하는 작업 종류와 순서, 그리고 각 단계에서 생성되는 결과물을 정리한 것이다.
소프트웨어 프로세스 모형이라고도 한다.
소프트웨어의 종류, 시스템의 규모, 난이도, 신뢰도, 조직의 규모 등에 따라 적합한 생명주기 모형이 다르다.
소프트웨어 생명주기 모형에 따라 소프트웨어의 품질을 높이고, 개발 기간과 비용을 줄일 수 있다.
소프트웨어 생명주기 모형에는 구축 및 수정 모형, 폭포수 모형, 프로토타입 모형, 나선형 모형, 애자일 모형 등이 있다.

프로토타입 모형(Prototyping Model)

실제 개발될 소프트웨어에 대한 시제품(Prototype)을 만들어 최종 결과물을 예측하는 모형이다.
요구 수집, 빠른 설계, 프로토타입 구축, 고객 평가, 프로토타입 조정, 구현의 단계를 통해 소프트웨어를 개발하는 모형이다.
최종 결과물이 만들어지기 전에 의뢰자가 최종 결과물의 일부 또는 모형을 볼 수 있다.
발주자(의뢰자)나 개발자 모두에게 공동의 참조 모델을 제공한다.
구축하고자 하는 시스템의 요구사항이 불명확한 경우 가장 적절하게 적용될 수 있다.
사용자 요구사항을 정확하게 파악하고 충실히 반영할 수 있다.
개발 단계 안에서 유지보수가 이루어지는 것으로 볼 수 있다.

폭포수 모형(Waterfall Model)

보헴(Boehm)이 제안한 고전적 생명주기 모형으로, 선형 순차적 모형이라고도 한다.

타당성 검토, 계획, 요구사항 분석, 구현, 테스트, 유지보수의 단계를 통해 소프트웨어를 개발하는 모형이다.
순차적인 접근 방법을 이용하며, 단계적 정의와 산출물이 명확하다.
각 단계의 결과가 확인되어야지만 다음 단계로 넘어간다.
개발 중 발생한 요구사항은 반영하기 어렵다.
가장 오래된 모형으로 모형의 적용 경험과 성공 사례가 많다.
나선형 모형(Spiral Model)

Boehm 이 제시하였으며, 반복적인 작업을 수행하는 모형으로 점증적 모형, 집중적 모형이라고도 한다. 완성도 높은 소프트웨어를 만들 수 있다.
여러 번의 개발 과정을 거쳐 완벽한 최종 소프트웨어를 개발하는 점진적 모형이다.
가장 큰 장점인 위험 분석 단계에서 기술과 관리의 위험 요소들을 하나씩 제거해 나감으로써 위험성 평가에 크게 의존하기 때문에 이를 발견하지 않으면 문제가 발생할 수 있다.
대규모 시스템의 소프트웨어 개발에 적합하다.
나선형 모형의 개발 단계

계획 수립(Planning) : 위험 요소와 타당성을 분석하여 프로젝트의 추진 여부를 결정한다.
위험 분석(Risk Analysis) : 개발 목적과 기능 선택, 제약 조건 등을 결정하고 분석한다.
개발 및 검증(Development) : 선택된 기능을 수행하는 프로토타입을 개발한다.
고객 평가(Evaluation) : 개발된 프로토타입을 사용자가 확인하고 추가 및 수정될 요구사항이 있으면 이를 반영한 개선 프로토타입을 만든다.
CMP(Critical Path Method)

프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는 기법이다.
노드와 간선으로 구성되며, 노드는 작업을 표시하고 간선은 작업 사이의 전후 의존 관계를 나타낸다.
박스 노드는 프로젝트의 중간 점검을 뜻하는 이정표로, 이 노드 위에 예상 완료 시간이 표시된다.
한 이정표에서 다른 이정표에 도달하기 전의 작업이 모두 완료되어야만 다음 작업의 진행이 가능하다.
모든 작업을 거치려면 2+2+3+3 (10)일, 2+3+5+4 (14)일과 같이 두 가지 작업 방식이 있으며, 짧은 작업보다 긴 작업을 선택해서 계산해야 그 시간 안에 모든 일을 처리할 수 있게 된다.
67. 소프트웨어 개발 방법론

소프트웨어 개발 방법론의 개요

소프트웨어 개발 생명주기에 소프트웨어 공학 원리를 적용한 것으로 소프트웨어 개발 전 과정에 지속적으로 작용할 수 있는 방법, 절차, 기법 등을 의미하며, 시스템 개발 주기라고도 한다.
소프트웨어 개발 과정을 정리하고 표준화하여 프로그래머 개인이 개발 과정에서의 일관성을 유지하고 프로그래머들 간의 효과적인 협업이 이루어질 수 있게 한다.
소프트웨어 개발 방법론의 목적

소프트웨어 개발 생산성 향상, 소프트웨어 품질 향상, 효과적인 프로젝트 관리, 의사소통 수단 제공
구현(Implementation)

프로그래밍 또는 코딩이라고 불리며 설계 명세서가 컴퓨터가 알 수 있는 모습으로 변환되는 과정을 의미한다.
구조적 방법론(Structured Development Methodolohy)

정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 체계적인 방법론이다.
요구사항 분석, 구조적 분석, 구조적 설계, 구조적 프로그래밍 단계로 구성된다.
쉽게 이해할 수 있고 검증할 수 있는 프로그램의 부호를 생성하는 것이 목적이다.
1970 년대까지 가장 많이 적용된 방법론이다.
시스템 분석을 위해 데이터 흐름 다이어그램(Data Flow Diagram)이 주로 사용된다.
시스템 설계를 위해 구조도(Structured Chart) 기획, 분석, 설계, 구축하는 데이터 중심의 방법론이다.
구조적 방법론의 거시적 관점 부재에서 등장하였다.
자료에 중점을 두어 자료와 프로세스를 별개의 작업으로 병행 진행한 후 서로 간의 오류를 상관 분석하여 검증한다.
정보 전략 계획(ISP), 업무 영역 분석(BAA), 업무 시스템 설계(BSD), 시스템 구축(SC) 단계로 구성된다.
객체지향 방법론(Object-oriented Engineering Methodolohy)

분석, 설계, 개발 단계에 객체지향 기법을 활용하는 방법론이다.
구조적 프로그래밍 기법의 한계와 소프트웨어 개발의 위기에서 등장하였다.
요구분석, 설계, 수현, 테스트 및 검증 단계로 구성된다.
객체지향의 기본 원칙은 캡슐화(Encapsulation), 정보 은닉(Information Hiding), 추상화(Abstraction), 상속(Inheritance), 다형성(Polymorphism)이다.
시스템 분석을 위해 유스케이스 다이어그램(UseCase Diagram)이 주로 사용된다.

시스템 설계를 위해 시퀀스 다이어그램(Sequence Diagram)이 주로 사용된다.
컴포넌트 기반 개발 방법론(CBD : Component Based Development)

재사용이 가능한 컴포넌트의 개발 또는 상용 컴포넌트들을 조합하여 애플리케이션 개발 생산성과 품질을 높이고, 시스템 유지보수 비용을 최소화할 수 있는 개발 방법 프로세스이다.
컴포넌트 단위의 개발 및 조립을 통해 정보 시스템의 신속한 구축, 변경, 확장의 용이성과 타 시스템과의 호환성을 달성하고자 하는 소프트웨어 공학 프로세스, 방법론 및 기술의 총체적 개념이다.
CBD(Component Based Development) SW 개발 표준 산출물

분석 : 사용자 요구사항 정의서, 유스케이스 명세서, 요구사항 추적표
설계 : 클래스 명세서, 사용자 인터페이스 설계서, 아키텍처 설계서, 총괄 시험 계획서, 시스템 시험 시나리오, 엔티티 관계 모형 설계서, 데이터베이스 설계서, 통합 시험 시나리오, 단위 시험 케이스, 데이터 전환 및 초기 데이터 설계서
구현 : 프로그램 코드, 단위 시험 결과서, 데이터베이스 테이블
시험 : 통합 시험 결과서, 시스템 시험 결과서, 사용자 지침서, 운영자 지침서, 시스템 설치 결과서, 인수 시험 시나리오, 인수 시험 결과서
CBD 방법론의 특징

개발 준비, 분석, 설계, 구현, 테스트, 전개, 인도 순으로 반복, 점진적 개발 프로세스를 제공하고, 시스템 설계를 위해 컴포넌트 설계서가 주로 사용된다.
컴포넌트(Component)는 DB 와 SW 의 모듈 단위로, 재사용이 가능하다.
시스템 분석을 위해 유스케이스 다이어그램(UseCase Diagram)이 주로 사용된다.
개발 기간 단축으로 인한 생산성이 향상되며 새로운 기능 추가가 쉬워 확장성이 높다.
소프트웨어 재사용(Software Reuse)

SW 개발의 품질과 생산성을 높이기 위한 방법으로, 이미 개발되어 안정화된 SW 의 전체 혹은 일부분을 다른 SW 개발이나 유지에 사용하는 것이다.
기존에 개발된 SW 와 경험, 지식 등을 새로운 SW 에 적용한다.
클래스, 객체 등의 소프트웨어 요소는 소프트웨어 재사용성을 크게 향상했다.
소프트웨어 부품(모듈)의 크기가 작고 일반적인 설계일수록 재사용률이 높다.
합성 중심(Composition-Based)
전자칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞춰 소프트웨어를 완성시키는 방법으로, 블록 구성 방법이라고도 한다.
생성 중심(Generation-Based)

추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법으로, 패턴 구성 방법이라고도 한다.

68. 비용 산정 모델

비용 산정 모델의 종류

전문가 감정 기법, 델파이(Delphi) 기법, LOC(Line Of Code) 기법, COCOMO(COnstructive CostMOdel) 모델, Putnam 모델, 기능 점수(FP: Functional Point)

전문가 감정 기법

개발 조직 내에 경험이 많은 2 인 이상의 전문가에게 비용 산정을 의뢰하는 기법이다.
의뢰자의 신뢰도가 높고 편리하게 비용을 산정할 수 있다.
과거 프로젝트와의 유사성이 낮을 수 있다.
전문가에 따라 감정의 편차가 클 수 있다.

델파이(Delph)기법

산정 요원과 조정자에 의해 산정하는 기법이다.
전문가가 독자적으로 감정할 때 발생할 수 있는 편차를 줄이기 위해 단계별로 전문가들의 견해를 조정자가 조정하여 최종 견적을 결정한다.
유사한 프로젝트 경험을 가진 전문가 집단을 구성하여 규모, 공수, 비용의 산정 의견을 구한다.
의견 일치가 이뤄지지 않을 경우 의견의 근거를 익명으로 집단 내에 배포하고 자신들의 산정을 수정할 수 있도록 한다.

LOC(Line Of Code) 기법

소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.
예측치 = a + (4 X C) + b / 6 (단 a 는 낙관치, b 는 비관치, c 는 기대치임)
ex. 규모 추정이 a: 60, b: 200, c:100 인 경우 LOC 는 다음과 같다.

```
LOC = 60 + (4 X 100) + 200 / 6 = 660 / 6 =  110
```

개발 기간 = 예측된 LOC / (개발자 수 X 1 인달 월 평균 생산 LOC)
LOC 기법에 의해 예측된 총 라인 수가 36,000 라인, 개발에 참여할 프로그래머가 6 명, 프로그래머들의 평균 생산성이 월간 300 라인일 때 개발에 소요되는 기간은 아래와 같다.

```
개발 기간 = 36,000 / (6 X 300)
          = 36,000 / 1,800
         = 20[개월]
```

COCOMO(COnstructive COst MOdel) 모델

보헴(Boehm)이 제안한 소스 코드(Source Code)의 규모에 의한 비용 예측 모델이다.

같은 규모의 소프트웨어라도 그 유형에 따라 비용이 다르게 산정된다.

소프트웨워 프로젝트 유형에 따라 다르게 책정되는 비용 산정 수식(Equation)을 이용한다.

산정 결과는 프로젝트를 완성하는데 필요한 MM(Man-Month)으로 나타난다.

프로젝트 특성을 15 개로 나누고 각각에 대한 승수 값을 제시한다.

개발 노력 승수(Development Effort Multipliers)를 결정한다.

비용 견적의 강도 분석 및 비용 견적의 유연성이 높아 소프트웨어 개발비 견적에 널리 통용되고 있다.

COCOMO 모형 종류 : Basic COCOMO, Intermediate COCOMO, Detailed COCOMO

소프트웨어 개발 유형

Organic Mode(단순형)
5 만 라인 이하의 소프트웨어를 개발하는 유형
기관 내부에서 개발된 중소 규모의 소프트웨어로 일괄 자료 처리나 과학기술 계산용, 비즈니스 자료 처리 등
노력(MM) = 2.4 X (KDSI) 1.05
Semi-Detached Mode(중간형)
30 만 라인 이하의 소프트웨어를 개발하는 유형
트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등
노력(MM) = 3.0 X (KDSI)1.12
Embeded Mode(임베디드형)
30 만 라인 이상의 소프트웨어를 개발하는 ㅇ뉴형
초대형 규모의 트랜잭션 처리시스템이나 운영체제 등
노력(MM) = 3.6 X (KDSI)1.20
KDSI(Kilo Delivered Source Instruction) : 전체 라인 수를 1,000 단위로 묶은 것으로, KLOC 와 같은 의미이다.

Putnam 모델

Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정 기법이다.
소프트웨어 개발 생명주기의 전 과정 동안에 사용될 노력의 분포를 예측한다.
SLIM : Rayleigh-Norden 곡선과 Putnam 의 모형에 기반을 둔 자동화 추정 도구이다.
기능 점수(FP : Functional Point)

시스템을 구현한 기술에 의존적이고 개발자에 의해 식별되는 기능에 기반하여 시스템의 크기를 측정하는 척도이다.
기능 점수는 소프트웨어 시스템이 가지는 기능을 정량화 한 것이다.
입력, 출력, 질의, 파일, 인터페이스의 개수로 소프트웨어의 규모를 표현한다.
경험을 바탕으로 단순, 보통, 복잡한 정도에 따라 가중치를 부여한다.
프로젝트의 영향도와 가중치의 합을 이용하여 실질 기능 점수를 계산한다.
기능 점수의 산출 시 적용되는 가중치는 시스템의 특성에 따라 달라질 수 있다.
기능 점수 비용산정 요소 : 코드 라인 수, 데이터 파일 수, 문서 페이지 수, 입력 유형의 수, 출력 보고서의 수, 외부 루틴과의 인터페이스 수, 명령어(사용자 질의 수)
기능별 가중치

소프트웨어 기능 증대 요인	가중치		
	단순	보통	복잡
입력(입력 양식)	3	4	6
출력(출력 보고서)	4	5	7
명령어(사용자 질의 수)	3	4	5
데이터 파일	7	10	15
인터페이스	5	7	10

69. 소프트웨어 개발 표준
ISO/IEC 12199

패키지 소프트웨어의 일반적인 품질 요구사항 및 테스트를 위한 국제 표준이다.
ISO/IEC 25051 로 대체되었다.
ISO/IEC 12207

소프트웨어 개발 작업에 일관적이고 체계적인 프레임워크를 제공하기 위해 1995 년에 ISO/IEC 에서 제정한 소프트웨어 생명주기 프로세스 국제 표준이다.
기본 생명주기 프로세스 구분 : 획득 프로세스(Acquisition Process), 공급 프로세스(Supply Process), 개발 프로세스(Development Process), 운영 프로세스(Operation Process), 유지보수(Maintenance)
SPICE(Software Process Improvement And Capability dEtermination)

소프트웨어 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준이다.

공식 명칭은 ISO/IEC 15504 이다.
ISO/IEC 12207 의 단점을 해결하기 위해 개발되었다.
SPICE 모델의 범주
고객-공급자 프로세스
스프트웨어를 개발하여 고객에게 전달하는 것을 지원하고, 소프트웨어를 정확하게 운용하고 사용하도록 하기 위한 프로세스로 구성된다.
10 개의 프로세스로 구성된다.
공학 프로세스
시스템과 소프트웨어 제품을 직접 명세화, 구현, 유지보수하는 프로세스로 구성된다.
9 개의 프로세스로 구성된다.
지원 프로세스
소프트웨어 생명주기에서 다른 프로세스에 의해 이용되는 프로세스로 구성된다.
4 개의 프로세스로 구성된다.
관리 프로세스
소프트웨어 생명주기에서 프로젝트 관리자에 의해 사용되는 프로세스로 구성된다.
4 개의 프로세스로 구성된다.
조직 프로세스
조직의 업무 목적을 수립하고, 조직이 업무 목표를 달성하는데 도움을 주는 프로세스로 구성된다.
9 개의 프로세스로 구성된다.
SPICE 모델의 레벨

레벨 5 최적(Optimizing) 단계
정의된 프로세스와 표준 프로세스가 지속적으로 개선되는 단계이다.
레벨 4 예측(Predictable) 단계
표준 프로세스 능력에 대해 정량적인 이해화 성능이 예측되는 단계이다.
레벨 3 확립(Established) 단계
표준 프로세스를 사용하여 계획되고 관리된 단계이다.
레벨 2 관리(Managed) 단계
프로세스가 정해진 절차에 따라 이뤄져 산출물을 내며, 모든 작업이 계획되고 추적되는 단계이다.
레벨 1 수행(Performed) 단계
해당 프로세스의 목적은 달성하지만 계획되거나 추적되지 않은 단계이다.
레벨 0 불완전(Incomplete) 단계
프로세스가 구현되지 않거나 프로세스 목적을 달성하지 못한 단계이다.
CMM(Capability Maturity Model, 능력 성숙도 모델)

조직의 업무 능력 평가 기준을 세우기 위한 평가 표준이다.
1991 년 카네기 멜런대학이 미국국방부의 의뢰를 받아 개발한 평가 모델이다.
소프트웨어 개발 능력 측정 기준과 소프트웨어 개발 조직의 성숙도 수준을 평가한다.
이후 CMM 은 CMMI 로 발전했다.
CMM 모델의 레벨 및 핵심 프로세스

레벨 5 최적(Optimizing) 단계
프로세스 변경 관리
기술 변경 관리
결함 방지
레벨 4 관리(Managed) 단계
소프트웨어 품질 관리
정량적 프로세스 관리
레벨 3 정의(Defined) 단계
조직 프로세스 집중
조직 프로세스 정의
동료 검토
교육 프로그램
교육 간 협력
레벨 2 반복(Repeatable) 단계
소프트웨어 프로젝트 계획
소프트웨어 프로젝트 추적 및 감독
소프트웨어 하청 관리
소프트웨어 품질 보증
소프트웨어 형상 관리
요구 관리
레벨 1 초보(Initial) 단계
.
CMMI(Capability Maturity Model Integration, 능력 성숙도 통합 모델)

조직의 개발 프로세스 역량 성숙도를 평가하는 표준이다.
CMM 은 소프트웨어 개발 프로세스의 성숙도를 다루고, CMMI 는 소프트웨어, 시스템,
프로덕트를 포함하는 세 분야를 통합 평가하는 모델이다.
24 개 프로세스 영역을 4 개 범주로 분할한다.
70. 테일러링과 프레임워크
소프트웨어 개발 방법론 테일러링(Tailoring)의 개념

기존 개발 방법론의 절차, 기법, 산출물 등을 프로젝트 상황에 맞게 수정하는 작업이다.
소프트웨어 개발 방법론 테일러링 수행 절차
프로젝트 특징 정의 => 표준 프로세스 선정/검증 => 상위 레벨 커스터마이징 => 세부 커스터마이징 => 테일러링 문서화
소프트웨어 개발 방법론 테일러링 시 고려사항

내부적 요건(내부 기준)
납기/비용 : 개발 소프트웨어의 납기일과 개발 비용
구성원 능력 : 개발에 참여하는 구성원 개개인의 능력
목표 환경 : 시스템의 개발 환경 및 유형이 서로 다른 경우
고객 요구사항 : 프로젝트의 생명주기 활동 측면에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우
프로젝트 규모 : 사업비, 참여 인력, 개발 기간 등 프로젝트의 규모가 서로 다른 경우
보유 기술 : 프로세스, 방법론, 산출물, 인력의 숙련도 등이 다른 경우
외부적 요건(외부 기준)
법적 제약사항 : 프로젝트별로 적용될 IT Compliance 가 서로 다른 경우 테일러링이 필요
표준 품질 기준 : 금융, 제조, 의료 업종별 표준 품질 기준이 상이하므로 방법론의 테일러링이 필요
소프트웨어 개발 방법론 테일러링 기법

프로젝트 규모와 복잡도에 따른 테일러링
프로젝트 구성원에 따른 테일러링
팀 내 방법론 지원에 따른 테일러링
자동화에 따른 테일러링
소프트웨어 프레임워크(Framework)의 개념

비슷한 유형의 응용 프로그램들을 위해 재사용이 가능한 아키텍처와 협력하는 소프트웨어 산출물의 통합된 집합이다.
특정 클래스의 재사용뿐만 아니라 응용 프로그램을 위한 핵심 아키텍처를 제공하여 설계의 재사용을 지원한다.
소프트웨어 개발 프레임워크의 개념

소프트웨어 개발을 도와주는 재사용이 가능한 클래스와 패턴의 집합이다.
소프트웨어 개발의 효율성을 높이고 소프트웨어 품질을 높이기 위한 반제품 성격의 소프트웨어이다.
소프트웨어의 틀과 구조를 결정하고, 이를 바탕으로 개발된 개발자의 코드를 제어한다.

소프트웨어 개발 프레임워크 적용 시

개발 용이성
공통 기능은 프레임워크가 제공한다.
패턴 기반 개발과 비즈니스 로직에만 집중한 개발이 가능하다.
시스템 복잡도 감소
시스템의 복잡한 기술은 프레임워크에 의해 숨겨진다.
미리 잘 정의된 기술 셋을 적용할 수 있다.
이식성
플랫폼 연동을 프레임워크가 제공한다.
플랫폼의 독립적인 개발이 가능하다.
품질 보증
검증된 개발 기술과 패턴에 따른 개발이 가능하다.
개발자의 경험과 능력 차이를 줄여준다.
운영 용이성
소프트웨어 변경이 용이하다.
비즈니스 로직 및 아키텍처 파악이 용이하다.
개발 코드 최소화
공통 컴포넌트와 서비스를 활용한다.
반복적인 코드 개발을 최소화한다.
변경 용이성
잘 구조화된 아키텍처를 적용한다.
플랫폼에 독립적이다.
설계 및 코드의 재사용성
프레임워크의 서비스와 패턴을 재사용한다.
이미 개발된 컴포넌트를 재사용한다.
스프링 프레임워크(Spring Framework)

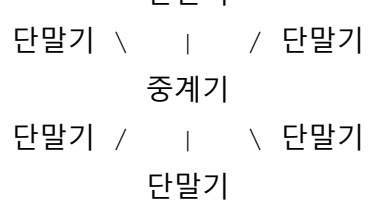
자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크이다.
동적인 웹 사이트 개발을 위해 여러 가지 서비스를 제공하고 있다.
전자정부 표준 프레임워크 기반 기술로 사용된다.
스프링 프레임워크의 주요 모듈
제어 반전 컨테이너
관점 지향 프로그래밍 프레임워크
데이터 액세스 프레임워크
트랜잭션 관리 프레임워크

모델 - 뷰 - 컨트롤러(MVC) 패턴
배치 프레임워크
전자정부 표준 프레임워크

공공부문 정보화 사업 시 플랫폼별 표준화된 개발 프레임워크를 말한다.
공공기관의 웹 서비스 개발 시 사용을 권장하고 있다.
전자정보 표준 프레임워크 적용 시 기대효과
전자정부 서비스 품질 향상
정보화 투자 효율성 향상
국가 정보화 투자 효율성 재고
중소 SI 업체 경쟁력 확보
선진 국가정보화 추진 기반 환경 재고
닷넷 프레임워크(.NET Framework)

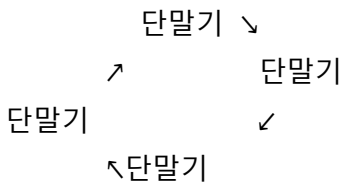
Microsoft 사에서 개발한 윈도우 프로그램 개발 및 실행 환경이다.
네트워크 작업, 인터페이스 등의 많은 작업을 캡슐화하였고, 공통 언어 런타임(CLR : Common Language Runtime) 가상 머신 위에서 작동한다.
오픈소스 버전으로 닷넷 코어가 있다.
71. 네트워크 구성
네트워크 구성
성형(Star Topology)

중앙에 호스트 컴퓨터(Host Computer)가 있고 이를 중심으로 터미널(Terminal)들이 연결되는 중앙 집중식의 네트워크 구성 형태이다.
중앙 컴퓨터와 직접 연결되어 응답이 빠르고 통신 비용이 적게 소요되지만, 중앙 컴퓨터에 장애가 발생하면 전체 시스템이 마비되는 분산 시스템의 위상 구조이다.



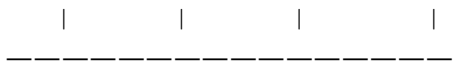
링형(Ring Topology)

서로 이웃한 컴퓨터와 노드끼리 연결한 네트워크 구성 형태이다.
각 노드가 공평한 서비스를 받으며, 전송 매체와 노드의 고장 발견이 쉽다.
데이터가 한 방향으로 전송되기 때문에 충돌(Collision) 위험이 없다.



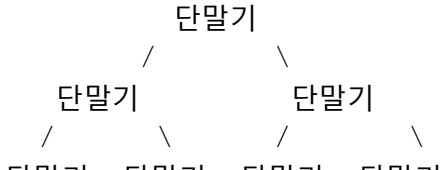
버스형(Bus Topology)

한 개의 통신 회선에 여러 개의 노드가 연결된 형태이다.
한 사이트의 고장은 나머지 사이트 간의 통신에 아무런 영향을 주지 않는다.
네트워크 트래픽이 많을 경우 네트워크 효율이 떨어진다.



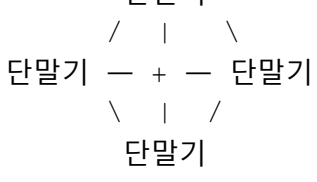
트리형(Tree Topology)

하나의 노드에 여러 개의 노드를 연결한 네트워크 구성 형태로 네트워크 관리가 용이하다.
각 노트가 계층적으로 구성되어 있어 계층형 또는 분산형이라고도 한다.



망형(Mesh Topology)

모든 컴퓨터와 노드들이 서로 연결된 네트워크 구성 형태로 그물형이라고도 한다.
응답 시간이 빠르고 노드의 연결성이 우수하다.
일반적으로 많은 단말기로부터 많은 양의 통신을 필요로 하는 경우에 유리하다.
n 개의 노드를 망형으로 구성 시 n(n-1)/2 개의 회선이 필요하다.



브릿지(Bridge)

데이터 링크 계층(Data Link Layer)에서 동작하며 같은 MAC(Midea Access Control) 프로토콜을 사용하는 근거리 통신망 사이를 연결하는 통신 장치이다.

스위치(Switch)

브릿지와 같이 두 개 이상의 LAN 을 연결하여 하나의 네트워크로 만드는 장치이다.

보안 및 트래픽 관리 기능도 제공할 수 있다.
스위치 분류
L2 스위치
OSI 2 계층에 속하는 네트워크 장비
가장 기본적인 스위치로, 단순히 스위치라고도 한다.
Adress Learning, Filtering, Forwarding 등의 기능을 수행한다.
L3 스위치
OSI 3 계층에 속하는 네트워크 장비
L2 스위치에 라우팅 기능이 추가되었다.
서로 다른 네트워크를 연결할 수 있다.
L4 스위치
OSI 4 계층에 속하는 네트워크 장비
L3 스위치에 로드밸런서(Load Balancer)가 추가되었다.
L7 스위치
OSI 7 계층에 속하는 네트워크 장비이다.
세밀한 로드밸런싱이 가능하다.
보안 기능을 대폭 강화하였다.
라우터(Router)

네트워크 계층(Network Layer)에서 동작하며 동일 전송 프로토콜을 사용하는 분리된 2 개 이상의 네트워크를 연결해주는 통신 장치이다.
네트워크상에서 가장 최적의 IP 경로를 설정하여 전송하는 장비이다.
게이트웨이(Gateway)

서로 다른 통신 프로토콜을 사용하는 네트워크 사이를 연결하여 데이터를 교환할 수 있도록 하는 역할을 한다.
두 개의 서로 다른 형태의 네트워크를 상호 연결시켜 주는 관문 역할을 하는 장치이다.
필요한 경우 프로토콜 변환을 수행한다.
VLAN(Virtual Local Area Network)

물리적 배치와 상관없이 논리적으로 LAN 을 구성하여 Broadcast Domain 을 구분할 수 있게 해주는 기술로 접속된 장비들의 성능 향상 및 보안성 증대 효과를 목표로 한다.
72. 네트워크 관련 신기술
RIP(Routing Information Protocol)

최단 경로 탐색에 Bellman-Ford 알고리즘을 사용하는 거리 벡터 라우팅 프로토콜이다.

최적의 경로를 산출하기 위한 정보로서 홉(거릿값)만을 고려하므로, RIP 를 선택한 경로가 최적의 경로가 아닌 경우가 많이 발생할 수 있다.
최대 홉 카운트를 15 홉 이하로 한정한다.
소규모 네트워크 환경에 적합하다.
OSPF(Open Shortest Path First Protocol)

대표적인 링크 상태(Link State) 라우팅 프로토콜로, IP 패킷에서 89 번 프로토콜을 사용하여 라우팅 정보를 전송하며 안정되고 다양한 기능으로 가장 많이 사용되는 것은 IGP(Interior Gateway Protocol)이다.
MQTT(Message Queuing Telemetry Transport)

IBM 이 주도하여 개발한 기술로 사물 인터넷과 같이 대역폭이 제한된 통신 환경에 최적화하여 개발된 푸시 기술 기반의 경량 메시지 전송 프로토콜이다.
TCP/IP 기반 네트워크에서 동작하는 발생-구독 기반의 메시징 프로토콜로 최근 IoT 환경에서 자주 사용되고 있는 프로토콜이다.
사물 인터넷(IoT : Internet of Things)

인터넷에 연결된 기기가 사람의 개입 업이 상호 간에 알아서 정보를 주고받아 처리한다.
사물은 물론이고 현실과 가상세계의 모든 정보와 상호 작용하는 개념이다.
WSN(Wireless Sensor Network)

센서를 네트워크로 구성한 것이다.
사물에 부착된 센서를 통해 탐지된 사물의 인식 정보는 물론 주변의 온도, 습도와 같은 환경 정보를 실시간으로 네트워크와 연결하여 수집하고 관리하는 네트워크 시스템이다.
클라우드 컴퓨팅(Cloud Computing)

사용자가 인터넷 등을 통해 하드웨어, 소프트웨워 등의 컴퓨팅 자원을 원격으로 필요한 만큼 빌려서 사용하는 방식의 서비스 기술로서 서비스 모델은 IaaS, PaaS, SaaS 로 구분한다.
가상화 기술, 서비스 프로비저닝(Provisioning) 기술, 과금 체계 등을 필요로 한다.
PaaS-TA : 국내 IT 서비스 경쟁력 강화를 목표로 개발, 인프라 제어 및 관리 환경, 실행 환경, 개발 환경, 서비스 환경, 운영환경으로 구성되어 있는 개방형 클라우드 컴퓨팅 플랫폼이다.
그리드 컴퓨팅(Grid Computing)

인터넷상에서 사용하지 않는 시간대의 연결된 수많은 컴퓨터를 하나의 고성능 컴퓨터처럼 활용할 수 있는 기술이다.
RFID(Radio Frequency IDentification)

전자 태그가 부착된 IC 칩과 무선 통신 기술을 이용하여 다양한 개체들의 정보를 관리할 수 있는 센서 기술이다.
NFC(Near Field Communication)

RFID 기술 중 하나로, 10cm 정도로 가까운 거리에서 장치 간에 양방향 무선 통신을 가능하게 해주는 기술이다. 13.56MHz 의 주파수 대역을 사용하는 비접촉식 통신 기술이다.
데이터 읽기와 쓰기 기능을 모두 사용할 수 있다.
WPAN(Wireless Personal Area Network)

사용자를 줓심으로 작은 지역에서 주로 블루투스 헤드셋, 스마트 워치 등과 같은 개인화 장치들을 연결시키는 무선 통신 규격이다.
IEEE 802.15 규격의 범주에 속한다.
PICONET(피코넷)

여러 개의 독립된 통신 장치가 UWB(Ultra Widebacn) 기술 또는 블루투스 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술이다.
스마트 그리드(Smart Grid)

전기 및 정보통신 기술을 활용하여 전력망을 지능화, 고도화함으로써 고품질의 전력 서비스를 제공하고 에너지 이용 효율을 극대화하는 전력망 시스템이다.
기존의 전력망에 정보 기술을 접목하여 전력 공급자와 소비자가 쌍방향으로 실시간 정보를 교환함으로써 에너지 효율을 최적화하고 새로운 부가가치를 창줄한다.
비콘(Beacon)

블루투스 4.0(BLE) 프로토콜 기반의 근거리 무선 통신 장치로 최대 70m 이내의 장치들과 교신할 수 있는 차세대 스마트폰 근거리 통신 기술이다.
저전력으로 모방리 결제 등을 가능하게 해주는 스마트폰 근거리 통신 기술이다.
NFC 보다 가용거리가 길고 5~10cm 단위 구별이 가능해 정확성이 높다.
포스퀘어(Foursquare)

스마트폰에 탑재된 GPS 를 활용해 위치 정보를 수집한다.
쇼핑 관광 등에 활용하는 위치 기반 소셜 네트워크 서비스이다.
ZigBee

IEEE 802.15.4 기반 PAN 기술이다.
낮은 전력을 소모하면서 저가의 센서 네트워크 구현에 최적의 방안을 제공한다.
빌딩 자동화나 홈 보안 시스템 등의 자동화에 적절한 통신 기술이다.
Mesh Network

기존 무선 랜의 한계 극복을 위해 등장하였다.
대규모 디바이스의 네트워크 생성에 최적화되어 차세대 이동통신 홈네트워킹, 공공 안전 등의 특수 목적을 위한 새로운 방식의 네트워크 기술이다.
Wavelength Division Multiplexing(WDM, 파장 분할 다중화)

레이저 빛이 다른 파장(다른 색)을 사용하여 여러 반송파 신호를 단일 광섬유에 적용하는 기술이다.
파장이 서로 다른 복수의 광신호를 동시에 이용하는 것으로 광섬유를 다중화하는 방식이다.
빛의 파장 축과 파장이 다른 광선은 서로 간섭을 일으키지 않는 성질을 이용한다.
73. SW/HW 관련 신기술
소프트웨어 관련 신기술
소프트웨어 정의 데이터 센터(SDDC : Software Defined Data Center)

가상 데이터 센터(Virtual Data Center : VDC)라고도 하며, 추상화, 풀링(Pooling), 자동화 등을 통해 인프라를 가상화하는 데이터 센터를 의미한다.
컴퓨팅, 네트워킹, 스토리지, 관리 등을 모두 소프트웨어로 정의한다.
인력 개입 없이 소프트웨어 조작만으로 자동 제어 관리한다.
데이터 센터 내 모든 자원을 가상화하여 서비스한다.
텐서플로우(TensorFlow)

구글에서 개발해서 공개한 인공지능 응용 프로그램 개발용 오픈소스 프레임워크이다.
텐서플로우를 사용할 때 인공지능 소프트웨어가 이미지 및 음성을 인식하기 위해서는 신경망의 합성곱 신경망 모델을 주로 사용한다.
머신 러닝(Machine Learning)

컴퓨터 프로그램이 데이터와 처리 경험을 이용한 학습을 통해 정보 처리 능력을 향상시키는 기술로 컴퓨터에게 많은 데이터를 주고 거기에서 일반적인 패턴을 찾아내게 한다.
자율 주행 자동차, 필기체 문자 인식 등과 같이 알고리즘 개발이 어려운 문제의 해결에 유용하다.
딥 러닝(Deep Learning)

일반적인 머신 러닝 모델보다 더 싶은 신경망 계층 구조를 이용하는 머신 러닝이다.
주로 여러 개의 은닉층(Hidden Layer)으로 구성된 인공 신경망을 활용한다.
디지털 트윈(Digital Twin)

물리적인 사물과 컴퓨터에 동일하게 표현되는 가상 모델이다.

실제 물리적인 자산 대신 소프트웨어로 가상화한 자산의 디지털 트윈을 만들어 시뮬레이션함으로써 실제 자산의 특성에 대한 정확한 정보를 얻을 수 있다.

HMD(Head Mounted Display)

머리에 착용하는 형태의 디스플레이로 HMD 장치를 머리에 쓰면 양쪽 눈에 근접한 위치에 소형 디스플레이가 있어 시차를 이용한 3D 영상이 투영된다.

블록체인(Blockchain)

공공 거래 장부로, 가상 화폐로 거래할 때 발생할 수 있는 해킹을 막는 기술이다.
하나의 블록은 트랜잭션의 집합과 헤더(Header)로 이루어져 있고 한 블록에는 앞의 블록에 대한 정보가 포함되어 있어, 앞 블록의 내용을 변경하면 뒤에 이어지는 블록도 변경해야 한다.

BaaS(Backend as a Service)

블록체인(Blockchain) 개발 환경을 클라우드로 서비스하는 개념으로 블록체인 네트워크에 노드의 추가 및 제거가 용이하다.
블록체인의 기본 인프라를 추상화하여 블록체인 응용 프로그램을 만들 수 있는 클라우드 컴퓨팅 플랫폼이다.

분산 원장 기술(Distributed Ledger Technology)

분산 네트워크 참여자가 암호화 기술을 사용하여 거래 정보를 검증하고 합의한 원장(Ledger)을 공동으로 분산/관리하는 기술이다.
수많은 사적 거래 정보를 개별적 데이터 블록으로 만들고, 이를 체인처럼 연결하는 블록체인 기술이다.

증강현실(AR : Augmented Reality)

현실을 기반으로 가상 정보를 실시간으로 결합하여 보여주는 기술이다.
예를 들어 스마트폰 카메라로 주변을 비추면 인근에 있는 상점의 위치, 전화번호 등의 정보가 입체 영상으로 표시된다.

매시업(Meshup)

웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술이다.
다수의 정보원이 제공하는 콘텐츠를 조합하여 하나의 서비스로 제공한다.
구글 지도에 부동산 매물 정보를 결합한 구글의 하우징 맵스(HousingMaps)가 대표적이다.

양자 암호(Quantum Cryptography)

양자 역학의 특성을 이용하여 안전하게 정보를 보호하기 위한 알고리즘 또는 정보 이론적/수학적 방법론이다.
양자 컴퓨터가 등장하면서 기존의 대칭키 암호 기법과 비대칭키 암호 기법은 안전성을 보장할 수 없게 되었다.
대표적인 양자 암호 기법으로 양자 암호키 분배(QKD : Quntum Key Distribution) 기법이다.
하드웨어 관련 신기술
양자 컴퓨터(Quantum Computer)

양자 역학적 현상을 이용하여 연산을 수행하는 컴퓨터이다.
양자 정보의 최소 단위인 큐비트(Qubit)의 상태를 제어하여 연산과 양자 알고리즘을 수행한다.
4D 프린팅

미리 설계된 시간이나 임의환경 조건이 충족되면 스스로 모양을 변경 또는 제조하여 새로운 형태로 바뀌는 제품을 3D 프린팅하는 기술이다.
온도, 습도, 진동 등 에너지에서 자극을 받으면 모양이 변하는 스마트 소재가 사용된다.
N-Screen

동일한 콘텐츠를 PC, 스마트 TV, 스마트폰, 태블릿 PC 등 다양한 디지털 정보기기에서 자유롭게 이용할 수 있는 서비스이다.
74. 데이터베이스 관련 기술 용어
RAID(Redundant Array of Indexpensive Disks)

데이터를 복수 또는 분할 저장하여 병렬로 데이터를 읽는 보조 기억 장치 또는 그 방법으로 디스크의 고장에 대비하여 데이터의 안정성을 높이는 기술이다.
한 개의 데이터를 여러 디스크에 저장하여 데이터 안정성을 향상시키기 위해 사용한다.
다수의 디스크에 데이터를 분할하여 전송함으로써 전체적인 데이터 전송 속도 향상을 위해 사용한다.
RAID 1 : 디스크 스트라이핑(Disk Striping) 방식으로 중복 저장과 오류 검출 및 교정이 없다.
RAID 2 : 비트 단위로 분산 저장하고 여러 개의 해밍코드 검사 디스크를 사용한다.
디스크 미러링(Disk Mirroring) 방식으로 높은 신뢰도를 갖는다.
RAID 3 : 데이터를 다수의 디스크에 스트라이핑하여 저장하며, 하나의 드라이브에 패리티를 저장한다. 패리티 드라이브를 사용한다.
RAID 4 : 각 디스크에 데이터를 블록 단위로 분산 저장하고 하나의 패리티 검사 디스크를 사용한다.(블록 인터리브된 패리티(Block-Interleaved Striping with Parity)).

RAID 5 : 별도의 패리티 디스크 대신 모든 디스크에 패리티 정보를 나누어 기록하는 방식으로 3 개 이상의 디스크 어레이를 요구하며 쓰기 작업이 많지 않은 다중 시스템에 적합하다.

웨어러블 컴퓨팅(Wearable Computing)

컴퓨터를 옷이나 안경처럼 착용할 수 있게 해주는 기술이다.
소형화, 경량화를 비롯해 음성과 동작 인식 등 다양한 기술이 적용되어 장소에 구애받지 않고 컴퓨터를 활용할 수 있다.

멤리스터(Memristor)

메모리와 레지스터의 합성어로, 전류의 방향과 크기 등 기준의 상태를 모두 기억하는 소자이다.
레지스터, 커패시터, 인덕터에 이어 네 번째 전자회로 구성 요소로 차세대 기억 소자, 회로 등에 응용될 수 있다. 에너지 소모와 부팅 시간을 획기적으로 줄일 수 있다.

직접 연결 저장 장치(DAS : Direct-Attached Storage)

하드디스크와 같은 데이터 저장 장치를 호스트 버스 어댑터에 직접 연결하는 방식이다.
저장 장치와 호스트 기기 사이에 네트워크 디바이스가 있지 말아야 한다.

SAN(Storage Area Network)

네트워크상에 광 채널 스위치의 이점인 고속 전송과 장거리 연결 및 멀티 프로토콜 기능을 활용하여 각기 다른 운영체제를 가진 여러 기종이 네트워크상에서 동일 저장 장치의 데이터를 공유하게 함으로써, 여러 개의 저장 장치나 백업 작비를 단일화시킨 시스템이다.

NAS(Network Attached Storage)

컴퓨터에 직접 연결하지 않고 네트워크를 통해 데이터를 주고받는 저장 장치이다.
구조적으로는 스토리지 서버를 단순화, 소형화한 것이다.

Software Defined Storage

가상화를 적용하여 필요한 공간만큼 나눠 사용할 수 있도록 하며, 서버 가상화와 유사하다.
컴퓨팅 소프트웨어로 규정하는 데이터 스토리지 체계이며, 일정 조직 내 여러 스토리지를 하나처럼 관리하고 운용하는 컴퓨터 이용 환경으로 스토리지 자원을 효율적으로 나누어 쓰는 방법이다.

데이터웨어하우스(Data Warehouse)

기간 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서 의사결정지원시스템을 지원하는 주제적, 통합적, 시간적 데이터의 집합체이다.
통합된 데이터에 대한 OLAP(On-Line Analytical Processing) 연산을 효율적으로 지원할 수 있다.

데이터 마트(Data Mart)

데이터웨어하우스와 사용자 사이의 중간층에 위치하며 데이터웨어하우스보다 규모나 비용 측면에서 축소된 개념이다.
빅데이터(Big Data)

많은 양의 정형 또는 비정형 데이터들로부터 가치를 추출하고 결과를 분석하는 기술이다.
빅데이터의 특성은 Volume(규모), Velocity(속도), Variety(다양성)이다.
구글 및 페이스북, 아마존의 경우 이용자의 성향과 검색 패턴, 구매패턴을 분석해 맞춤형 광고를 제공하는 등 빅데이터의 활용을 증대시키고 있다.
데이터마이닝(Data Mining)

대량의 데이터를 분석하여 데이터 속에 있는 변수 사이의 상호관계를 규명하여 일정한 패턴을 찾아내는 기법이다.
데이터웨어하우징에서 수집되고 분석된 자료를 사용자에게 제공하기 위해 분류 및 가공되는 요소 기술이다.
디지털 아카이빙(Digital Archiving)

디지털 정보 자원을 장기적으로 보존하기 위한 작업이다.
아날로그 콘텐츠는 디지털로 변환해 압축해서 저장하고, 디지털 콘텐츠도 체계적으로 분류하고 메타 데이터를 만들어 DB 화하는 작업이다.
하둡(Hadoop)

오픈소스를 기반으로 한 분산 컴퓨팅 플랫폼으로 일반 PC 급 컴퓨터들로 가상화된 대형 스토리지를 형성하고, 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 빅데이터 분산 처리를 돕는 자바 소프트웨어 오픈소스 프레임워크이다.
다양한 소스를 통해 생성된 빅데이터를 효율적으로 저장하고 처리한다.
하둡의 필수 핵심 구성 요소는 맵리듀스와 하둡 분산 파일 시스템이다.
Sqoop : 하둡과 관계형 데이터베이스 간에 데이터를 전송할 수 있도록 설계된 도구이다.
맵리듀스(MapReduce)

Hadoop 의 핵심 구성 요소로서 대용량 데이터를 분산 처리하기 위한 목적으로 개발된 프로그래밍 모델이다.
Google 에 의해 고안된 기술로써 대표적인 대용량 데이터 처리를 위한 병렬 처리 기법을 제공한다.
임의의 순서로 정렬된 데이터를 분산 처리하고 이를 다시 합치는 과정을 거친다.
75. 소프트웨어 개발 보안

소프트웨어 개발 보안의 개념

소프트웨어 개발 보안은 소프트웨어 개발 과정에서 발생될 수 있는 보안 취약점이나 보안 약점들을 최소화하여 사이버 보안 위협에 대응할 수 있는 안전한 소프트웨어를 개발하기 위한 보안 활동이다.
소프트웨어 개발 생명주기(SDLC : Software Development Lift Cycle)의 단계별로 요구되는 보안 활동을 수행하여 안전한 소프트웨어를 개발한다.
소프트웨어 보안 취약점 발생 원인

보안 요구사항이 정의되지 않거나 논리적인 오류를 가지는 설계를 수행하였다.
기술 취약점을 가지는 코딩 규칙을 적용하거나 소프트웨어 배치가 적절하지 않았다.
발견된 취약점에 대해 적절한 관리 또는 패치를 하지 않았다.
소프트웨어 개발 보안 체계

소프트웨어 개발 보안 관련 활동 주체는 행정안전부, 발주기관(행정기관 등),
한국인터넷진흥원, 사업자, 감리법인(진단원) 등으로 구분할 수 있다.
개발 보안 주체별로 잘 정의된 개발 보안 활동과 주체 간의 유기적인 협력이 필요하다.
활동 주체별 개발 보안 활동
행정안전부
지침 고시, 가이드 배포, 진단원 자격 여부 등
발주기관에 개발 보안 지침 - 가이드 제공
한국 인터넷 진흥원에 정책 지원
발주기관
개발 보안 지침 준수, 사업자에 개발 요청, 감리 법인에 확인 요청
한국인터넷징흥원
정책-기술 지원, 가이드 개발, 교육과정 문의, 발주기관에 기술 지원
사업자 및 감리법인에 교육 제공 가이드 안내
사업자
교육 이수, 시큐어 코딩 적용, 보안 약점 제거 등
감리법인
보안 약점 진단, 사업자에 개발 보안 적용 확인
프로젝트 참여 역할별 보안 활동

프로젝트 관리자(Projeect Manager)
팀 구성원에게 응용 프로그램의 보안 전략을 알려야 한다.
보안 위험과 비즈니스에 응용 프로그램 보안의 영향을 이해시킨다.
조직의 상태를 모니터링한다.

요구사항 분석가(Requirement Specifier)
아키텍트가 고려해야 할 여러 가지 보안 관련 비즈니스 요구사항들을 설명할 수 있어야 한다.
프로젝트팀이 고려해야 할 구조를 정의한 뒤, 해당 구조에 존재하는 자원에 대한 보안 요구사항이 무엇인지 결정한다.
보안 수준을 추상화할 때 다른 프로젝트에 적용되었던 보안 요구사항을 재사용하여 시간을 절약할 수 있어야 한다.
아키텍트(Architect)
명백한 보안 오류를 도입하지 않도록 충분히 보안 기술의 문제를 이해할 수 있어야 한다.
시스템에 사용되는 모든 리소스를 가능한 자세하게 정의한다.
시스템에서 각각 리소스의 역할에 적절한 보안 요구사항이 적용되도록 한다.
각 리소스가 시스템 라이프 사이클을 통한 서로 간의 상호작용을 이해할 수 있게 해야 한다.
설계자(Designer)
특정 기술이 설계 보안 항목을 만족하는지 확인하고 제대로 그 기술이 사용될 수 있는 방법을 파악해야 한다.
일반적으로 결과를 평가하고 최선의 문제 해결 방법을 결정해야 한다.
설계자는 모든 기존 개발 역할의 보안 관련 작업을 수행할 수 있어야 한다.
구현개발자(Implementer)
고도로 구조화된 개발 환경에서 프로그램을 구현하기 위해 안전한 코딩 표준을 준수하여 개발하여야 한다.
제 3 자가 소프트웨어 안전 여부를 쉽게 판단할 수 있도록 문서화해야 한다.
테스트 분석가(Test Analyst)
요구사항 구현 결과를 반복적으로 테스트해야 한다.
테스트 그룹은 반드시 보안 전문가일 필요는 없으며, 테스트가 가능할 정도의 위험에 대한 학습이나 툴 사용 방법을 숙지하고 있으면 된다.
보안감사자(Security Auditor)
프로젝트의 현재 상태를 검사하고 현재 상태의 보안을 보장한다.
설계단계에서는 일반적으로 취약성으로 이어질 수 있는 사항이 있는지 점검한다.
Secure OS

컴퓨터 운영체제의 커널에 보안 기능을 추가한 것으로 운영체제의 보안상 결함으로 인해 발생 가능한 각종 해킹으로부터 시스템을 보호하기 위하여 사용된다.
네트워크 보안 제품의 무력화 시 최후 시스템 보호 역할을 수행하며 조직의 보안 정책 및 역할에 최적화되어 보안 정책 관리를 지원한다.
Secure OS 의 목적

안정성 : 중단 없는 안정적 서비스 지원
보안성 : 핵심 서버 침입 차단 및 통합 보안 관리
신뢰성 : 중요 정보의 안전한 보호 기반 신뢰성 확보
버퍼 오버플로우(Buffer Overflow)

운영 체제가 메모리를 조작하는 동안 잘못된 동작을 하는 프로그램의 취약점으로 보통 데이터 저장 과정에서 데이터를 저장할 메모리 위치의 유효성을 검사하지 않을 때 발생한다.
버퍼 오버플로우 대응 방안

운영체제의 주기적 최신 패치와 입력값 검증이 가능한 안전 함수를 사용한다.
스택 실드 : 함수 시작 시 복귀 주소를 'Global RET'이라는 특수 스택에 저장해 두고 함수 종료 시 스택의 RET 값을 비교해 다를 경우 오버플로우 상태로 간주하여 프로그램 실행을 중단한다.
ASLR : 메모리 공격 방어를 위해 주소 공간 배치를 난수화, 실행 시마다 메모리 주소를 변경하여 오버플로우를 통한 특정 주소의 호출을 차단한다.
스택 가드(Stack Guard) : 메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값(카나리)을 저장해 두었다가 그 값이 변경되었을 경우 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술이다.
시스로그(Syslog)

Linux 에서 다양한 이벤트를 로그 파일에 기록하는 것을 의미한다.
다른 의미로는 Syslog Server 라고 불리는 이벤트 메시지(로그) 수집기 쪽으로 IP 네트워크를 통해서 장치(Machine)의 이벤트 메시지들을 전송할 수 있게 해주는 프로토콜이다.
76. 소프트웨어 개발 보안 구축 및 방법론의 종류
Secure SDLC(Software Developent Life Cycle)
소프트웨워 개발 보안 방법론의 개념

기존의 소프트웨어 개발 방법론이 적용된 프로젝트에서 안전한 소프트웨어 개발에 요구되는 보안 활동들을 적용하는 개발 방법이다.
SDLC(소프트웨어 개발 생명주기)에 걸쳐 추가되는 보안 활동은 다음과 같다.
요구사항 분석
요구사항 중 보안 항목 식별, 요구사항 명세서
설계
위협원 도출을 위한 위협 모델링
보안 설계 검토 및 보안 설계서 작성, 보안 통제 수립

구현
표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수해 개발
소스 코드 보안 약점 진단 및 개선
테스트
모의 침투 테스트 또는 동전 분석을 통한 보안 취약점 진단 및 개선
유지보수
지속적인 개선, 보안 패치
소프트웨어 개발 보안 방법론의 종류
MS-SDL(Microsoft-Secure Development LifeCycle)

마이크로소프트사에서 보안 수준이 높은 안전한 소프트웨어를 개발하기 위해 수행한 프로세스 개선 작업으로 자체 수립한 SDL 방법론을 적용하였다.
교육
소프트웨어 개발 보안 교육
안전 설계, 위협 모델링, 시큐어 코딩, 보안 테스팅, 프라이버시 관련 보안 교육
계획/분석
소프트웨어의 질과 버그 경계 정의, 보안과 프라이버시 위험 분석
설계
공격 영역 분석, 위협 모델링
구현
도구 명세, 금지된 함수 사용 제한, 정적 분석
시험/검증
위협모델 검토 및 수정
배포/운영
사고 대응 계획, 최종 보안 검토, 기록 보관
대응
사고 대응 수행
Seven Touchpoints

소프트웨어 보안의 모범 사례를 SDLC 에 통합한 개발 보안 방법론이다.
공통 위험 요소를 파악하고 이해하며, 보안을 설계하고 모든 소프트웨어 산출물에 대해 철저하고 객관적인 위험 분석 및 테스트를 거쳐 안전한 소프트웨어를 만들어내는 방법을 정의하고 있다.
SDLC 의 각 단계에서 7 개의 보안 강화 활동을 집중적으로 관리하도록 개발자에게 요구한다.
보안 강화활동\SDLC 단계 요구 사항 및 UseCases 구조 설계 테스트 계획 코드 테스트 및 테스트 결과 현장과의 피드백

악용 사례	•			
보안 요구사항	•			
위험 분석	•	•		•
위험 기반 보안 테스트				•
코드 검토			•	
침투 테스트				• •
보안 운영				•

CLASP(Comprehensive, Lightweight Application Security Process)

SDLC 초기 단계에 보안 강화를 목적으로 하는 정형화된 개발 보안 프로세스이다
활동 중심의 프로세스와 역할 기반의 프로세스로 구성된 집합체이다.
안전한 소프트웨어를 개발하기 위해 개념 관점, 역할 기반 관점, 활동 평가 관점, 활동 구현 관점, 취약성 관점 등 5 가지 관점에 따라 개발 보안 프로세스를 수행한다.
개념 관점
CLASP 구조와 CLASP 프로세스 컴포넌트 간의 종속성을 제공한다.
CLASP 프로세스 컴포넌트들의 상호 작용 방법과 취약성 관점을 통해서 역할 기반 관점에 적용하는 방법을 기술한다.
역할 기반 관점
24 개의 보안 관련 CLASP 활동들에 요구되는 각 역할을 창출하여 활동 평가 관점, 활동 구현 관점, 취약성 관점에서 사용한다.
활동 평가 관점
활동 평가관점, 활동 구현 관점, 취약성 관점에서의 적합성과 관련하여 보안 관련 CLASP 활동들에 대한 타당성을 평가한다.
활동 구현 관점
활동 평가 관점에서 선택한 24 개의 보안 관련 CLASP 활동들을 수행한다.
취약성 관점
문제 타입에 대한 솔루션을 활동 평가 관점, 활동 구현 관점으로 통합한다.
정보 보안의 3 대 요소

기밀성(Confidentiality)
인가된 사용자만 정보 자산에 접근할 수 있다.
일반적인 보안의 의미와 가장 가깝다.
방화벽, 암호 패스워드 등이 대표적인 예이다.
신분 위장(Masquerading) 등과 같은 공격에 의해 위협받을 수 있다.
무결성(Integrity)
시스템 내의 정보는 오직 인가된 사용자가 인가된 방법으로만 수정할 수 있다.

변경, 가장, 재전송 등과 같은 공격에 의해 위협받을 수 있다.
가용성(Availability)
사용자가 필요할 때 데이터에 접근할 수 있는 능력을 말한다.
인가된 사용자가 조직의 정보 자산에 적시에 접근하여 업무를 수행할 수 있도록 유지하는 것을 목표로 한다.
가용성을 유지하기 위해 데이터 백업, 위협 요소 제거 등의 기술을 사용할 수 있다.
서비스 거부(Denial of Service) 등과 같은 공격에 의해 위협받을 수 있다.
OWASP(The Open Web Application Security Project)
오픈소스 웹 애플리케이션 보안 프로젝트로서 주로 웹을 통한 정보 유출, 악성 파일 및 스크립트, 보안 취약점 등을 연구하는 곳이다.
연구 결과에 따라 취약점 발생 빈도가 높은 10 가지 취약점을 공개한다.
77. 시큐어 코딩 가이드 1
입력 데이터 검증 및 표현
입력 데이터 검증 및 표현의 개념

프로그램에 입력되는 데이터로 인해 여러 가지 보안 약점이 발생할 수 있다.
이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

운영체제 명령어 삽입, 위험한 형식 파일 업로드, 신뢰되지 않는 URL 주소로 자동 접속 연결된다.

SQL Injection

검증되지 않은 외부 입력값이 SQL 쿼리문에 삽입되어 공격할 수 있는 보안 약점이다.
DB 에 컴파일된 SQL 쿼리문을 전달함으로써 방지할 수 있다.
SQL Injection 취약점이 발생하는 곳을 주로 웹 애플리케이션과 데이터베이스가 연동되는 부분이다.
로그인과 같이 웹에서 사용자의 입력값을 받아 데이터베이스 SQL 문으로 데이터를 요청하는 경우 SQL Injection 을 수행할 수 있다.
경로 조작 및 자원 삽입

검증되지 않은 외부 입력값이 시스템 자원 접근 경로를 조작하거나 시스템 자원에 삽입되어 공격할 수 있는 보안 약점이다.
크로스 사이트 스크립트(XSS, Cross Site Scripting)

게시판의 글에 원본과 함께 악성 코드를 삽입하여 글을 읽으면 악성코드가 실행되도록 하여 클라이언트의 정보를 유출하는 공격 방법이다.

웹페이지에 악의적인 스크립트를 포함시켜 사용자 측에서 실행되게 유도함으로써, 정보 유출 등의 공격을 유발할 수 있는 취약점이다.
외부 입력값에 스크립트가 삽입되지 못하도록 문자열 치환 함수를 사용하거나 JSTL 이나 크로스사이트 스크립트 방지 라이브러리를 사용함으로써 방지할 수 있다.
XQuery 삽입

XQuery 를 사용하여 XML 데이터에 대한 동적 쿼리 생성 시 검증되지 않은 외부 입력값이 쿼리문 구조 변경에 사용될 수 있는 보안 약점이다.
XQuery 에 사용되는 외부 입력값에 대하여 특수문자 및 쿼리 예약어 필터링을 통해 방지할 수 있다.
XPath 삽입

검증되지 않은 외부 입력값으로 XPath 쿼리문을 생성하여 쿼리문의 의미나 구조가 변경될 수 있는 보안 약점이다.
LDAP 삽입

외부 입력값이 올바르게 처리되지 못하여 LDAP(Lightweight Directory Access Protocol) 쿼리문의 구성 변경에 사용될 수 있는 보안 약점이다.
DN(Distinguished Name)과 필터에 사용되는 외부 입력값에 특수문자를 제거함으로써 방지할 수 있다.
보안 기능
보안 기능의 개념

인증, 접근 제어, 기밀성, 암호화, 권한 관리 등의 보안 기능을 부적절하게 구현하여 여러 가지 보안 약점이 발생할 수 있다.
이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

적절한 인증 없는 중요 기능 허용
적절한 인증 없이 중요 정보를 읽거나 변경할 때 발생하는 보안 약점이다.
인증 과정 없이 서버에 접근하지 못하도록 하고, 중요 정보는 재인증을 거치도록 함으로써 방지할 수 있다.
부적절한 인가
접근 가능한 실행 경로에 대한 접근 제어 검사를 완전하게 하지 않아 정보가 유출되는 보안 약점이다.
노출되는 실행 경로를 최소화하고 사용자의 권한에 따라 접근 제어 리스트(Access Control list)를 관리함으로써 방지할 수 있다.

중요한 자원에 대한 잘못된 권한 설정
보안 또는 설정 파일과 같이 중요한 자원에 대해 읽기나 쓰기 권한을 잘못 설정하여 발생하는 보안 약점이다.
중요한 자원은 관리자만 읽고 쓰기가 가능하게 하고 사용자의 권한을 검사함으로써 방지할 수 있다.
취약한 암호화 알고리즘 사용
취약하거나 위험한 암호화 알고리즘을 사용하여 패스워드가 유출되는 보안 약점이다.
잘 알려진 안전한 암호화 알고리즘을 사용함으로써 방지할 수 있다.
중요 정보 평문 저장
개인정보, 금융정보, 패스워드 등의 중요 정보를 암호화하지 않고 평문으로 저장하여 중요 정보가 노출되는 보안 약점이다.
중요 정보를 암호화하여 저장하고 중요 정보 접근 시 사용자의 권한을 검사함으로써 방지할 수 있다.
중요 정보 평문 전송
중요 정보를 암호화하지 않고 평문으로 전송하여 중요 정보가 노출되는 보안 약점이다.
중요 정보를 암호화하여 전송하거나 보안 채널을 사용함으로써 방지할 수 있다.
하드 코딩된 비밀번호
프로그램 코드 내에 데이터를 직접 입력하는 하드 코드된 패스워드를 포함시켜 사용하여 관리자의 정보가 노출되는 보안 약점이다.
패스워드는 암호화하여 별도의 파일에 저장하여 사용하고 디폴트 패스워드 대신 사용자 입력 패스워드를 사용함으로써 방지할 수 있다.
충분하지 않은 키 길이 사용
길이가 짧은 키로 암호화 및 복호화를 함으로써 짧은 기간 안에 키를 찾아낼 수 있는 보안 약점이다.
RSA 알고리즘은 2,048 비트 이상, 대칭 암호화 알고리즘은 128 비트 이상의 키를 사용함으로써 방지할 수 있다.
적절하지 않은 난수값 사용
적절하지 않은 난수값을 사용하여 난수가 예측 가능해질 수 있는 보안 약점이다.
난수 값을 결정하는 현재 시각 기반 등으로 시드값을 매번 변경함으로써 방지할 수 있다.
78. 시큐어 코딩 가이드 2
시간 및 상태
시간 및 상태의 개념

동시 수행을 지원하는 병렬 시스템이나 여러 개의 프로세스가 동작되는 멀티 프로세스 환경에서 시간 및 상태를 부적절하게 사용하여 여러 가지 보안 약점이 발생할 수 있다.
이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.

보안 약점의 종류

경쟁 조건 : 검사 시점과 사용 시점(TOCTOU)
자원을 검사하는 시점(TOC : Time Of Check)과 사용하는 시점(TOU : Time Of Use)이 달라서 발생하는 보안 약점이다.
여러 프로세스가 공유 자원 접근 시 동기화 구문으로 한 번에 하나의 프로세스만 접근하게 함으로써 방지할 수 있다.
종료되지 않는 반복문 또는 재귀함수
종료 조건이 없는 반복문이나 재귀 함수를 사용하여 무한 반복하며 자원 고갈이 발생하는 보안 약점이다.
재귀 호출 횟수르 제한함으로써 방지할 수 있다.
에러 처리
에러 처리의 개념

발생한 에러를 처리하지 않거나 완전하게 처리하지 않아 에러 정보에 중요 정보가 포함되어 여러 가지 보안 약점이 발생할 수 있다. 이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

에러 메시지를 통한 정보 노출 : 에러 메시지에 실행 환경이나 사용자 관련 등 민감한 정보가 포함되어 외부에 노출되는 보안 약점이다.
에러 상황 대응 부재 : 에러가 발생할 수 있는 에러 상황에 대해 예외 처리를 하지 않아 프로그램이 동작하지 않거나 제대로 동작하지 않는 보안 약점이다.
부적절한 예외 처리 : 프로그램 수행 중에 함수의 결과값에 대해 적절하게 처리하지 않거나 예외 상황에 대해 조건을 적절하게 검사하지 않아 발생하는 보안 약점이다.
코드 오류
코드 오류의 개념

개발자가 흔히 실수하는 프로그램 오류들로 인해 여러 가지 보안 약점이 발생할 수 있다. 이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

Null Pointer(널 포인터) 역참조 : 일반적으로 객체가 Null 이 될 수 없다는 가정을 위반하여 공격자가 의도적으로 Null Pointer 역참조를 발생시켜 공격에 사용하는 보안 약점이다.
부적절한 자원 해제 : 오픈 파일 디스크립터, 힙 메모리, 소켓 등의 유한한 자원을 할당받아 사용한 후 프로그램 에러로 반환하지 않아 발생하는 보안 약점이다.

해제된 자원 사용 : 해제된 자원을 참조하여 의도하지 않은 값이나 코드를 실행하게 됨으로써 의도하지 않은 결과가 발생하는 보안 약점이다.
초기화되지 않은 변수 사용 : 초기화되지 않은 변수를 사용하면 임의의 값이 사용되어 의도하지 않은 결과가 발생하는 보안 약점이다.
캡슐화
캡슐화의 개념

중요한 데이터나 기능성을 잘못 캡슐화하거나 잘못 사용하면 여러 가지 보안 약점이 발생할 수 있다. 이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

잘못된 세션에 의한 데이터 정보 노출
다중 스레드 환경에서 정보를 저장하는 멤버 변수가 포함되어 서로 다른 세션에서 데이터를 공유하여 발생하는 보안 약점이다.
싱글톤(Singleton) 패턴 사용 시 변수 범위를 제한하여 방지할 수 있다.
제거되지 않고 남은 디버그 코드
개발 완료 후에 디버그 코드가 제거되지 않은 채로 배포되어 발생하는 보안 약점이다.
소프트웨어가 배포되기 전에 디버그 코드를 삭제해 방지할 수 있다.
시스템 데이터 정보 노출
시스템, 관리자, DB 정보 등의 시스템 데이터 정보가 공개되어 발생하는 보안 약점이다.
예외 상황 발생 시 시스템 메시지 등의 시스템 데이터 정보가 화면에 출력되지 않게 함으로써 방지할 수 있다.
public 메서드로부터 반환된 private 배열
private 선언된 배열을 public 선언된 메서드를 통해 반환하여 그 배열의 레퍼런스가 외부에 공개되어 발생하는 보안 약점이다.
private 선언된 배열을 public 선언된 메서드를 통해 반환하지 않게 함으로써 방지할 수 있다.
private 배열에 public 데이터 할당
public 선언된 메서드의 인자가 private 선언된 배열에 저장되어 그 배열을 외부에서 접근할 수 있게 되는 보안 약점이다.
public 선언된 메서드의 인자를 private 선언된 배열에 저장되지 않도록 함으로써 방지할 수 있다.
API 오용
API 오용의 개념

서비스에서 제공되는 사용법에 반하는 방법으로 API 를 사용하거나 보안에 취약한 API 를 사용하여 여러 가지 보안 약점이 발생할 수 있다. 이러한 보안 약점을 방지하기 위한 보안 점검 항목들이다.
보안 약점의 종류

DNS lookup 에 의존한 보안 결정
도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 하면 공격자가 DNS 엔트리를 속여 동일 도메인에 속한 서버인 것처럼 위장하는 보안 약점이다.
보안 결정 시 도메인명을 이용한 DNS lookup 에 의존하지 않도록 함으로써 방지할 수 있다.
취약한 API 사용
보안 문제로 금지된 함수 또는 오용될 가능성이 있는 API 등의 취약한 API 를 사용하여 발생하는 보안 약점이다.
보안 문제로 금지된 함수는 안전한 대체 함수를 사용함으로써 방지할 수 있다.
79. 암호화 알고리즘
암호 알고리즘
암호 알고리즘(Cryptographic Algorithm)의 개념

평문을 암호문으로 바꾸고, 암호문을 다시 평문으로 바꿀 때 사용되는 알고리즘을 의미한다.
평문을 암호문으로 바꾸는 과정을 암호화(Encryption)라고 하고, 암호문을 다시 평문으로 바꾸는 과정을 복호화(Decryption)라고 한다.
암호화 및 복호화 과정에 암호키(Cryptographic key)가 필요하다.
암호 방식의 분류

암호화 방식
단방향
해시
양방향
비밀키
스트림 방식
블록 방식
공개 키
공개키(Public Key, 비대칭키) 암호화 기법

암호키와 해독키가 서로 다른 기법으로 키 개수는 2N 개가 필요하다.
비대칭키 암호화 기법 또는 공중키 암호화 기법이라고도 한다.

키 분배가 비밀키 암호화 기법보다 쉽고, 암호화/복호화 속도가 느리며 알고리즘이 복잡하다.

RSA, ElGama 기법 등이 있다.

RSA(Rivest Shamir Adieman)

소인수 분해의 어려움에 기초를 둔 알고리즘

1978 년 MIT 에 의해 제안됨

전자문서에 대한 인증 및 부인 방지에 활용된다.

ElGama

이산대수 문제의 어려움에 기초를 둔 알고리즘

동일한 메시지라도 암호화가 이루어질 때마다 암호문이 변경되고 암호문의 길이가 2 개로 늘어나는 특징이 있다.

비밀키(Private Key, 대칭키) 암호화 기법

동일한 키로 암호화하고 복호화하는 기법으로 키 개수는 N(N-1)/2 개가 필요하다.

대칭키 암호화 기법 또는 개인키 암호화 기법이라고도 한다.

암호화/복호화 속도가 빠르고 알고리즘이 단순하다.

키 분배가 공개키 암호화 기법보다 어렵다.

스트림 방식과 블록 방식으로 분류된다.

스트림 방식

평문의 길이와 동일한 스트림(Stream)을 생성하여 비트 단위로 암호화하는 대칭키 암호화 방식이다. 암호화할때 XOR 연산을 수행한다.

종류 : RC4, A5/1, LSFR, SEAL, WEP, OFB

블록 방식

평문을 블록 단위로 암호화하는 대칭키 암호화 방식이다.

종류

DES(Data Encryption Standard)

1970 년대 초 IBM 이 개발한 알고리즘이다.

16 라운드 Feistel 구조를 가진다.

평문을 64 비트로 블록화를 하고, 실제 키의 길이는 56 비트를 이용한다.

전사 공격(Brute-Force Attack)에 취약하다.

AES(Advanced Encryption standard)

DES 를 대신하여 새로운 표준이 되었다.

블록 크기는 128 비트이고, 키 길이는 128/192/256 비트이다.

SPN(Substitution-Permutation Network) 구조이다.

ARIA

국내 기술로 개발된 암호 알고리즘이다.

경량 환경 및 하드웨어 구현에서의 효율성 향상을 위해 개발되었다.

우리나라 국가 표준으로 지정되었다.
블록 크기와 키 길이가 AES 와 동일하다.
SEED
국내 기술로 개발된 128 비트 블록 암호 알고리즘이다.
Feistel 구조이다.
2005 년 국제 표준으로 제정되었다.
IDEA
DES 를 대체하기 위해서 스위스에서 개발한 알고리즘이다.
상이한 대수 그룹으로부터의 세 가지 연산을 혼합하는 방식이다.
해시(HASH) 암호화 방식
임의의 길이의 메시지를 입력으로 하여 고정된 길이의 출력값을 변환하는 시법이다.
주어진 원문에서 고정된 길이의 의사난수를 생성하며, 생성된 값을 해시값이라고 한다.
해시 함수라고도 한다.
디지털 서명에 이용되어 데이터 무결성을 제공한다.
블록체인에서 체인 형태로 사용되어 데이터의 신뢰성을 보장한다.
SHA, SHA1, SHA256, MD5, RMD160, HAS-160, HAVAL 기법 등이 있다.
SHA(Secure Hash Algorithm)
1993 년에 미국 NIST 에 의해 개발되었고 가장 많이 사용되고 있는 방식이다.
SHA-1 은 DSA 에서 사용하게 되어 있으며 많은 인터넷 응용에서 Default 해시 알고리즘으로 사용된다.
SHA-256, SHA-384, SHA-512 는 AES 의 키 길이인 128, 192, 256bit 에 대응하도록 출력 길이를 늘린 해시 알고리즘이다.
SALT
시스템에 저장되는 패스워드들은 Hash 또는 암호화 알고리즘의 결과값으로 저장된다. 이때 암호 공격을 막기 위해 똑같은 패스워드들이 다른 암호 값으로 저장되도록 추가되는 값을 의미한다.
80. 서비스 공격 유형
Dos(Denial of Service, 서비스 거부)

시스템의 자원을 부족하게 하여 원래 의도된 용도로 사용하지 못하게 하는 공격 방법이다.

정보보호의 3 대 목표 중 가용성(Availability)을 위협하는 행위로서 공격자가 임의로 자신의 IP 주소를 속여서 다량으로 서버에 보낸다.

헤더가 조작된 일련의 IP 패킷 조각들을 전송한다.

라우터, 웹, 전자 우편, DNS 서버 등 모든 네트워크 장비를 대상으로 이루어질 수 있다.

공격 종류

스머프(Smurf)
공격 대상의 IP 주소를 근원지로 대량의 ICMP 응답 패킷을 전송하여, 서비스 거부를 유발시키는 공격 방법이다.
IP 또는 ICMP 의 특성을 악용하여 특정 사이트에 집중적으로 데이터를 보내 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법이다.
SYN 플러딩(SYN Flooding)
TCP 연결 설정 과정의 취약점을 악용한 서비스 거부 공격이다.
TCP 3-Way Handshaking 과정에서 Half Open 연결 시도가 가능하다는 취약성을 이용한 공격 방법이다.
UDP 플러딩(UDP Flooding)
대량의 UDP 패킷을 만들어 보내 정상적인 서비스를 하지 못하도록 하는 공격 방법이다.
ICMP Unreachable : 공격 과정에서 지정된 UDP 포트가 나타내는 서비스가 존재하지 않을 때 발생되는 패킷이다.
Ping 플러딩(Ping Flooding)
네트워크의 정상 작동 여부를 확인하기 위해 사용하는 Ping 테스트를 공격자가 공격 대상 컴퓨터를 확인하기 위한 방법으로 사용하는 공격 방법이다.
특정 사이트에 매우 많은 ICMP Echo 를 보내면, 이에 대한 응답을 하기 위해 시스템 자원을 모두 사용해버려 시스템이 정상적으로 동작하지 못하도록 하는 공격 방법이다.
Ping of Death
비정상적인 ICMP 패킷을 전송하여, 시스템의 성능을 저하시키는 공격 방법이다.
티어드랍(TearDrop)
패킷 재조합의 문제를 악용하여 오프셋이나 순서가 조작된 일련의 패킷 조각들을 보냄으로써 지원을 고갈시키는 공격 방법이다.
랜드(LAND, Local Area Network Denial) Attack
공격자가 패킷의 출발지 IP 주소나 포트(Port)를 임의로 변경하여 출발지와 목적지 주소(또는 포트)를 동일하게 함으로써, 공격 대상 컴퓨터의 실행 속도가 느려지거나 동작이 마비되어 서비스 거부 상태에 빠지도록 하는 공격 방법이다.
DDoS(Distributed Denial of Service, 분산 서비스 거부)

여러 대의 공격자를 분산 배치하여 동시에 서비스 거부 공격함으로써 공격 대상이 되는 시스템이 정상적인 서비스를 할 수 없도록 방해하는 공격 방법이다.
공격용 도구 : Trinoo, TFN(Tribe Flood Network), TFN2K, Stacheldraht 등이 있다.
피싱(Phishing)

소셜 네트워크에서 진짜 웹 사이트와 거의 동일하게 꾸며진 가짜 웹 사이트를 통해 개인정보를 탈취하는 수법이다.
금융기관 등의 웹 사이트에서 보내온 메일로 위장하여 개인의 인증번호나 신용카드번호, 계좌정보 등을 빼내 이를 불법적으로 이용한다.
이블 트윈 공격(evil twin Attack)

피싱 사기의 무선 버전이다. 공격자는 합법적인 제공자처럼 행세하며 노트북이나 휴대전화로 핫스팟에 연결한 무선 사용자들의 정보를 탈취한다.
파밍(Pharming)

도메인을 탈취하거나 악성코드를 통해 DNS 의 이름을 속여 사용자가 진짜 웹 사이트로 오인하게 만들어 개인정보를 탈취하는 수법이다.
랜섬웨어(Ransomware)

개인과 기업, 국가적으로 큰 위협이 되고 있는 주요 사이버 범죄 중 하나로 Snake, Darkside 등 시스템을 잠그거나 데이터를 암호화해 사용할 수 없도록 하고 이를 인질로 금전을 요구하는데 사용되는 악성 프로그램이다.
키 로거(Key Logger)

컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 공격 방법이다.
무작위 대입 공격(Brute-Force Attack)

패스워드(Password)에 사용될 수 있는 문자열의 범위를 정하고, 그 범위 내에서 생성 가능한 패스워드를 활용하는 공격 방법이다.
APT(Advanced Persistent Threat, 지능적 지속 위협)

개인 단체, 정치 단체, 국가, 산업체 등 목표 조직을 타깃으로 하여 다양한 보안 위협을 만들어 침해에 성공해 정보를 유출하거나 장기간의 접속 권한을 획득하기 위해 또는 장기간의 접근을 위해 지속적으로 수행되는 공격법이다.
제로데이(Zero-day) 공격

조사된 정보를 바탕으로 정보 시스템, 웹 애플리케이션 등의 알려지지 않은 취약점 및 보안 시스템에서 탐지되지 않은 악성코드 등을 감염시키는 것이다.
백도어(Back Door)

프로그램이나 손상된 시스템에 허가되지 않는 접근을 할 수 있도록 정상적인 보안 절차를 우회하는 악성 소프트웨어이다. 트랩 도어(Trap Door)라고도 한다.

백도어 공격 도구로는 NetBus, Back Orifice, RootKit 등이 있다.
백도어 탐지 방법에는 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등이 있다.
tripwire : 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구이다.
TCP 세션 하이재킹

서버와 클라이언트 통신 시에 TCP 의 3-Way Handshake 단계에서 발생하는 취약점을 이용한 공격기법으로 서버와 클라이언트가 TCP 를 이용하여 통신하고 있을 때, RST 패킷을 전송하여 일시적으로 TCP 세션을 끊고 시퀀스 번호를 새로 생성하여 세션을 탈취하고 인증을 회피하는 공격 기법이다.
비동기화 상태와 동기화 상태 2 가지가 존재한다.
세션 하이재킹 탐지 기법 : 비동기화 상태 감지, ACK STORM 탐지, 패킷의 유실 및 재전송 증가 탐지, 예상치 못한 접속의 리셋 탐지
SSH 같은 세션 인증 수준이 높은 프로토콜 사용을 통해 방어하도록 한다.
81. 서버 인증 및 서버 접근 통제
서버 인증
사용자 인증 기법

지식 기반 인증(Knowledge-based Authentication)
사용자가 기억하고 있는 지식을 기초로 접근 제어를 수행하는 사용자 인증 기법이다.
아이디, 패스워드, PIN(Personal Identification Number) 번호 등
소유 기반 인증(Authentication by what the entity has)
사용자가 소유하고 있는 인증 토큰을 기반으로 하는 사용자 인증 기법이다.
지식 기반 인증 기법보다 보안성이 높다.
건물 출입 시 사용되는 스마트 카드, 인터넷 뱅킹 시 사용되는 OTP(One Time Password) 단말, 공인인증서 등
생체 기반 인증
사람의 정적인 신체적 특성 또는 동적인 행위적 특성을 이용하는 사용자 인증 기법이다.
지문 인식, 홍재 인식, 정맥 인식, 음성 인식 등이 해당된다.
일성, 영속성, 정량성, 보편성 등
서버 접근 통제
접근 통제(Access Control)의 개념

시스템의 자원 이용에 대한 불법적인 접근을 방지하는 과정이다.
크래커(Cracker)의 침입으로부터 보호한다.

종류 : 강제적 접근 통제, 임의적 접근 통제, 역할 기반 접근 통제 - Secure OS 의 보안 기능과 동일하다.
접근 통제 요소

식별 : 인증 서비스에 스스로를 확인시키기 위하여 정보를 공급하는 주체의 활동이다.
인증 : 주체의 신원을 검증하기 위한 사용자의 증명의 두 번째 부분이다.
인가 : 인증을 통해 식별된 주체의 실제 접근 가능 여부와 주체가 수행 가능한 일을 결정하는 과정이다.
대표적 접근 통제 모델

벨라파듈라 모델(BLP : Bell-LaPadula Confidentiality Model) : 군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용하며, 자신의 권한보다 낮은 보안 레벨 권한을 가진 경우에는 높은 보안 레벨의 문서를 읽을 수 없고 자신의 권한보다 낮은 수준의 문서만을 읽을 수 있다.
SSO(Single Sign-On) : 스템이 몇 대가 되어도 하나의 시스템에서 인증에 성공하면 다른 시스템에 대한 접근 권한도 얻는 시스템이다.
Biba Integrity Model : 무결성을 위한 최초의 상업적 모델이다.(BLP 를 보완, MAC).
무결성 목표 중 비인가자에 의한 데이터 변형 방지만 취급한다.(변조 방지를 목적으로 함)
CWM(Clark-Wilson Integrity Model) : 무결성 중심의 상업적 모델로 사용자가 직접 객체에 접근할 수 없고 프로그램을 통해서만 객체에 접근할 수 있게 하는 보안 모델이다.
무결성의 3 가지 목표를 모델을 통해서 각각 제시한다.
강제적 접근 통제(MAC: Mandatory Access Control)
중앙에서 정보를 수집하고 분류하여 보안 레벨을 결정하고 정책적으로 접근 제어를 수행하는 방식으로 다단계 보안 모델이라고도 한다.
어떤 주체가 특정 개체에 접근하려 할 때 양쪽의 보안 레이블(Security Label)에 기초하여 높은 보안 수준을 요구하는 정보(객체)가 낮은 보안 수준의 주체에게 노출되지 않도록 하는 접근 제어 방법이다.
대표적 접근통제 모델로 BLP(Bell-Lapadula), Biba, Clark-Wilson, 만리장성 모델 등이 있다.
임의적 접근 통제(DAC: Discretionary Access Control)
정보의 소유자가 보안 레벨을 결정하고 이에 대한 정보의 접근 제어를 설정하는 방식이다.
주체 또는 소속 그룹의 아이디(ID)에 근거하여 객체에 대한 접근 제한을 설정한다.
역할 기반 접근 통제(RBAC: Role Based Access Control)
사람이 아닌 직책에 대해 권한을 부여함으로써 효율적인 권한 관리가 가능하다.
접근 권한은 직무에 허용된 연산을 기준으로 허용함으로 조직의 기능 변화에 따른 관리적 업무의 효율성을 높일 수 있다.

MAC vs DAC vs RBAC

정책 MAC DAC RBAC
권한 부여 시스템 데이터 소유자 중앙 관리자
접근 결정 보안 등급(Label) 신분(Identity) 역할(Role)
장점 안정적 중앙 집중적 구현 용이 유연함 관리 용이
보안 아키텍처
보안 아키텍쳐(Security Architecture)의 개념

보안 설계 감독을 위한 원칙과 보안 시스템의 모든 양상에 대한 세부 사항을 의미한다.
보안 요구사항을 충족시키는 시스템 구성 방법에 대한 세부 사항이다.
정보 자산의 기밀성, 무결성 및 가용성을 높이기 위한 보안 영역의 구성 요소와 관계에 대한 세부 사항이다.
보안 프레임워크
보안 프레임워크(Security Framework)의 개념

정보의 기밀성, 무결성 및 가용성을 높이기 위한 정보 보안 시스템의 기본이 되는 뼈대이다.
보안 프레임워크는 기술적 보안, 관리적 보안, 물리적 보안 프레임워크로 나누어진다.
82. 보안 솔루션과 보안 아키텍쳐
보안 솔루션
IDS(Intrusion Detection System, 침입 탐지 시스템)

침입 공격에 대하여 탐지하는 것을 목표로 하는 보안 솔루션이다.

외부 침입에 대한 정보를 수집하고 분석하여 침입 활동을 탐지해 이에 대응하도록 보안 담당자에게 통보하는 기능을 수행하는 네트워크 보안 시스템이다.

예방적이고 사전에 조처를 하는 기술로서 HIDS 와 NIDS 로 구분한다.

HIDS(Host-based IDS, 호스트 기반 IDS)
컴퓨터 시스템의 내부를 감시하고 분석하여 침입을 탐지하는 시스템이다.
컴퓨터 시스템의 동작이나 상태를 모두 감시하거나 부분적으로 감시한다.
CPU, 메모리, 디스크 등 호스트 자원을 일정 부분 점유한다.
NIDS(Network-based IDS, 네트워크 기반 IDS)
네트워크상의 모든 패킷을 캡처링한 후 이를 분석하여 침입을 탐지한다.
네트워크 위치에 따라 설치할 수 있으며, 적절한 배치를 통하여 넓은 네트워크 감시가 가능하다.

HIDS 에 탐지 못 한 침입을 탐지할 수 있다.
침입 탐지 기법

오용 탐지(Misuse Detection)
이미 발견되어 알려진 공격 패턴과 일치하는지 검사하여 침입을 탐지한다.
속도가 빠르고 구현이 간단하다.
False Positive 가 낮은 반면 False Negative 가 높다.
이상 탐지(Anomaly Detection)
장기간 수집된 올바른 사용자 행동 패턴을 활용해 통계적으로 침입을 탐지한다.
알려지지 않은 공격을 탐지하는데 적합하다.
False Negative 가 높은 반면 False Positive 가 낮다.
호스트 기반과 네트워크 기반 침입 탐지 시스템에 모두 적용될 수 있다.
False Positive : 오탐 - 정상 패킷을 비정상으로 탐지했기 때문에 로그나 경고가 남아 번거로움이 생기나, 공격이 아니기 때문에 치명적인 결과를 낳지 않음

False Nagative : 미탐 - 비정상 패킷을 정상으로 판단하여 아무런 로그나 경고를 남기지 않아 호스트가 공격을 당할 위험이 발생함. 따라서 False Negative 를 줄이는 것이 가장 중요하다.

방화벽(Firewall)

내부-외부 네트워크 사이에 위치하여, 보안 정책을 만족하는 트래픽만 통과할 수 있다.

방화벽이 제공하는 기능에는 접근 제어, 인증, 감사 추적, 암호화 등이 있다.

불법 사용자의 침입 차단을 위한 정책과 이를 지원하는 하드웨어 및 소프트웨어를 제공한다.

방화벽 하드웨어 및 소프트웨어 자체의 결함에 의해 보안상 취약점을 가질 수 있다.

내부 네트워크에서 외부 네트워크로 나가는 패킷을 그대로 통과시키므로 내부 사용자에 의한 보안 침해는 방어하지 못한다.

방화벽의 유형

패킷 필터링(Packet Filtering)
패킷의 출발지 및 목적지 IP 주소, 서비스의 포트 번호 등을 이용한 접속 제어를 수행한다.
특정 IP, 프로토콜, 포트의 차단 및 허용을 할 수 있다.

바이러스에 감염된 파일 전송 시 분석이 불가능하다.
OSI 참조 모델의 제 3/4 계층에서 처리되므로 처리 속도가 빠르다.
상태 검사(Stateful Inspection)
패킷 필터링 기능을 사용하며 현재 연결 세션의 트래픽 상태와 미리 저장된 상태와의 비교를 통하여 접근을 제어한다.
응용 레벨 게이트웨이(Application Level Gateway)
OSI 참조 모델의 8 계층의 트래픽을 감시하여 안전한 데이터만을 네트워크 중간에서 릴레이한다.
응용 프로그램 수준의 트래픽을 기록하고 감시하기가 용이하며, 추가로 사용자 인증과 같은 부가 서비스를 지원할 수 있다.
응용 계층에서 동작하기 때문에 다른 방식의 방화벽에 비해 처리 속도가 가장 느리다.
회선 레벨 게이트웨이(Circuit Level Gateway)
종단 - 대 종단 TCP 연결을 허용하지 않고, 두 개의 TCP 연결을 설정한다.
시스템 관리자가 내부 사용자를 신뢰할 경우 일반적으로 사용한다.
내부 IP 주소를 숨길 수 있다.
베스천 호스트(Bastion Host) : 중세 성곽의 가장 중요한 수비 부분을 의미하는 단어로, 방화벽 시스템 관리자가 중점 관리하는 시스템을 말하며 액세스 제어 및 응용 시스템 게이트웨이로서 프록시 서버의 설치, 인증, 로그 등을 담당하느 호스트를 말한다.

방화벽 5 가지 구성 형태
스크리닝 라우터(Screening Router)

외부(인터넷)과 내부망의 가운데에서 패킷 필터링 규칙을 적용해서 방화벽의 역할을 수행하는 구조이다.
3 계층과 4 계층에서 IP 와 Port 에 대해 접근 제어를 하는 스크리닝 라우터는 매우 저렴하게 방화벽의 역할을 수행할 수 있으나 세부적인 규칙을 적용하기 어렵고, 만약에 접속이 폭주할 경우 부하가 걸려 효과적이지 못하다.
이중 홈 게이트웨이(Dual-Homed Gateway)

2 개의 네트워크 인터페이스를 가진 베스천호스트로서 하나의 NIC 는 내부 네트워크와 연결하고 다른 NIC 는 외부 네트워크와 연결한다. 방화벽은 하나의 네트워크에서 다른 네트워크로 IP 패킷을 라우팅하지 않기 때문에 프록시 기능을 부여한다.
내부에서 외부로 가려면 반드시 이중 홈 게이트웨이를 지나가야 하므로 좀 더 효율적으로 트래픽을 관리할 수 있다.
듀얼 홈드 호스트(Dual-Homed Host)

2 개의 네트워크 인터페이스를 가진 베스트천호스트로서 하나의 NIC 는 내부 네트워크와 연결하고 다른 NIC 는 외부 네트워크와 연결한다. 방화벽은 하나의 네트워크에서 다른 네트워크로 IP 패킷을 라우팅하지 않기 때문에 프록시 기능을 부여한다.
두 개의 인터페이스를 가지는 장비를 말하며, 하나의 인터페이스는 외부 네트워크와 연결되고 다른 인터페이스는 내부 네트워크로 연결되며, 라우팅 기능이 없는 방화벽을 설치하는 형태이다.
단일 홈 게이트웨이(Single-Homed Gateway)

스크리닝 라우터와 비슷한 구조를 가진다. 접근 제어, 프록시, 인증, 로깅 등 방화벽의 기본 기능을 수행하며, 보다 강력한 보안 정책을 실행할 수 있지만 방화벽이 손상되면 내부의 공격에 대해 무방비 상태가 된다.
2 계층에서 우회를 통한 공격이 가능하다.
스크린된 호스트 게이트웨이(Screened Host Gateway)

듀얼 홈드 게이트웨이와 스크리닝 라우터를 결합한 형태로 '숨겨진'이라는 의미로 방화벽이 숨겨져 있다. 패킷 필터링 호스트와 베스천 호스트로 구성되어 있다.
패킷 필터링 라우터는 외부 및 내부 네트워크(인터넷 쪽)에서 발생하는 패킷을 통과시킬 것인지를 검사하고 외부에서 내부로 유입되는 패킷(라우터와 내부 네트워크 사이)에 대해서는 베스천호스트로 검사된 패킷을 전달한다. 베스천호스트는 내부 및 외부 네트워크 시스템에 대한 인증을 담당한다.
3 계층과 4 계층에 대해서 접근 제어를 해주고 베스천호스트에서 7 계층에 대한 접근 제어를 하게 되지만 구축 비용은 위의 방식들보다 많이 비싼 편이다.
스크린된 서브넷 게이트웨이(Screend Subnet Gateway)

스크린드 호스트의 보안상 문제점을 보완한 모델로, 외부 네트워크와 내부 네트워크 사이에 하나 이상의 경계 네트워크를 두어 내부 네트워크를 외부 네트워크로 분리하기 위한 구조이다.
스크린된 서브넷 게이트웨이 방식은 외부와 내부의 가운데에 DMZ 를 위치시키며 방화벽도 DMZ 부분에 위치하고 주로 프록시가 설치된다.
설치 및 관리가 어렵고 속도가 느리며 고비용이다.
정보보호 대책
IPS(Intrusion Prevention System, 침입 방지 시스템)

사후에 조치를 취하는 기술로서 침입 공격에 대하여 방지하는 것을 목표로 하는 보안 솔루션이다.
IDS 와 방화벽의 장점을 결합한 네트워크 보안 시스템이다.
호스트의 IP 주소, 포트 번호, 사용자 인증에 기반을 두고 외부 침입을 차단한다.

허용되지 않는 사용자나 서비스에 대해 사용을 거부하여 내부 자원을 보호한다.

DMZ(DeMilitarized Zone, 비무장지대)

DMZ 는 보안 조치가 취해진 네트워크 영역이다.
메모리, 네트워크 연결, 접근 포인트 등과 같은 자원에 대한 접근을 제한하기 위한 구축된다.
내부 방화벽과 외부 방화벽 사이에 위치할 수 있다.
웹 서버, DNS 서버, 메일 서버 등이 위치할 수 있다.

IPSec(IP security)

통신 세션의 각 IP 패킷을 암호화하고 인증하는 안전한 인터넷 프로토콜(IP)이다.
ESP(Encapsulation Security Payload)는 발신지 인증, 데이터 무결성, 기밀성 모두를 보장한다.
운영 보드는 Tunnel 모드와 Transport 모드로 분류된다.
AH(Authentication Header)는 발신지 호스트를 인증하고, IP 패킷의 무결성을 보장한다.

DLP(Data Loss Prevention)

기업 데이터 유출을 방지하는 것을 목표로 하는 보안 솔루션이다.
사용자의 PC 에서 기업 내 기밀 데이터가 외부로 반출되는 것을 항시 감시하고 기록하며, 정책에 따라 유출을 차단시킨다.

ESM(Enterprise Security Management, 통합 보안 관리)

방화벽, 침입 탐지 시스템, 가상 사설망 등의 보안 솔루션을 하나로 모은 통합 보안 관리 시스템으로 서로 다른 보안 장비에서 발생한 각종 로그를 통합적으로 관리하여 통합 보안 관제 서비스를 제공한다.
전사적 차원의 보안 정책 통합 관리와 적용을 통해 정보 시스템 보안성을 향상시키고 안정성을 높인다.

VPN(Virtual Private Network, 가상 사설망)

이용자가 인터넷과 같은 공중망에 사설망을 구축하여 마치 전용망을 사용하는 효과를 가지는 보안 솔루션이다.
안전하지 않은 공용 네트워크를 이용하여 사설 네트워크를 구성하는 기술이다.
전용선을 이용한 사설 네트워크에 비해 저렴한 비용으로 안전한 망을 구성할 수 있다.
공용 네트워크로 전달되는 트래픽은 암호화 및 메시지 인증 코드 등을 사용하여 기밀성과 무결성을 제공한다.
인터넷과 같은 공공 네트워크를 통해서 기업의 재택근무자나 이동 중인 직원이 안전하게 회사 시스템에 접근할 수 있도록 해준다.