5과목-정보시스템 구축 관리

(Part 3. 소프트웨어 개발 보안 구축)

정보시스템 구축 관리 총 파트

정보시스템 구축 관리 5과목은 총 4Part로 이루어져 있다.

1장 소프트웨어 개발 방법론 활용(30.39%)

2장 IT프로젝트 정보 시스템 구축 관리 (36.46%)

3장 소프트웨어 개발 보안 구축(19.34%)

4장 시스템 보안 구축(13.81%)

소프트웨어 개발 보안 구축

소프트웨어 개발 보안 구축 Part는 7개의 섹션으로 구성되어 있다.

001 Secure SDLC

002 세션 통제

003 입력 데이터 검증 및 표현

004 보안 기능

005 코드오류

006 캡슐화

007 암호 알고리즘

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

1) 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 방법론이다. •활동 중심 역할 기반의 프로세스로 구성되어 있으며, 현재 운용 중인 시스템에 적용하기에 적합하다.
SDL	•마이크로소프트 사에서 안전한 소프트웨어 개발을 위해 기존의 SDLC를 개선한 방법론이다. •전통적인 나선형 모델을 기반으로 한다.
Seven Touchpoints	•소프트웨어 보안의 모범사례를 SDLC에 통합한 방법론이다. •설계 및 개발 과정의 모든 산출물에 대해 위험 분석 및 테스트를 수행한다. •SDLC의 각 단계에 관련된 7개의 보안 강화 활동을 수행한다.

소프트웨어 개발 생명주기(SDLC; Software Development Life Cycle)
소프트웨어 개발 생명주기는 소프트웨어 개발 방법론의 바탕이 되는 것으로 소프트웨어를 개발하기 위해 정의하고 운용, 유지보수 등의 전 과정을 각 단계별로 나눈 것이다.

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

2) 요구사항 분석 단계에서의 보안 활동

; 요구사항 분석 단계에서는 보안 항목에 해당하는 요구사항을 식별하는 작업을 수행한다.

- 전산화되는 정보가 가지고 있는 보안 수준을 보안 요소별로 등급을 구분하여 분류한다.
- 조직의 정보보호 관련 보안 정책을 참고하여 소프트웨어 개발에 적용할 수 있는 보안 정책 항목들의 출처, 요구 수준, 세부 내용 등을 문서화한다.

요구 수준 : 요구 수준은 해당 보안 정책 항목의 적용이 필수적인지 선택적인지를 의미한다. 예를 들어 주민번호는 정해진 수집 및 취급 방법이 법령에 있으므로 필수적으로 적용해야 한다

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

2) 요구사항 분석 단계에서의 보안 활동

● 보안요소

보안 요소는 소프트웨어 개발에 있어 충족시켜야 할 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 허용된다. •정보가 전송 중에 노출되더라도 데이터를 읽을 수 없다.
무결성	시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다.
가용성	인가 받은 사용자는 언제라도 사용할 수 있다.
인증	•시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다. •대표적 방법으로는 패스워드, 인증용 카드 지문 검사 등이 있다.
부인 방지	데이터를 송, 수신한 자가 송, 수신 사실을 부인할 수 없도록 송, 수신 증거를 제공한다.

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

3) 설계 단계에서의 보안 활동

; 설계 단계에서는 식별된 보안 요구사항들을 소프트웨어 설계서에 반영하고, 보안 설계서를 작성한다.

- 소프트웨어에서 발생할 수 있는 위협을 식별하여 보안대책, 소요예산, 사고 발생 시 영향 범위와 대응책 등을 수립한다.
- 네트워크, 서버, 물리적 보안, 개발 프로그램 등 환경에 대한 보안 통제 기준을 수립하여 설계에 반영한다.
  - 네트워크 : 외부의 사이버 공격으로부터 개발 환경을 보호하기 위해 네트워크를 분리하거나 방화벽을 설치한다.
  - 서버 : 보안이 뛰어난 운영체제를 사용하고 보안 업데이트, 외부접속에 대한 접근통제 등을 실시한다.
  - 물리적 보안 : 출입통제, 개발 공간 제한, 폐쇄회로 등의 감시설비를 설치한다.
  - 개발 프로그램 : 허가되지 않은 프로그램을 통제하고 지속적인 데이터 무결성 검사를 실시한다.

위협(Threat)
위협이란 불법적인 유출, 위조 변조 삭제 파손 등 소프트웨어 발생할 수 있는 재산상의 손해를 말한다.

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

4) 구현 단계에서의 보안 활동

; 구현 단계에서는 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며, 설계서에 따라 보안 요구사항들을 구현한다.

- 개발 과정 중에는 지속적인 단위 테스트를 통해 소프트웨어에 발생할 수 있는 보안 취약점을 최소화 해야 한다.
- 코드 점검 및 소스 코드 진단 작업을 통해 소스 코드의 안정성을 확보해야 한다.

표준 코딩 정의서 : 코딩 시 다른 개발자나 운영자가 쉽게 접근할 수 있도록 클래스 메소드 등의 네이밍 규칙, 주석 첨부 방식 등을 정의해 둔 문서이다.
소프트웨어 개발 보안 가이드 : 소프트웨어 개발 보안 가이드는 안전한 소프트웨어 개발을 위해 정부에서 제작하여 배포하고 있는 지침이다.
단위 테스트(Unit Test) : 프로그램의 단위 기능을 구현하는 모듈이 정해진 기능을 정확히 수행하는지 검증하는 것이다

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

5) 시큐어 코딩(Secure Coding)

; 시큐어 코딩은 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것을 의미한다.

- 보안 취약점을 사전에 대응하여 안정성과 신뢰성을 확보하기 위해 사용된다.
- 보안 정책을 바탕으로 시큐어 코딩 가이드를 작성하고, 개발 참여자에게는 시큐어 코딩 교육을 실시해야 한다.

5. 소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC)

6) 테스트 단계에서의 보안 활동

; 테스트 단계에서는 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검한다.

- 동적 분석 도구 또는 모의 침투 테스트를 통해 설계 단계에서 식별된 위협들의 해결 여부를 검증한다.
- 설계 단계에서 식별된 위협들 외에도 구현 단계에서 추가로 제시된 위협들과 취약점들을 점검할 수 있도록 테스트 계획을 수립하고 시행한다.
- 테스트 단계에서 수행한 모든 결과는 문서화하여 보존하고, 개발자에게 피드백 되어야 한다.

7) 유지보수 단계에서의 보안 활동

; 유지보수 단계에서는 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안사고들을 식별하고, 사고 발생 시 이를 해결하고 보안 패치를 실시한다.

동적 분석 도구 : 프로그램을 실제 또는 가상으로 실행시킨 상황에서 메모리 분석, 보안 취약점 검색 오류 탐지 등의 다양한 기능을 수행하는 소프트웨어이다.

소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(Secure SDLC)

1. 실무적으로 검증된 개발 보안 방법론 중 하나로, SW 보안의 모범 사례를 SDLC(Software Development Life Cycle)에 통합한 소프트웨어 개발 보안생명주기 방법론은?

① CLASP ② CWE

③ PIMS ④ Seven Touchpoints

'모범사례'하면, Seven Touchpoints라는 것을 떠올리자.

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 최소화 하기 위해 등장하였다.

▶ Secure SDLC는 요구사항 분석, 설계, 구현, 테스트, 유지보수 등 SDLC 전체 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.

▶ Secure SDLC의 대표적인 방법론

CLASP

▶ Secure Software 사에서 개발하였으며, SDLC의 초기 단계에서 보안을 강화하기 위해서 개발된 방법론이다.

▶ 활동 중심 역할 기반의 프로세스로 구성되어 있으며, 현재 운용

3. 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있는 보안 요소는?

① 기밀성 ② 부인방지

③ 가용성 ④ 무결성

기밀성은 인가된 사용자에게만 접근이 허용되는 것, 부인 방지는 송,수신 사실의 부인을 막기 위해 증거를 제공하는 것, 가용성은 인가 받은 사용자는 언제든지 자원을 사용할 수 있도록 하는 것

4. 소프트웨어 개발 보안 중 다음 설명에 해당하는 것은?

- 소프트웨어 개발 생명주기에 보안 프로세스를 포함하는 것이다.
- 유지 보수 단계에서 보안 문제를 해결하는데 큰 비용이 소모되는 것을 예방하기 위해 등장하였다.
- 대표적으로 CLASP, SDL이 있다.

① Secure Coding ② Secure SDLC

③ Secure Architecture ④ Secure Framework

Secure Coding 은 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것을 의미한다.

▶ 보안 취약점을 사전에 대응하여 안정성과 신뢰성을 확보하기 위해

소프트웨어 개발 보안 구축 - SEC_01(Secure SDLC) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(Secure SDLC)

5. Secure SDLC의 구현 단계에 대한 설명으로 가장 거리가 먼 것 은?

① 보안 요구사항들을 구현하는 단계이다.

② 설계 단계에서 작성한 보안 설계서에 따라 소프트웨어를 구현한다.

③ 지속적인 점검 및 진단작업으로 코드의 안정성을 확보한다.

④ 동적 분석도구의 사용 또는 모의 침투테스트를 통해 보안 위협들의 해결 여부를 검증한다.

테스트 단계에서의 보안 활동

테스트 단계에서는 설계 단계에서 작성한 보안 설계서를 방탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검한다.

▶ 동적 분석 도구 또는 모의 침투 테스트를 통해 설계 단계에서 식별된 위협들의 해결 여부를 검증한다.

▶ 설계 단계에서 식별된 위협들 외에도 구현 단계에서 추가로 제시된 위협들과 취약점들을 점검할 수 있도록 테스트 계획을 수립하고 시행한다.

▶ 테스트 단계에서는 수행한 모든 결과는 문서화하여 보존하고, 반드시 개발자에게 피드백 되어야 한다.

5. 소프트웨어 개발 보안 구축 - SEC_02(세션 통제)

1) 세션 통제의 개요

; 세션은 서버와 클라이언트의 연결을 의미하고, 세션 통제는 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것을 의미한다.

- 세션 통제는 소프트웨어 개발 과정 중 요구사항 분석 및 설계 단계에서 진단해야 하는 보안 점검 내용이다.
- 세션 통제의 보안 약점에는 불충분한 세션 관리, 잘못된 세션에 의한 정보 노출이 있다.

2) 불충분한 세션 관리

; 불충분한 세션 관리는 일정한 규칙이 존재하는 세션 ID가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점이다.

- 세션 관리가 충분하지 않으면 침입자는 세션 하이재킹과 같은 공격을 통해 획득한 세션 ID로 인가되지 않은 시스템의 기능을 이용하거나 중요한 정보에 접근할 수 있다.

세션 ID(Session ID) : 서버가 클라이언트들을 구분하기 위해 부여하는 키(Key)로, 클라이언트가 서버에 요청을 보낼 때마다 세션 ID 를 통해 인증이 수행된다.

5. 소프트웨어 개발 보안 구축 - SEC_02(세션 통제)

3) 세션 하이재킹(Session Hijacking)

; 세션 하이재킹은 서버에 접속하고 있는 클라이언트들의 세션 정보를 가로채는 공격 기법으로 세션 가로채기 라고도 한다.

- 정상적인 연결을 RST(Reset) 패킷을 통해 종료시킨 후 재 연결 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.

ACK Storm : 세션 하이재킹 과정 중에 패킷량이 비정상적으로 늘어나는 현상을 의미한다.

5. 소프트웨어 개발 보안 구축 - SEC_02(세션 통제)

4) 잘못된 세션에 의한 정보 노출

; 잘못된 세션에 의한 정보 노출은 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다.

- 싱글톤 패턴에서 발생하는 레이스컨디션으로 인해 동기화 오류가 발생하거나, 멤버 변수의 정보가 노출될 수 있다.
- 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.

다중 스레드(Multi-Thread) : 프로세스 내의 작업 단위로 시스템의 자원을 할당 받아 실행하는 프로그램의 단위를 스레드라고 하며, 두 개 이상의 스레드가 생성되어 동시 처리되는 다중 작업(Multi tasking)을 다중 스레드 또는 멀티 스레드라고 부른다.
멤버 변수(Member Variable) : 멤버 변수는 객체와 연결된 변수로 클래스 내에 선언되어 클래스의 모든 메소드들이 접근 가능한 변수이다. 멤버 필드라고도 부르며, 종류에는 클래스(정적, static) 변수, 인스턴스 변수가 있다.
싱글톤(Singleton) : 싱글톤은 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만, 여러 프로세스가 동시에 참조할 수는 없는 디자인 패턴이다.
레이스컨디션(Race Condition) : 레이스컨디션은 두 개 이상의 프로세스가 공용 지원을 획득하기 위해 경쟁하고 있는 상태를 의미한다.

5. 소프트웨어 개발 보안 구축 - SEC_02(세션 통제)

5) 세션 설계 시 고려 사항

- 시스템의 모든 페이지에서 로그아웃이 가능하도록 UI(User Interface)를 구성한다.
- 로그아웃 요청 시 할당된 세션이 완전히 제거되도록 한다.
- 세션 타임아웃은 중요도가 높으면 2 ~ 5분, 낮으면 15 ~ 30분으로 설정한다.
- 이전 세션이 종료되지 않으면 새로운 세션이 생성되지 못하도록 설계한다.
- 중복 로그인을 허용하지 않은 경우 클라이언트의 중복 접근에 대한 세션 관리 정책을 수립한다.
- 패스워드 변경 시 활성화된 세션을 삭제하고 재할당한다.

6) 세션 ID의 관리 방법

- 세션 ID는 안전한 서버에서 최소 128비트의 길이로 생성한다.
- 세션 ID의 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
- 세션 ID가 노출되지 않도록 URL Rewrite 기능을 사용하지 않는 방향으로 설계한다.
- 로그인 시 로그인 전의 세션 ID를 삭제하고 재할당한다.
- 장기간 접속하고 있는 세션 ID는 주기적으로 재할당되도록 설계한다.

URL Rewrite : 쿠키를 사용할 수 없는 환경에서 세션 ID 전달을 위해 URL에 세션 ID를 포함시키는 것이다.

소프트웨어 개발 보안 구축 - SEC_02(세션 통제) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(세션 통제)

1. 세션 하이재킹을 탐지하는 방법으로 거리가 먼 것은?

① FTP SYN SEGMENT 탐지

② 비동기화 상태 탐지

③ ACK STORM 탐지

④ 패킷의 유실 및 재전송 증가 탐지

세션 하이재킹(Session Hijacking)

세션 하이재킹은 서버에 접속하고 있는 클라이언트들의 세션 정보를 가로채는 공격 기법으로 세션 가로채기 라고도 한다.

▶ 정상적인 연결을 RTS(Reset) 패킷을 통해 종료시킨 후 재 연결 시 희생자(클라이언트)가 아닌 공격자에게 연결하는 방식이다.

▶ 공격자는 서버와 상호 간의 동기화 된 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.

FTP PASV DoS

DoS의 공격의 한 방법으로 FTP 서비스에 접속하여 서버가 응답 하기

3. 세션이 안전하게 관리되도록 코딩 되지 않았을 때 발생할 수 있는 문제점으로 옳은 것은?

① 교착상태나 동기화 오류 등이 발생할 수 있다.(TOCTOU 경쟁 조건)

② 세션 ID를 탈취하여 시스템의 기능을 이용하거나 중요 정보에 접근 할 수 있다.

③ 자원 고갈로 인해 서비스나 시스템에 장애가 발생할 수 있다.(종료되지 않는 반복문 또는 재귀함수)

④ 오류 메시지를 통해 시스템의 중요 정보가 노출될 수 있다.(오류 메시지를 통해 발생할 수 있는 문제점)

세션 ID의 관리 방법

▶ 세션 ID는 안전한 서버에서 최소 128비트의 길이로 생성한다.

▶ 세션 ID의 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.

▶ 세션 ID가 노출되지 않도록 URL Rewrite 기능을 사용하지 않는 방향으로 설계한다.

▶ 로그인 시 로그인 전의 세션 ID를 반드시 삭제하고 재할당한다.

▶ 장기간 접속하고 있는 세션 ID는 주기적으로 재할당되도록 설계한다.

교착 상태(Dead Lock)

두 개 이상의 프로세스나 스레드가 서로 공유 자원을 얻지 못해서 다음

소프트웨어 개발 보안 구축 - SEC_02(세션 통제) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(세션 통제)

5. 세션 ID를 관리하는 방법에 대한 설명으로 잘못된 것은?

① 안전한 서버에서 최소 길이 128bit의 세션 ID를 사용하는 것이 좋다.

② HASH 함수 등의 난수 알고리즘을 사용하여 세션 ID를 발급한다.

③ 쿠키를 사용할 수 없는 경우 URL Rewrite 기능을 사용하여 세션 ID를 관리한다.

④ 오래된 세션 ID는 주기적으로 재할당하도록 설계해야 한다.

세션 ID를 URL로 전달하는 URL Rewrite 기능을 사용하는 것은 오히려 보안을 위협하는 방법에 해당한다.

URL Rewrite : 쿠키를 사용할 수 없는 환경에서 세션 ID 전달을 위해 URL에 세션 ID를 포함시키는 것이다.

HASH 함수

HASH 함수는 임의의 길이를 갖는 메시지를 입력 받아 고정된 길이의 해시 값을 출력하는 함수이다. 암호 알고리즘에는 키가 사용되지만, 해시 함수는 키를 사용하지 않으므로 같은 입력에 대해서는 항상 같은 출력이 나오게 된다. 이러한 해시함수를 사용하는 목적은 메시지의 오류나 변조를 탐지할 수 있는 무결성을

5. 소프트웨어 개발 보안 구축 - SEC_03(입력 데이터 검증 및 표현)

1) 입력 데이터 검증 및 표현의 개요

; 입력 데이터 검증 및 표현은 입력 데이터로 인해 발생하는 문제들을 예방하기 위해 구현 단계에서 검증해야 하는 보안 점검 항목들이다.

- 입력 데이터로 인해 발생하는 문제를 예방하기 위해서는 소프트웨어 개발의 구현 단계에서 유효성 검증 체계를 갖추고, 검증되지 않은 데이터가 입력되는 경우 이를 처리할 수 있도록 구현해야 한다.
- 입력 데이터를 처리하는 객체에 지정된 자료형이 올바른지 확인하고, 일관된 언어 셋을 사용하도록 코딩한다.

언어 셋(Character Set) : 문자(Character)를 컴퓨터에서 처리하기 위해 사용하는 코드표를 의미하며, 종류에는 ASCII, UNICODE, UTF-8 등이 있다.

5. 소프트웨어 개발 보안 구축 - SEC_03(입력 데이터 검증 및 표현)

2) 입력 데이터 검증 및 표현의 보안 약점

; 입력 데이터 검증 및 표현과 관련된 점검을 수행하지 않은 경우 SQL 삽입, 자원 삽입, 크로스사이트 스크립팅(XSS), 운영체제 명령어 삽입 등의 공격에 취약해진다.

● 보안 약점의 종류

SQL 삽입 (SQL Injection)	•웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고, 관리자 인증을 우회하는 보안 약점이다. •동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지할 수 있다.
경로 조작 및 자원 삽입	•데이터 입,출력 경로를 조작하여 서버 자원을 수정 삭제할 수 있는 보안 약점이다. •사용자 입력 값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지할 수 있다.
크로스사이트 스크립팅 (XSS; Cross Site Scripting)	•웹 페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점이다. •HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 '<' , '>', '&' 등의 문자를 다른 문자로 치환함으로써 방지할 수 있다.
운영체제 명령어 삽입	•외부 입력 값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점이다. •웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력값을 검증 없이 내부 명령어로 사용하지 않음으로써 방지할 수 있다.

동적 쿼리(Dynamic Query) : 동적 쿼리는 질의어 코드를 문자열 변수에 넣어 조건에 따라 질의를 동적으로 변경하여 처리하는 방식을 의미한다.
경로 순회(Directory Traversal) : 경로를 탐색할 때 사용하는 '/', '₩', '.. ' 등의 기호를 악용하여 허가되지 않은 파일에 접근 하는 방식이다.

5. 소프트웨어 개발 보안 구축 - SEC_03(입력 데이터 검증 및 표현)

2) 입력 데이터 검증 및 표현의 보안 약점

● 보안 약점의 종류

위험한 형식 파일 업로드	•악의적인 명령어가 포함된 스크립트 파일을 업로드 함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점이다. •업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹 사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 등의 방법으로 방지할 수 있다.
신뢰되지 않는 URL 주소로 자동접속 연결	•입력 값으로 사이트 주소를 받는 경우 이를 조직하여 방문자를 피싱 사이트로 유도하는 보안 약점이다. •연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지할 수 있다.
메모리 버퍼 오버플로	•연속된 메모리 공간을 사용하는 프로그램에서 할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 할 때 발생하는 보안 약점이다. •프로그램의 오동작을 유발시키거나 악의적인 코드를 실행시켜 공격자가 프로그램을 통제할 수 있는 권한을 획득하게 한다. •메모리 버퍼를 사용할 경우 적절한 버퍼의 크기를 설정하고, 설정된 범위의 메모리 내에서 올바르게 읽거나 쓸 수 있도록 함으로써 방지할 수 있다.

스크립트(Script) : 소프트웨어를 수행하는데 필요한 처리 절차가 기록된 텍스트로 대표적인 스크립트 파일의 확장자에는 .asp, .jsp, .php 등이 있다.

소프트웨어 개발 보안 구축 - SEC_03(입력 데이터 검증 및 표현) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(입력 데이터 검증 및 표현)

1. SQL Injection 공격과 관련한 설명으로 틀린 것은?

① SQL Injection은 임의로 작성한 SQL 구문을 애플리케이션에 삽입하는 공격 방식이다.

② SQL Injection 취약점이 발생하는 곳은 주로 웹 애플리케이션 과 데이터베이스가 연동되는 부분이다.

③ DBMS의 종류와 관계없이 SQL Injection 공격 기법은 모두 동일 하다.

④ 로그인과 같이 웹에서 사용자의 입력 값을 받아 데이터베이스 SQL문으로 데이터를 요청하는 경우 SQL Injection을 수행할 수 있다.

DBMS의 종류가 많다 보니 접근하는 방법이 모두 다르므로 공격 기법 또한 달라질 수 밖에 없다.

SQL 삽입(SQL Injection)

▶ 웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고, 관리자 인증을 우회하는 보안 약점이다.

▶ 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않도록 필터링 되도록 설정하여 방지할 수 있다.

3. 다음 내용이 설명하는 소프트웨어 취약점은?

메모리를 다루는 데 오류가 발생하여 잘못된 동작을 하는 프로그램 취약점

① FTP 바운스 공격

② SQL 삽입

③ 버퍼 오버플로

④ 디렉토리 접근 공격

메모리 버퍼 오버플로

▶ 연속된 메모리 공간을 사용하는 프로그램에서 할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 할 때 발생하는 보안 약점이다.

▶ 프로그램의 오동작을 유발시키거나 악의적인 코드를 실행시켜 공격자가 프로그램을 통제할 수 있는 권한을 획득하게 한다.

▶ 메모리 버퍼를 사용할 경우 적절한 버퍼의 크기를 설정하고, 설정된 범위의 메모리 내에서 올바르게 읽거나 쓸 수 있도록 함으로써 방지할 수 있다. 그리고 자원을 코드에서 열었다면, 반드시 닫아야(해제)한다.

FTP 바운스 공격(FTP bounce attack)

FTP 프로토콜의 구조의 허점을 이용한 공격 방법이다. FTP서버는

소프트웨어 개발 보안 구축 - SEC_03(입력 데이터 검증 및 표현) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(입력 데이터 검증 및 표현)

5. 크로스사이트 스크립팅(XSS)과 관련된 내용으로 옳지 않은 것은?

① 악의적인 스크립트 파일을 업로드 함으로써 시스템에 손상을 주는 보안 약점이다.

② 게시판이나 메일 등에 HTML 태그 또는 스크립트 명령어를 삽입하는 방식을 이용한다.

③ 공격 대상 사이트의 장애를 유발하거나, 방문자들의 정보를 탈취하는 용도로 사용된다.

④ 입력 데이터에 대해 올바른 유효성 검증 체계를 갖추지 않은 경우 발생할 수 있다.

위험한 형식 파일 업로드

▶ 악의적인 명령어가 포함된 스크립트 파일을 업로드 함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점이다.

▶ 업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹 사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 방법으로 방지할 수 있다.

6. 신뢰되지 않은 URL 주소에 자동 접속으로 연결할 경우 발생할

7. 다음 중, 입력 데이터 검증 및 표현은 어떤 단계에서 이루어지는가?

① 구현 단계　② 테스트 단계

③ 설계 단계　④ 분석 단계

입력 데이터 검증 및 표현

입력 데이터 검증 및 표현은 입력 데이터로 인해 발생하는 문제들을 예방하기 위해 **구현 단계**에서 검증해야 하는 보안 점검 항목들이다.

▶ 입력 데이터로 인해 발생하는 문제를 예방하기 위해서는 소프트웨어 개발의 구현 단계에서 유효성 검증 체계를 갖추고, 검증되지 않은 데이터가 입력되는 경우 이를 처리할 수 있도록 구현해야 한다.

▶ 입력 데이터를 처리하는 객체에 지정된 자료형(Data Type)이 올바른지 확인하고, 일관된 언어 셋(문자 셋)을 사용하도록 코딩한다.

보안 요소는 소프트웨어 개발에 있어 반드시 충족시켜야 할 요소 및 요건을 의미한다.

보안 요소에는 기밀성, 무결성, 가용성, 인증, 부인 방지가 있다.

5. 소프트웨어 개발 보안 구축 - SEC_04(보안 기능)

1) 보안 기능의 개요

; 보안 기능은 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목들이다.

- 각 보안 기능들은 서비스 환경이나 취급 데이터에 맞게 처리될 수 있도록 구현해야 한다.
- 소프트웨어의 기능 또는 데이터에 접근하려는 사용자 별로 중요도를 구분하고, 차별화된 인증 방안을 적용한다.
- 인증된 사용자가 이용할 기능과 데이터에 대해 개별적으로 접근 권한을 부여하여 인가되지 않은 기능과 데이터로의 접근을 차단한다.
- 개인정보나 인증정보와 같은 중요한 정보의 변조·삭제·오남용 등을 방지하기 위해 안전한 암호화 기술을 적용한다.

5. 소프트웨어 개발 보안 구축 - SEC_04(보안 기능)

2) 보안 기능의 보안 약점

; 보안 기능에 대한 점검을 수행하지 않을 경우 인증 없이 중요한 기능을 허용하거나 비밀번호가 노출되는 등 다음과 같은 보안 약점이 발생할 수 있다.

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

보안인증 사무국 : 정보 보호 제품의 평가·인증을 수행하고 인증제품 목록을 공개 및 관리하는 국가 보안 기술 연구소 산하의 기관
HTTPS(Hypertext Transfer Protocol Secure) : 웹 브라우저와 서버 간의 안전한 통신을 위해 HTTP와 암호 통신규약을 결합한 것.
SSL(Secure Sockets Layer) : 데이터를 송,수신하는 두 컴퓨터 사이에 위치하여 인증, 암호화, 무결성을 보장하는 업계 표준 프로토콜.
하드코드 : 데이터를 코드 내부에 직접 입력하여 프로그래밍하는 방식
디폴트 패스워드(Default Password) : 사용자를 등록하기 전에 설치 권한을 획득하기 위해 사용되는 초기 설정 암호.

소프트웨어 개발 보안 구축 - SEC_04(보안 기능) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(보안 기능)

1. 다음 JAVA 코드에서 밑줄로 표시된 부분에는 어떤 보안 약점이 존재하는가? (단, key는 암호화 키를 저장하는 변수이다.)

```
//생략
public String encripString(String usr) throws Exception {

    byte[] bToEncrypt = new byte[100];
    String key = "22df3023sf~2iasn!@#/>as";
    if (key != null)
        bToEncrypt = usr.getBytes("UTF-8");
    //생략
```

① 무결성 검사 없는 코드 다운로드

② 중요 자원에 대한 잘못된 권한 설정

③ 하드코드된 암호화 키 사용

④ 적절한 인증 없는 중요 기능 허용

비밀키(Secret Key), 문자열 암호화(encriptString) 등의 메서드 명으로 보아 암호키를 관리하는 코드로 유추할 수 있는데, Key 라는 문자열 변수에 값이 직접 입력된 것으로 하드코드 된 암호화 키가 답임을 알 수가 있다.

하드코드된 비밀번호

▶ 소스코드를 유출 시에 내부에 하드코드 패스워드를 이용하여 관리자 권한을 탈취할 수 있다.

3. 소프트웨어 개발의 구현 단계에서 보안 기능의 점검 미비로 인해 발생할 수 있는 보안 약점에 해당하지 않는 것은?

① 종료되지 않은 반복문 또는 재귀함수

② 부적절한 인가

③ 중요한 자원에 대한 잘못된 권한 설정

④ 적절한 인증 없이 중요기능 허용

종료되지 않은 반복문 또는 재귀함수로 인하여 자원 고갈로 인해 서비스나 시스템에 장애가 발생할 수 있다.

부적절한 인가

▶ 접근 제어 기능이 없는 실행 경로를 통해 정보 또는 권한을 탈취할 수 있다.

▶ 모든 실행 경로에 대한 접근 제어 검사를 수행하고, 사용자에게는 반드시 필요한 접근 권한만 부여하여 방지할 수 있다.

중요한 자원에 대한 잘못된 권한 설정

▶ 권한 설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용할 수 있다.

▶ 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않은 사용자의 중요 자원에 대한 접근 여부를 검사함으로써 방지

소프트웨어 개발 보안 구축 - SEC_04(보안 기능) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(보안 기능)

5. 취약한 암호화 알고리즘 사용 시 발생하는 보안 약점에 대한 설명으로 가장 옳지 않은 것은?

① 충분히 오래 사용되어 검증된 암호화 알고리즘을 사용하여 예방할 수 있다.

② 암호화된 환경설정 파일을 해독하여 중요정보를 탈취할 수 있다.

③ IT 보안 인증 사무국의 인증제품 목록을 참고하여 암호화 알고리즘을 선정한다.

④ 소프트웨어 개발의 구현 단계에서 검증해야 하는 보안 점검 항목이다.

암호 알고리즘은 점점 보안성이 높아지고 있기 때문에 오래된 알고리즘 사용을 하지 않고 IT 보안 인증 사무국이 인정한 강력 하고 안전한 암호화 알고리즘을 사용하는 것이 좋다.

취약한 암호화 알고리즘 사용

▶ 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요 정보 를 탈취할 수 있다.

▶ 안전한 암호화 알고리즘을 이용하고, 업무관련 내용이나 개인 정보 등에 대해서는 IT 보안 인증 사무국이 안정성을 확인한 암호

5. 소프트웨어 개발 보안 구축 - SEC_05(코드 오류)

1) 코드 오류의 개요

; 코드 오류는 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형(Type)변환, 자원 반환 등의 오류를 예방하기 위한 보안 점검 항목들이다.

● 코드 오류로 발생할 수 있는 보안 약점에는 널 포인터 역참조, 부적절한 자원 해제, 해제된 자원 사용, 초기화되지 않은 변수 사용이 있다.

널 포인터(Null Pointer) : 널(Null)은 값이 없음을 의미하며, 포인터(Pointer)는 메모리의 위치를 가리키는 요소이다. 널 포인터(Null Pointer)는 포인터에 널이 저장되어 어떠한 곳도 가리키지 못하는 상태의 요소를 말한다.

5. 소프트웨어 개발 보안 구축 - SEC_05(코드 오류)

2) 널 포인터(Null Pointer) 역참조

; 널 포인터 역참조는 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다.

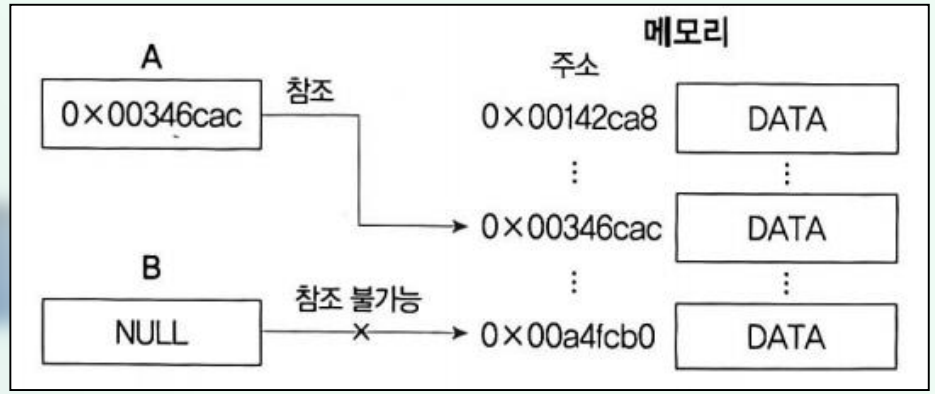
- 많은 라이브러리 함수들이 오류가 발생할 경우 널 값을 반환하는데, 이 반환값을 포인터로 참조하는 경우 발생한다.
- 대부분의 운영체제에서 널 포인터는 메모리의 첫 주소를 가리키며, 해당 주소를 참조할 경우 소프트웨어가 비정상적으로 종료될 수 있다.
- 공격자가 널 포인터 역참조로 발생하는 예외 상황을 악용할 수 있다.
- 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.

널 포인터 역참조로 오류가 발생하는 경우 "메모리 0x00000000을 참조하였습니다."라는 오류 메시지가 발생한다.

5. 소프트웨어 개발 보안 구축 - SEC_05(코드 오류)

2) 널 포인터(Null Pointer) 역참조

예제) 다음은 A와 B를 포인터로 해서 참조하는 메모리에 값을 저장하는 경우 발생하는 상황이다.



해설)

- A를 포인터로 해서 참조하는 경우 A에는 정상적인 메모리 주소가 저장되어 있으므로 해당 위치의 메모리에 값을 저장할 수 있다.
- B를 포인터로 해서 참조하는 경우 B에는 NULL이 저장되어 있어 참조가 불가능하여 오류가 발생한다. 공격자는 이러한 오류로 발생하는 예외 상황을 이용하여 추후 공격을 계획하는데 사용할 수 있다.

5. 소프트웨어 개발 보안 구축 - SEC_05(코드 오류)

2) 널 포인터(Null Pointer) 역참조

● **스택 가드(Stack Guard)**

- 널 포인터 역참조와 같이 주소가 저장되는 스택에서 발생하는 보안 약점을 막는 기술 중 하나이다.
- 메모리 상에서 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.

3) 부적절한 자원 해제

; 부적절한 자원 해제는 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점이다.

● 힙 메모리(Heap Memory), 소켓(Socket) 등의 유한한 시스템 자원이 계속 점유하고 있으면 자원 부족으로 인해 새로운 입력을 처리하지 못 할 수 있다.

● 프로그램 내에 자원 반환 코드가 누락되었는지 확인하고, 오류로 인해 함수가 중간에 종료되었을 때 예외 처리에 관계없이 자원이 반환되도록 프로그래밍 함으로써 방지할 수 있다.

힙 메모리(Heap Memory) : 힙 메모리는 소프트웨어가 자유롭게 사용할 수 있는 사용자 메모리 공간이다.
소켓(Socket) : 소켓은 데이터 교환을 위한 통로이다.

5. 소프트웨어 개발 보안 구축 - SEC_05(코드 오류)

4) 해제된 자원 사용

; 해제된 자원 사용은 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점이다.

- 반환된 메모리를 참조하는 경우 예상하지 못한 값 또는 코드를 수행하게 되어 의도하지 않은 결과가 발생할 수 있다.
- 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화함으로써 방지할 수 있다.

5) 초기화되지 않은 변수 사용

; 초기화 되지 않은 변수 사용은 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점이다.

- 변수가 선언되어 메모리가 할당되면 해당 메모리에 이전에 사용하던 내용이 계속 남아 있어 변수가 외부에 노출되는 경우 중요 정보가 악용될 수 있다.
- 변수 선언 시 할당된 메모리를 초기화함으로써 방지할 수 있다.

소프트웨어 개발 보안 구축 - SEC_05(코드 오류) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(코드 오류)

1. 메모리 상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장해 두었다가 그 값이 변경되었을 경우 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술은?

① 모드체크 ② 리커버리 통제

③ 시스로그 ④ 스택 가드

스택 가드(Stack Guard)

▶ 널 포인터 역참조와 같이 주소가 저장되어 있는 스택에서 발생하는 보안 약점을 막는 기술 중 하나이다.

▶ 메모리 상에서 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.

모드 체크(mode check)

입력될 수 있는 글자들의 종류가 제한된 경우 입력 문자를 조사하여 올바른 글자가 입력되었는지 검사하는 일을 의미한다.

복구(Recovery)

정보 시스템의 재해 및 장애 상황 발생 시 훼손된 자료를 일정

3. 널 포인터가 가리키는 곳에 데이터를 저장하는 경우 발생하는 보안 약점에 대한 설명으로 가장 옳지 않은 것은?

① 값이 없는 포인터 변수를 참조하여 데이터를 저장할 때 발생한다.

② 널 포인터는 메모리의 마지막 주소인 FxFFF...F를 가리킨다.

③ 널 포인터를 참조하는 경우 소프트웨어는 비정상적으로 종료될 수 있다.

④ 널 값을 가질 가능성이 있는 포인터 변수를 확인하여 사용 전에 널 여부를 검사하여 예방할 수 있다.

널 포인터(Null Pointer) 역참조

널 포인터 역참조는 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다.

▶ 많은 라이브러리 함수들이 오류가 발생할 경우 Null 값을 반환하는데, 이 반환값을 포인터로 참조하는 경우 발생한다.

▶ 대부분의 운영체제에서 널 포인터는 메모리의 첫 주소를 가리키며, 해당 주소를 참조할 경우 소프트웨어가 비정상적으로 종료될 수 있다.

▶ 공격자가 널 포인터 역참조로 발생하는 예외 상황을 악용할 수 있다.

▶ 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

1) 캡슐화의 개요

; 캡슐화는 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목들이다.

● 캡슐화로 인해 발생할 수 있는 보안 약점에는 잘못된 세션에 의한 정보 노출, 제거되지 않고 남은 디버그 코드, 시스템 데이터 정보 노출 등이 있다.

캡슐화 : 클래스 안에 서로 연관 있는 속성과 기능들을 하나의 캡슐(capsule)로 만들어 데이터를 외부로부터 보호하는 것을 말한다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

2) 잘못된 세션에 의한 정보 노출

; 잘못된 세션에 의한 정보 노출은 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다.

- 싱글톤 패턴에서 발생하는 레이스컨디션으로 인해 동기화 오류가 발생하거나, 멤버 변수의 정보가 노출될 수 있다.
- 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.

다중 스레드(Multi-Thread) : 프로세스 내의 작업 단위로 시스템의 자원을 할당 받아 실행하는 프로그램의 단위를 스레드라고 하며, 두 개 이상의 스레드가 생성되어 동시 처리되는 다중 작업(Multi tasking)을 다중 스레드 또는 멀티 스레드라고 부른다.
멤버 변수(Member Variable) : 멤버 변수는 객체와 연결된 변수로 클래스 내에 선언되어 클래스의 모든 메소드들이 접근 가능한 변수이다. 멤버 필드라고도 부르며, 종류에는 클래스(정적, static) 변수, 인스턴스 변수가 있다.
싱글톤(Singleton) : 싱글톤은 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만, 여러 프로세스가 동시에 참조할 수는 없는 디자인 패턴이다.
레이스컨디션(Race Condition) : 레이스컨디션은 두 개 이상의 프로세스가 공용 지원을 획득하기 위해 경쟁하고 있는 상태를 의미한다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

3) 제거되지 않고 남은 디버그 코드

; 제거되지 않고 남은 디버그 코드는 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점이다.

- 소프트웨어 제어에 사용되는 중요한 정보가 디버그 코드로 인해 노출될 수 있다.
- 디버그 코드에 인증 및 식별 절차를 생략하거나 우회하는 코드가 포함되어 있는 경우 공격자가 이를 악용할 수 있다.
- 소프트웨어를 배포하기 전에 코드 검사를 통해 남아있는 디버그 코드를 삭제함으로써 방지할 수 있다.

디버그(Debug) : 프로그래밍 과정 중에 발생하는 오류나 비정상적인 연산, 즉 버그를 찾고 수정하는 것이다.
이 과정을 디버깅(Debugging)이라 하기도 한다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

4) 시스템 데이터 정보 노출

; 시스템 데이터 정보 노출은 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩 했을 때 발생하는 보안 약점이다.

● 시스템 메시지를 통해 노출되는 메시지는 최소한의 정보만을 제공함으로써 방지할 수 있다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

5) Public 메소드로부터 반환된 Private 배열

; 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다.

- Public 메소드가 Private 배열을 반환하면 배열의 주소가 외부로 공개되어 외부에서 접근할 수 있게 된다.
- Private 배열을 별도의 메소드를 통해 조작하거나, 동일한 형태의 복제본으로 반환 받은 후 값을 전달하는 방식으로 방지할 수 있다.

접근 지정자(접근 제어자)

- 접근 지정자는 프로그래밍 언어에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어이다(접근 가능 : O , 접근 불가능 : X).

접근 지정자	클래스 내부	패키지 내부	하위 클래스	패키지 외부
Public	O	O	O	O
Protected	O	O	O	X
Default	O	O	X	X
Private	O	X	X	X

패키지(Package) : 패키지는 관련 클래스나 인터페이스 등을 하나로 모아둔 것이다.

5. 소프트웨어 개발 보안 구축 - SEC_06(캡슐화)

6) Private 배열에 Public 데이터 할당

; Private 배열에 Public 으로 선언된 데이터 또는 메소드의 파라미터를 저장할 때 발생하는 보안 약점이다.

- Private 배열에 Public 데이터를 저장하면 Private 배열을 외부에서 접근할 수 있게 된다.
- Public으로 선언된 데이터를 Private 배열에 저장할 때, 레퍼런스가 아닌 값을 직접 저장함으로써 방지할 수 있다.

파라미터(Parameter) : 파라미터는 메소드의 외부에서 전달된 값을 저장하는 변수로 매개변수 또는 형식 매개변수라고도 한다.
레퍼런스(Reference) : 레퍼런스를 전달 또는 할당한다는 것은 메모리의 위치(주소값)를 공유한다는 의미이다.
예를 들어, 배열 A를 선언하여 값을 저장하고 배열 B 선언 시 B = A라고 했을 때, 배열 B는 배열 A와 동일한 메모리를 공유 하게 된다. 즉, 배열 B에는 어떠한 값도 저장하지 않았지만 배열 A에 저장한 값들을 B를 통해 접근할 수 있게 되는 것이다.

소프트웨어 개발 보안 구축 - SEC_06(캡슐화) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(캡슐화)

1. 자바에서 사용하는 접근 제어자의 종류가 아닌 것은?

① internal ② private

③ default ④ public

자바의 접근 제어자의 종류에는 public, protected, default, private가 있다.

C언어에서 변수를 external과 internal로 구분할 수 있다.

external variable(전역 변수)은 모든 함수의 밖에서 선언되고, 모든 함수들에서 접근이 가능하다.

반면에 internal variable 혹은 local variable(지역 변수)은 함수 내부,, scope 안에서 선언된다.

지역 변수는 선언된 함수 안에서만 사용할 수 있고(더 정확히는 해당 Scope 내) 다른 함수에서는 사용할 수 없다. 즉 함수 안에 종속되어 존재한다. 함수 밖에서는 존재하지 않는 것으로 취급한다. 이렇듯 C 언어는 범위에 따라 변수의 접근을 제한한다.

2. 보안 점검 내용에서 캡슐화의 정의로 가장 적합한 것은?

① 인터페이스를 제외한 세부 내용이 은폐되도록 데이터와 함수 를 객체로 묶어 코딩하는 것

3. 캡슐화에 대한 점검이 충분하지 않을 때 발생하는 보안 약점 중 제거되지 않고 남은 디버그 코드와 관련된 설명으로 가장 옳지 않은 것은?

① 개발 중 버그 수정이나 결과값 확인을 위해 남겨둔 코드로 인해 발생하는 보안 약점이다.

② 디버그 코드에 포함된 제어 정보가 노출될 수 있다.

③ 디버그 코드를 이용하여 인증 및 식별 절차를 우회할 수 있다.

④ 최소한의 정보만 노출되도록 제한하여 방지할 수 있다.

디버그 코드는 완전히 삭제해야지 조금이라도 남아 있다면 문제를 야기시킬 가능성을 열어 둔 것이다.

제거되지 않고 남은 디버그 코드

제거되지 않고 남은 디버그 코드는 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점이다.

▶ 소프트웨어 제어에 사용되는 중요한 정보가 디버그 코드로 인해 외부에 노출될 수 있다.

▶ 디버그 코드에 인증 및 식별 절차를 생략하거나 우회하는 코드가 포함되어 있는 경우 공격자가 이를 악용할 수 있다.

▶ 소프트웨어를 배포하기 전에 코드 검사를 통해 남아있는 디버그

소프트웨어 개발 보안 구축 - SEC_06(캡슐화) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(캡슐화)

5. 소프트웨어 구현 과정에서 작성한 디버그 코드로 인해 발생하는 보안 약점에 대한 설명으로 잘못된 것은?

① 소프트웨어의 중요 정보가 디버그 코드로 인해 노출될 수 있다.

② 디버그 코드에 식별 절차를 생략할 수 있는 코드가 포함되어 있는 경우 공격자에 의해 악용될 수 있다.

③ 레이스컨디션으로 인한 동기화 오류가 발생할 수 있다.

④ 디버그 코드는 소프트웨어 배포 전 반드시 삭제해야 한다.

레이스컨디션으로 인한 동기화 오류는 잘못된 세션에 의해 발생한다.

잘못된 세션에 의한 정보 노출

잘못된 세션에 의한 정보 노출은 다중 스레드(multi thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다.

▶ 싱글톤 패턴에서 발생하는 레이스컨디션으로 인해 동기화 오류가 발생하거나, 멤버 변수의 정보가 노출될 수 있다.

▶ 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.

6. 객체지향 개념을 활용한 소프트웨어 구현과 관련한 설명 중 틀린 것은?

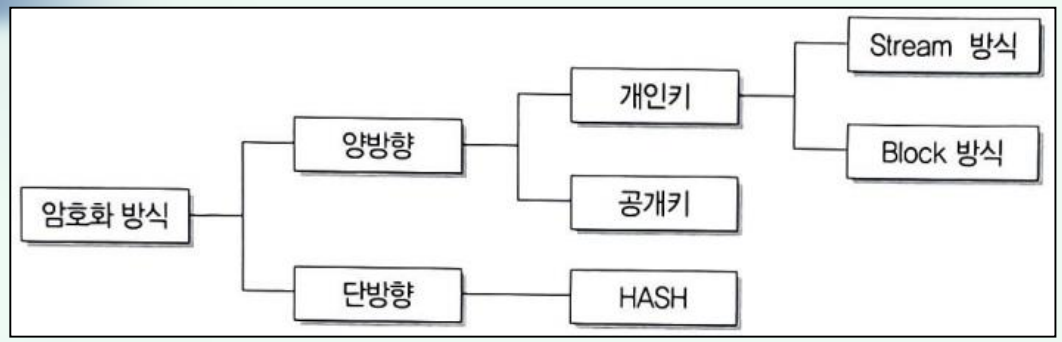
5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

1) 암호 알고리즘의 개요

; 암호 알고리즘은 패스워드, 주민번호, 은행계좌와 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미한다.

● 암호 알고리즘은 해시(Hash)를 사용하는 단방향 암호화 방식과, 개인키 및 공개키로 분류되는 양방향 암호화 방식이 있다.

● 암호 방식 분류



해시(Hash) : 데이터를 다루는 기법 중에 하나로 검색과 저장이 아주 빠르게 진행된다. 아주 빠르게 진행될 수 있는 이유는 데이터를 검색할 때 사용할 key와 실제 데이터의 값이 (value가) 한 쌍으로 존재하고, key값이 배열의 인덱스로 변환되기 때문에 검색과 저장의 평균적인 시간 복잡도가 O(1)에 수렴하게 된다. 하지만 암호 알고리즘에서의 해시는 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.

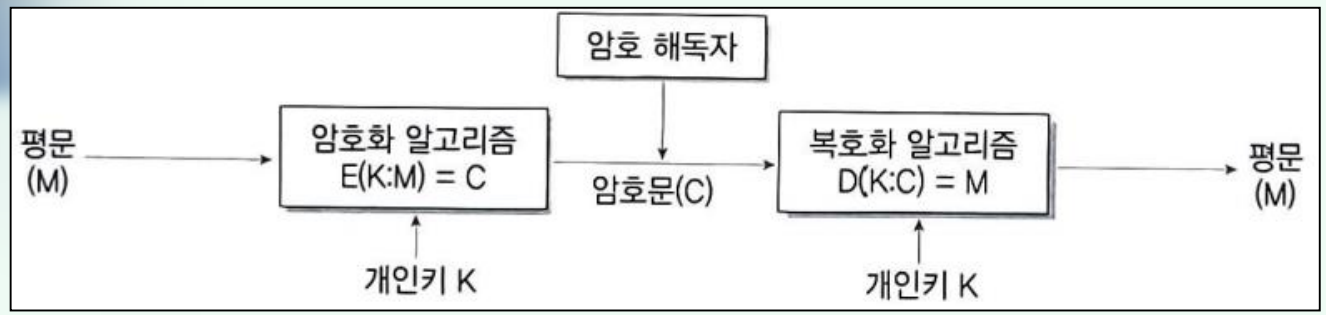
시간 복잡도(time complexity) : 컴퓨터 프로그램의 입력값과 연산 수행 시간의 상관관계를 나타내는 척도이다. 시간 복잡도와 로직의 수행 시간은 비례하므로 시간 복잡도 수치가 작을수록 효율적인 알고리즘임을 뜻한다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

2) 개인키 암호화(Private Key Encryption) 기법

; 개인키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화 한다.

- 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 n(n-1) / 2 이다.
- 개인키 암호화 기법은 대칭 암호 기법 또는 비밀키 암호화 기법이라고도 한다.
- 개인키 암호화 기법은 한 번에 하나의 데이터 블록을 암호화 하는 블록 암호화 방식과, 평문과 동일한 길이의 스트림을 생성하여 비트/바이트/워드 단위로 암호화 하는 스트림 암호화 방식으로 분류된다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

2) 개인키 암호화(Private Key Encryption) 기법

- 종류
  - 블록 암호화 방식 : DES, SEED, AES, ARIA, IDEA
  - 스트림 암호화 방식 : LFSR, RC4
- 장점 : 암호화/복호화 속도가 빠르며, 알고리즘이 단순하고, 공개키 암호 기법보다 파일의 크기가 작다.
- 단점 : 사용자의 증가에 따라 관리해야 할 키의 수가 상대적으로 많아진다.

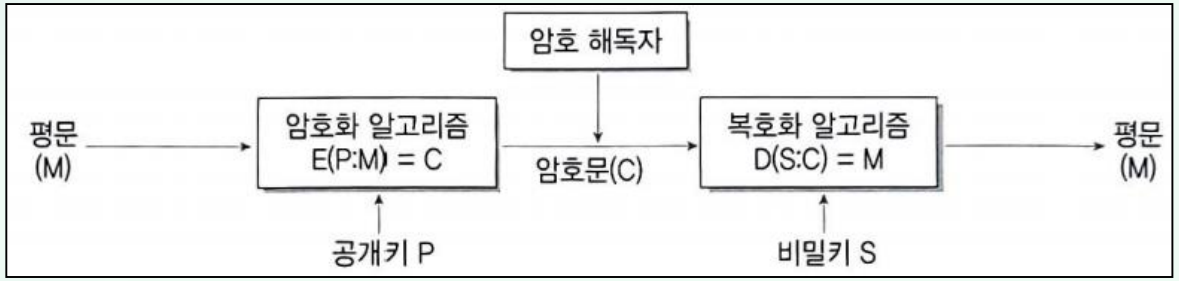
블록 암호화 방식 : 평문을 블록이라고 부르는 고정길이의 입력으로 나누어 블록 단위로 암호화 하는 기법
스트림 암호화 방식 : 난수 발생기를 이용, 생성된 키를 입력된 데이터에 비트 또는 바이트 단위로 암호화 하는 기법
DES(Data Encryption Standard) : 블록 암호 알고리즘의 한 종류로, DES 알고리즘은 암호문을 작성할 때 사용하는 암호키와 암호문을 해독할 때 사용하는 해독키가 같다. 따라서 이 키는 절대로 외부에 유출되지 않도록 관리해야 하여 비밀키(Secret Key)라고 부른다.
SEED : 전자 상거래, 금융, 무선통신 등에서 전송되는 중요 정보를 보호하기 위해 순수 국내 기술로 개발한 블록 암호 알고리즘이다.
AES : 블록 암호 알고리즘의 한 종류로 암호화 및 복호화 시 동일한 키를 사용하는 대칭키 알고리즘이다.
ARIA(아리아) : 경량 및 하드웨어 구현을 위해 최적화된, 범용 블록 알고리즘이며 차세대 국가 암호화 알고리즘이다.
IDEA(International Data Encryption Algorithm; 국제 데이터 암호화 알고리즘) : DES를 대체하기 위해서 스위스에서 개발 128 비트의 키로 64비트 블록을 암호화/복호화를 하는 대칭키 암호 알고리즘이다.
선형 피드백 시프트 레지스터(Linear feedback shift register, LFSR) : 스트림 암호 설계에서 가장 대중적으로 사용되는 키 스트림 생성기는 선형 피드백 시프트 레지스터라는 이진 스트림 생성기이다. LFSR은 하드웨어로 쉽게 구현할 수 있으며, 긴 의사 난수 결과를 생성하는 가장 좋은 방법 중에 하나이다.
RC4 : 스트림 암호 방식의 하나로 기본적으로 256 바이트 값의 배열이 들어있는 검색표이기 때문에 매우 간단하다. 단순히 순차적이지 않은 어떠한 테이블을 기반으로 정해진 규칙에 의해 키 스트림을 만들어내는 방식이다. 즉, 키가 배열을 구성하고, 배열이 키스트림을 구성하고, 키스트림이 암호문을 만들어낸다고 이해하면 좋다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

3) 공개키 암호화(Public Key Encryption) 기법

; 공개키 암호화 기법은 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화 할 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 복호화 하기 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.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

3) 공개키 암호화(Public Key Encryption) 기법

- 공개키 암호화 기법은 비대칭 암호 기법이라고도 하며, 대표적으로는 RSA(Rivest Shamir Adleman) 기법이 있다.
- 자신만이 보관하는 비밀키를 이용하여 인증, 전자서명 등에 적용이 가능하다.
- 장점 : 키의 분배가 용이하고, 관리해야 할 키의 개수가 적다.
- 단점 : 암호화/복호화 속도가 느리며, 알고리즘이 복잡하고, 개인키 암호화 기법보다 파일의 크기가 크다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

3) 공개키 암호화(Public Key Encryption) 기법

● 양방향 알고리즘 종류

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

SEED	•1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘이다. •블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류된다.
ARIA(Academy, Research Institute, Agency)	•2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘이다 •ARIA는 학계(Academy), 연구기관(Research Institute), 정부(Agency)의 영문 앞 글자로 구성되었다. •블록 크기는128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다.
DES(Data encryption Standard)	•1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘이다. •DES를 3번 적용하여 보안을 더욱 강화한 3DES(Triple DES)도 있다. •블록 크기는 64비트이며, 키 길이는 56비트이다.
AES(Advanced Encryption Standard)	•2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘이다 •DES의 한계를 느낀 NIST에서 공모한 후 발표하였다. •블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다.
RSA(Rivest Shamir Adleman)	•1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘이다. •큰 숫자를 소인수 분해 하기 어렵다는 것에 기반하여 만들어졌다. •공개키와 비밀키를 사용하는데, 여기서 키란 메시지를 열고 잠그는 상수(Constant)를 의미한다.

NBS(National Bureau of Standards) : NBS는 미국 표준 기술 연구소(NIST)의 과거 이름이다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

4) 해시(Hash)

; 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.
- 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- 복호화가 거의 불가능한 일방향 함수에 해당한다.

일방향 함수(One-way Function) : 평문을 암호화하여 암호문으로는 변경할 수 있으나, 암호문을 복호화 하여 평문으로 변경하는 것은 불가능한 함수를 의미한다. 이와 달리 암호화와 복호화가 모두 가능한 함수를 양방향 함수(Two way Function)라고 한다.

5. 소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘)

4) 해시(Hash)

● 해시 함수의 종류에는 SHA 시리즈, MD5, N-NASH, SNEFRU 등이 있다.

SHA 시리즈	•1993년 미국 국가 안보국(NSA)이 처음 설계했으며, 미국 국립표준기술 연구소(NIST)에 의해 발표되었다. •초기 개발된 SHA-0 이후 SHA-10이 발표되었고, 다시 SHA-2라고 불리는 SHA-224, SHA-256, SHA-384, SHA-512 발표되었다.
MD5	•1991년 R.Rivest가 MD4를 대체하기 위해 고안한 암호화 해시 함수이다. •블록 크기는 512비트이며, 키 길이는 128비트이다.
N-NASH	•1989년 일본의 전신전화주식회사(NTT)에서 발표한 암호화 해시 함수이다. •블록 크기와 키 길이가 모두 128비트이다.
SNEFRU	•1990년 R.C.Merkle에 의하여 발표된 해시 함수이다. •32비트 프로세서에서 구현을 용이하게 할 목적으로 개발되었다. •블록 크기는512비트이며, 키 길이에 따라 128 256으로 분류된다.

● 솔트(Salt)

- 둘 이상의 계정에 대해 패스워드를 'qwer1234' 라고 지정하고, 같은 암호화 알고리즘을 적용하게 되면 결과도 마찬가지로 동일하게 나타난다. 이 경우 공격자가 나타난다면 하나의 암호만 해제 해도 둘 이상의 계정을 얻게 된다. 이를 방지하고자 암호화를 수행하기에 앞서 원문에 무작위의 값을 덧붙이는 과정을 수행하는데, 이 때 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 되는 것이다.

소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(캡슐화)

1. 대칭 암호 알고리즘과 비대칭 암호 알고리즘에 대한 설명으로 틀린 것은?

① 대칭 암호 알고리즘은 비교적 실행 속도가 빠르기 때문에 다양한 암호의 핵심 함수로 사용될 수 있다.

② 대칭 암호 알고리즘은 비밀키 전달을 위한 키 교환이 필요하지 않아 암호화 및 복호화의 속도가 빠르다.

③ 비대칭 암호 알고리즘은 자신만이 보관하는 비밀키를 이용하여 인증, 전자서명 등에 적용이 가능하다.

④ 대표적인 대칭키 암호 알고리즘으로는 AES, IDEA 등이 있다.

대칭 암호 알고리즘(개인키 암호 알고리즘)은 동일한 키로 데이터를 암호화하고 복호화 해야 하므로 암호화 한 키를 수신자에게 전달해야 수신가 복호화를 할 수 있다. 그렇기에 비밀키 전달을 위한 키 교환이 필요하다.

개인키 암호화(Private Key Encryption) 기법

개인키 암호화 기법은 동일한 키로 데이터를 암호화 하고 복호화 한다.

▶ 데이터베이스 사용자는 평문의 정보 M을 암호화 알고리즘 E와

3. 공개키 암호화 방식에 대한 설명으로 틀린 것은?

① 공개키로 암호화된 메시지는 반드시 공개키로 복호화 해야 한다.

② 비대칭 암호 기법이라고도 한다.

③ 대표적인 기법은 RSA 기법이 있다.

④ 키 분배가 용이하고, 관리해야 할 키 개수가 적다.

공개키 암호화 기법은 암호화 할 때는 공개키(Public key)를 복호화 할 때는 비밀키(Secret Key)를 사용한다.

공개키 암호화(Public Key Encryption) 기법

공개키 암호화 기법은 데이터를 암호화 할 때는 사용하는 공개키(Public key)는 데이터베이스 사용자에게 공개하고, 복호화 할 때는 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 복호화 하기 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.

소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(캡슐화)

5. 큰 숫자를 소인수 분해하기 어렵다는 기반 하에 1978년 MIT에 의해 제안된 공개키 암호화 알고리즘은?

① DES ② ARIA

③ SEED ④ RSA

DES, ARIA, SEED는 모두 개인키 암호화 기법이다.

SEED

▶ 1999년 한국 인터넷 진흥원(KISA)에서 개발한 블록 암호화 알고리즘이다.

▶ 블록 크기는 128비트이며, 키 길이에 따라서 128, 256으로 분류된다.

ARIA(Academy, Research Institute, Agency)

▶ 2004년 국가 정보원과 산학 연합회가 개발한 블록 암호화 알고리즘이다.

▶ ARIA는 학계, 연구기관, 정부의 영문 앞 글자로 구성되었다.

▶ 블록 크기는 128비트이며, 키 길이에 따서 128, 192, 256으로 분류된다.

DES(Data Encryption Standard)

7. 스트림 암호화 방식의 설명으로 옳지 않은 것은?

① 비트/바이트/단어들을 순차적으로 암호화한다.

② 해쉬 함수를 이용한 해쉬 암호화 방식을 사용한다.

③ RC4는 스트림 암호화 방식에 해당한다.

④ 대칭키 암호화 방식이다.

스트림 암호화 방식 : 난수 발생기를 이용하여 생성된 키를 입력된 데이터에 비트 또는 바이트 단위로 암호화 하는 기법을 의미한다. 스트림 암호화 방식은 개인키(대칭키, 비밀키) 암호화 기법으로 암호화와 복호화가 반드시 필요하다. 따라서 **복호화가 불가능한 해시 암호화 방식은 사용하지 못한다.**

8. 공개키 암호에 대한 설명으로 틀린 것은?

① 10명이 공개키 암호를 사용할 경우 5개의 키가 필요하다.

② 복호화 키는 비공개 되어 있다.

③ 송신자는 수신자의 공개키로 문서를 암호화한다.

④ 공개키 암호로 널리 알려진 알고리즘은 RSA가 있다.

공개키 암호화 기법에서 키의 개수는 2이므로, 10명이 암호를 사용할 때 키의 개수를 구하는 **공식인 2n**에 따라 계산을 하면 2 * 10 = 20개의 키가 필요하다.

소프트웨어 개발 보안 구축 - SEC_07(암호 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(캡슐화)

9. 암호화 키와 복호화 키가 동일한 암호화 알고리즘은?

① RSA ② AES

③ DSA ④ ECC

암호화 키와 복호화 키가 동일한 암호화 알고리즘은 개인키 (비밀키) 암호화 기법을 의미하며, 종류에는 AES, DES, SEED, ARIA, IDEA 등이 있다. RSA는 공개키 암호화 알고리즘의 기법이다.

DSA

▶ 1991년에 미국 국립 표준 기술 연구소(NIST)가 공표한 디지털 서명 방법이다. 그 후 미국 연방 표준(FIPS 186)으로 발전되었고, 현재는 디지털 서명 표준(DSS; Digital Signature Standards) 라고도 한다.

ECC(Elliptic Curve Cryptography)

▶ 짧은 키로도 동일한 암호 성능을 가지는데, 이는 컴퓨터 성능 이 낮아도 암호 성능을 유지할 수 있게 되었다. 하여, 이러한 이유로 인해 RSA를 대체할 차세대 공개키 암호 기술로 부상하고 있다. 타원 곡선 암호화 알고리즘이라고도 불리며, 공개키 암호화 방식이다.

11. 다음 암호 알고리즘 중 성격이 다른 하나는?

① MD4 ② MD5

③ SHA-1 ④ AES

AES는 개인키 암호화 알고리즘이고, MD4, MD5, SHA시리즈는 해시 알고리즘이다.

MD4

▶ 메시지-다이제스트 알고리즘(MD4; Message-Digest Algorithm)이고 1990년 로널드 라이베스트가 개발한 암호화 해시 함수이다. 다이제스트 길이는 128비트이다. 이 알고리즘은 MD5, SHA-1 알고리즘과 같은 이후의 디자인에 영향을 주었다. MD는 Message-Digest의 약어이다.

SHA 시리즈

▶ 1993년 미국 국가 안보국(NSA)이 처음 설계했으며, 미국 국립 표준 기술 연구소(NIST)에 의해 발표되었다.

▶ 초기 개발된 SHA-0 이후 SHA-10이 발표되었고, 다시 SHA-2라고 불리는 SHA-224, SHA-256, SHA-384, SHA-512까지 발표되었다.

MD5

▶ 1991년 R.Rivest가 MD4를 대체하기 위해 고안한 암호화 해시 함수이다.

▶ 블록 크기는 512비트이며, 키 길이는 128비트이다.

감사합니다.