Char9. 소프트웨어 개발 보안

소프트웨어 개발 보안

소프트웨어 개발 과정에서 발생할 수 있는 보안 취약점을 최소화하여 보안 위협으로부터 안전한 소프트웨어를 개발하기 위한 일련의 보안 활동

데이터의 **기밀성 Confidentiality, 무결성 Integrity, 가용성 Availability**을 유지하는 것이 목표

- 소프트웨어 **보안 취약점**이 발생하는 경우

1. 보안 요구사항이 정의되지 않는 경우
2. 소프트웨어 설계 시 논리적 오류가 포함된 경우
3. 기술 취약점을 갖고 있는 코딩 규칙을 적용한 경우
4. 소프트웨어의 배치가 적절하지 않는 경우
5. 보안 취약점 발견 시 적절하게 대응하지 못한 경우

- 소프트웨어 개발 보안 관련 기관

행정안전부	• **소프트웨어 개발 보안 정책을 총괄** • **소프트웨어 개발 보안 관련 법규, 지침, 제도를 정비** • **소프트웨어 보안 약점을 진단하는 사람의 양성 및 관련 업무를 수행**
한국인터넷진흥원 KISA	• **소프트웨어 개발 보안 정책 및 가이드를 개발** • **소프트웨어 개발 보안에 관한 기술을 지원하고 교육과정 및 자격제도를 운영**
발주기관	• **소프트웨어 개발 보안의 계획을 수립** • **소프트웨어 개발 보안 사업자 및 감리법인을 선정**
사업자	• **소프트웨어 개발 보안 관련 기술 수준 및 적용 계획을 명시** • **소프트웨어 개발 보안 관련 인력을 대상으로 교육 실시** • **소프트웨어 개발 보안 가이드를 참조하여 개발** • **자체적으로 보안 약점을 진단하고 제거** • **소프트웨어 보안 약점과 관련된 시정 요구사항을 이행**
감리법인	• **감리 계획을 수립하고 협의** • **소프트웨어 보안 약점의 제거 여부 및 조치 결과를 확인**

- 소프트웨어 개발 보안 활동 관련 법령

개인 정보 보호법	개인정보의 처리 및 보호에 관한 사항을 정함으로써 개인의 자유와 권리를 보호
정보통신망 이용 촉진 및 정보 보호 등에 관한 법률	정보통신망의 이용 촉진 및 정보통신 서비스를 이용하는 이용자들의 개인정보를 보호
신용정보의 이용 및 보호에 관한 법률	개인 신용정보의 효율적 이용과 체계적인 관리를 통해 정보의 오남용을 방지
위치정보의 보호 및 이용에 관한 법률	개인 위치정보의 안전한 이용 환경을 조성하여 정보의 유출이나 오남용을 방지
표준 개인정보 보호 지침	개인정보의 처리에 관한 기준, 개인정보 침해의 유형 및 예방조치 등에 관한 세부사항을 규정
개인정보의 안전성 확보 조치 기준	개인정보 처리자가 개인정보를 처리하는데 있어 개인정보가 분실, 도난, 유출, 위조, 변조, 훼손되지 않도록 안전성 확보에 필요한 기술적, 관리적, 물리적 안전조치에 관한 최소한의 기준을 규정
개인정보 영향평가에 관한 고시	개인정보 영향평가를 위한 평가기관의 지정, 영향평가의 절차 등에 관한 세부기준을 규정

- 소프트웨어 개발 보안 활동 관련 기타 규정

RFID 프라이버시 보호 가이드라인	RFID 시스템의 이용자들의 프라이버시를 보호하고 안전한 RFID 이용환경을 조정하기 위한 가이드라인
위치정보의 보호 및 이용 등에 관한 법률	개인 위치 정보의 유출 및 오남용을 방지하기 위한 법률

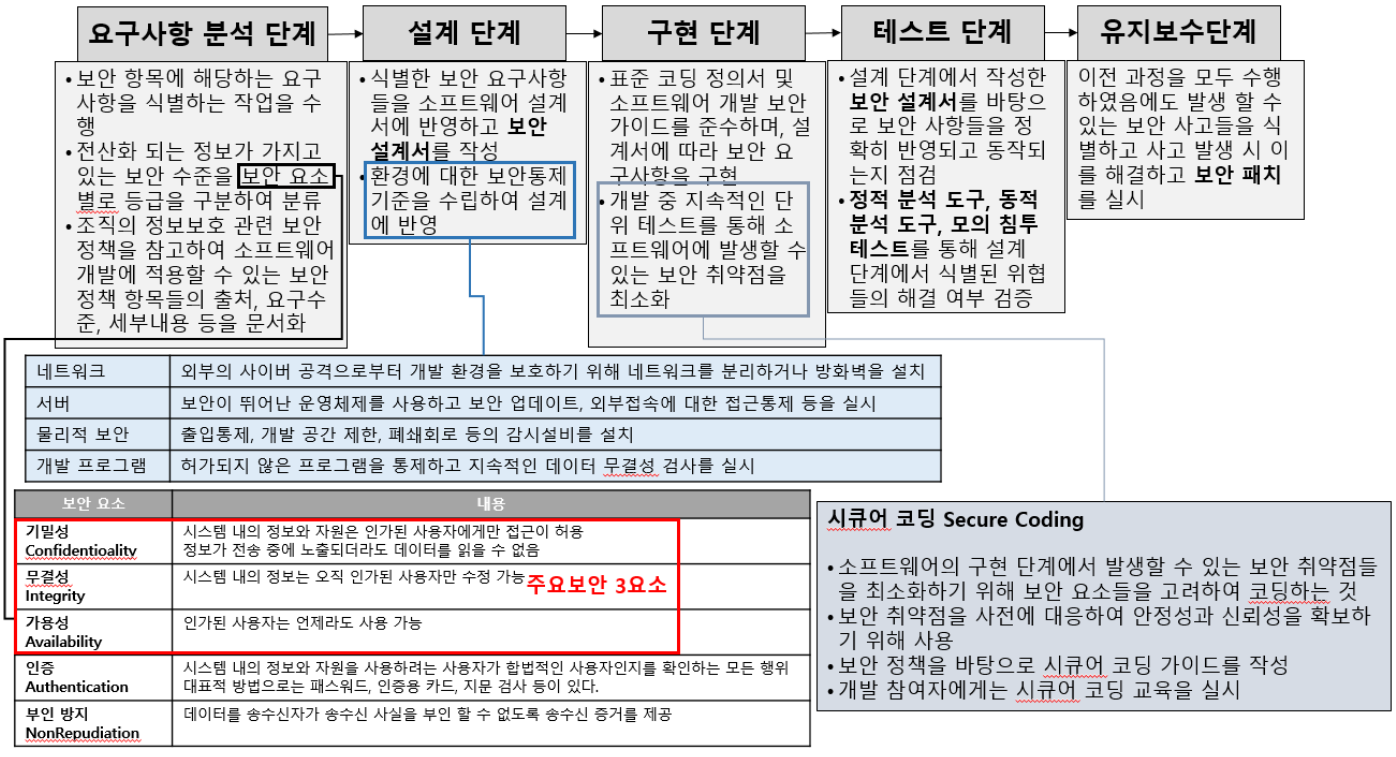
Secure SDLC

보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안강화를 위한 프로세스를 포함한 것을 의미

소프트웨어의 유지보수단계에서 보안 이슈를 해결하기 위해 소모되는 많은 비용을 최소화하기 위해 등장

대표적 방법론으로는 Secure Software사의 CLASP, Microsoft사의 SDL이 있다.

SDLC 전체 단계에 걸쳐 수행되어야 할 보안 활동을 제시



세션 통제

세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것을 의미

소프트웨어 개발 과정 중 요구사항 분석 및 설계 단계에서 진단해야 하는 보안 점검 내용

- 보안 약점

1. 불충분한 세션 관리

- **불충분한 세션 관리는 일정한 규칙이 존재하는 세션ID가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점**
- 세션 관리가 충분하지 않으면 침입자는 **세션 하이재킹**과 같은 공격을 통해 획득한 세션ID로 인가되지 않는 시스템의 기능을 이용하거나 중요 정보에 접근할 수 있다.

2. 잘못된 세션에 의한 정보 노출

- **잘못된 세션에 의한 정보 노출은 다중 스레드 Multi-Thread 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점**이다.
- **싱글톤 패턴**에서 발생하는 **레이스컨디션 Race Condition**으로 인해 동기화 오류가 발생하거나 **멤버 변수**의 정보가 노출될 수 있다.
- **멤버 변수보다 지역변수를 활용하여 변수의 범위를 제한함**으로써 **방지**할 수 있다.

세션 설계 시 **고려사항**

- **시스템의 모든 페이지에서 로그아웃이 가능하도록 UI를 구성**
- **로그아웃 요청 시 할당된 세션이 완전히 제거되도록 한다.**
- **세션 타임아웃은 중요도가 높으면 2~5분, 낮으면 15~30분으로 설정**
- **이전 세션이 종료되지 않으면 새 세션이 생성되지 못하도록 설계**
- **중복 로그인을 허용하지 않는 경우 클라이언트의 중복 접근에 대한 세션 관리 정책을 수립**

하향식 비용 선정 방법

- 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.
- 종류

전문가 감정 기법	조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법
델파이 기법	전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법

상향식 비용 선정 방법

- 프로젝트의 세부적인 작업 단위별로 비용을 산정한 후 집계하여 전체 비용을 산정하는 방법이다.
- 종류

LOC(원시 코드 라인 수, source Line Of Code) 기법	• 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법 • 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) 예 LOC 기법에 의하여 예측된 총 라인 수가 40,000라인, 개발에 참여할 프로그래머가 10명, 프로그래머들의 평균 생산성이 월간 400라인일 때 개발에 소요되는 기간은? • 노력(인월) = LOC/1인당 월평균 생산 코드 라인 수 = 40000/400 = 100명 • 개발 기간 = 노력(인월)/투입 인원 = 100/10 = 10개월
개발 단계별 인월수(Effort Per Task) 기법	LOC 기법을 보완하기 위한 기법으로, 각 기능을 구현시키는 데 필요한 노력을 생명 주기의 각 단계별로 산정

서비스 거부 공격 유형

Ping of Death (죽음의 핑)	Ping 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위(65,536 바이트) 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법
Smurfing(스머핑)	IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법
SYN Flooding	TCP(Transmission Control Protocol)는 신뢰성 있는 전송을 위해 3-way-handshake를 거친 후에 데이터를 전송하게 되는데, SYN Flooding은 공격자가 가상의 클라이언트로 위장하여 3-way-handshake 과정을 의도적으로 중단시킴으로써 공격 대상인 서버가 대기 상태에 놓여 정상적인 서비스를 수행하지 못하게 하는 공격 방법
TearDrop	데이터의 송 · 수신 과정에서 패킷의 크기가 커 여러 개로 분할되어 전송될 때 분할 순서를 알 수 있도록 Fragment Offset 값을 함께 전송하는데, TearDrop은 이 Offset 값을 변경시켜 수신 측에서 패킷을 재조립할 때 오류로 인한 과부하를 발생시킴으로써 시스템이 다운되도록 하는 공격 방법
Land	패킷을 전송할 때 송신 IP 주소와 수신 IP 주소를 모두 공격 대상의 IP 주소로 하여 공격 대상에게 전송하는 것으로, 이 패킷을 받은 공격 대상은 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 되는데, 이러한 패킷이 계속해서 전송될 경우 자신에 대해 무한히 응답하게 하는 공격
DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격	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 이용함

네트워크 침해 공격관련 용어

스미싱 (Smishing)	각종 행사 안내, 경품 안내 등의 문자 메시지(SMS)를 이용해 사용자의 개인 신용정보를 빼내는 수법
스피어 피싱 (Spear Phishing)	사회 공학의 한 기법으로, 특정 대상을 선정한 후 그 대상에게 일반적인 이메일로 위장한 메일을 지속적으로 발송하여, 발송 메일의 본문 링크나 첨부된 파일을 클릭하도록 유도해 사용자의 개인 정보를 탈취 함
APT(Advanced Persistent Threats, 지능형 지속 위협)	다양한 IT 기술과 방식들을 이용해 조직적으로 특정 기업이나 조직 네트워크에 침투해 활동 거점을 마련한 뒤 때를 기다리면서 보안을 무력화시키고 정보를 수집한 다음 외부로 빼돌리는 형태의 공격
무작위 대입 공격 (Brute Force Attack)	암호화된 문서의 암호키를 찾아내기 위해 적용 가능한 모든 값을 대입하여 공격하는 방식
큐싱(Qshing)	QR코드(Quick Response Code)를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법의 하나로, QR코드와 개인정보 및 금융정보를 낚는다(Fishing)는 의미의 합성 신조어
SQL 삽입(Injection) 공격	전문 스캐너 프로그램 혹은 봇넷 등을 이용해 웹사이트를 무차별적으로 공격하는 과정에서 취약한 사이트가 발견되면 데이터베이스 등의 데이터를 조작하는 일련의 공격 방식
크로스사이트 스크립팅(XSS)	웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 등 스크립트의 취약점을 악용한 해킹 기법

보안 기능의 보안 약점

적절한 인증 없이 중요기능 허용	**보안검사를 우회하여 인증과정 없이 중요한 정보 도는 기능에 접근 및 변경이 가능 중요정보나 기능을 수행하는 페이지에서는 재인증 기능을 수행하도록 하여 방지**
부적절한 인가	**접근제어 기능이 없는 실행경로를 통해 정보 또는 권한을 탈취 가능 모든 실행경로에 대해 접근제어검사를 수행하고 사용자에게 반드시 필요한 접근권한만을 부여하여 방지**
중요한 자원에 대한 잘못된 권한 설정	**권한설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용 가능 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않는 사용자의 중요 자원에 대한 접근여부를 검사함으로써 방지**
취약한 암호화 알고리즘 사용	**암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요정보를 탈취 가능 안전한 암호화 알고리즘을 이용하고 업무관련 내용이나 개인정보 등에 대해서는 IT보안인증 사무국이 안정성을 확인한 암호모듈을 이용함으로써 방지**
중요정보 평문 저장 및 전송	**암호화되지 않는 평문 데이터를 탈취하여 중요한 정보를 획득 가능 중요한 정보를 저장하거나 전송할 때 반드시 암호화 과정을 거치도록 하고 HTTPS 또는 SSL과 같은 보안 채널을 이용함으로써 방지**
하드코드된 비밀번호	**소스코드 유출 시 내부에 하드코드된 패스워드를 이용하여 관리자 권한을 탈취 가능 패스워드는 암호화하여 별도의 파일에 저장하고, 디폴트 패스워드나 디폴트 키 사용을 피함으로써 방지**

정보보안 침해 공격관련 용어

좀비(Zombie) PC	악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터로, C&C(Command & Control) 서버의 제어를 받아 주로 DDoS 공격 등에 이용됨
C&C 서버	해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버를 말함
봇넷(Botnet)	악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태를 말함
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 형태임
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, 로그 분석, SetUID 파일 검사
트로이 목마 (Trojan Horse)	정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없음

입력 데이터 검증 및 표현의 보안 약점

SQL삽입	**입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작하는 보안 약점** **동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지**
경로조작 및 자원삽입	**데이터 입출력 경로를 조작하여 서버자원을 수정•삭제할 수 있는 보안 약점** **방지:** **사용자 입력값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지**
크로스 사이트 스크립팅 XSS	**웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점** **HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 특수문자를 다른문자로 치환함으로써 방지**
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 검증없이 내부 명령어로 사용하지 않음으로써 방지**
위험한 형식 파일 업로드	**악의적인 명령어가 포함된 스크립트 파일을 업로드함으로써 시스템에 손상을 주거나, 시스템 제어가능한 보안 약점** **업로드 파일의 확장자 제한, 파일명의 암호화, 실행 속성을 제거하는 등의 방법으로 방지**
신뢰하지 않는 URL주소로 자동접속 연결	**입력값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점** **연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지**

시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점에는 **TOCTOU 경쟁조건, 종료되지 않는 반복문•재귀함수** 등이 있다.

TOCTOU 경쟁조건 = 검사 시점과 사용 시점을 고려하지 않고 코딩한 경우 발생하는 보안 약점

- 프로세스가 가진 자원정보와 실제 자원상태가 일치하지 않는 동기화 오류, 교착상태 등이 발생
- 방지: 코드내에 동기화 구문을 사용하여 해당 자원에는 한번에 하나의 프로세스만 접근가능 하도록 구성

종료되지 않는 반복문•재귀함수(서비스 또는 시스템이 자원고갈로 인해 정지하거나 종료될 수 있음**) 방지하는 법**

- 모든 반복문이나 재귀함수의 수행 횟수를 제한하는 설정을 추가
- 종료조건을 점검하여 반복 또는 호출의 종료 여부를 확인

에러처리

소프트웨어 실행 중 발생할 수 있는 오류들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목들

문제점

- **예외처리 구문으로 처리하지 못한 오류들은 중요정보를 노출시킴**
- **소프트웨어의 실행이 중단됨**

보안 약점

- **오류메시지를 통한 정보노출**
- **오류상황 대응 부재**
- **부적절한 예외처리**

오류 메시지를 통한 정보 노출

오류발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점

스택 트레이스 Stack Trace

오류가 발생한 위치를 추적하기 위해 소프트웨어가 실행 중에 호출한 메소드의 리스트를 기록한 것

방지하는 법

오류 발생시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 준비된 메시지만 출력되도록 함

오류 상황 대응 부재

소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점

방지하는 법

오류가 발생할 수 있는 부분에 예외처리 구문을 작성하고 제어문을 활용하여 오류가 악용되지 않도록 코딩

부적절한 예외처리

함수의 반환값 또는 오류들을 세분화하여 처리하지 않고 광범위하게 묶어 한 번에 처리하거나, 누락된 예외가 존재할 때 발생하는 보안 약점

방지하는 법

모든 함수의 반환값이 의도대로 출력되는지 확인하고, 세분화된 예외처리를 수행

널 포인터 Null Pointer 역참조

널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점

- 방지: 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사

부적절한 자원 해제

자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점

- 방지: 프로그램 내 자원 반환 코드가 누락되었는지 확인하고, 오류로 인해 함수가 중간에 종료되었을 때 예외처리에 관계없이 자원이 반환되도록 코딩

해제된 자원 사용

이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점

- 반환된 메모리를 참조하는 경우 발생
- 방지: 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화

초기화되지 않는 변수 사용

변수 선언 후 값이 부여되지 않는 변수를 사용할 때 발생하는 보안 약점

- 방지: 변수 선언 시 할당된 메모리를 초기화

캡슐화로 인해 발생할 수 있는 보안 약점

- 잘못된 세션에 의한 정보 노출
- 제거하지 않고 남은 디버그 코드
- 시스템 데이터 정보 노출
- Public 메소드로부터 반환된 Private 배열
- Private 배열에 Public 데이터 할당

Public 메소드로부터 반환된 Private 배열

선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점

- 문제점: 배열의 주소가 외부로 공개되어 외부에서 접근 가능
- 방지: Private 배열을 별도의 메소드를 통해 조작하거나,

  동일한 형태의 복제본으로 반환받은 경우 값을 전달하는 방식

Private 배열에 Public 데이터 할당

- 문제점: Private 배열을 외부에서 접근 가능
- 방지: Public 으로 선언된 데이터를 Private 배열에 저장할 때, 레퍼런스가 아닌 값을 직접 저장

API 오용으로 인해 발생할 수 있는 보안 약점

- DNS Loop 에 의존한 보안 결정
- 취약한 API 사용

DNS Loop 에 의존한 보안 결정

도메인 명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점

- 방지: 직접 IP 주소를 입력

취약한 API 사용

보안 문제로 사용이 금지된 API 사용하거나 잘못된 방식으로 API 사용할 경우 발생하는 보안약점

- 예

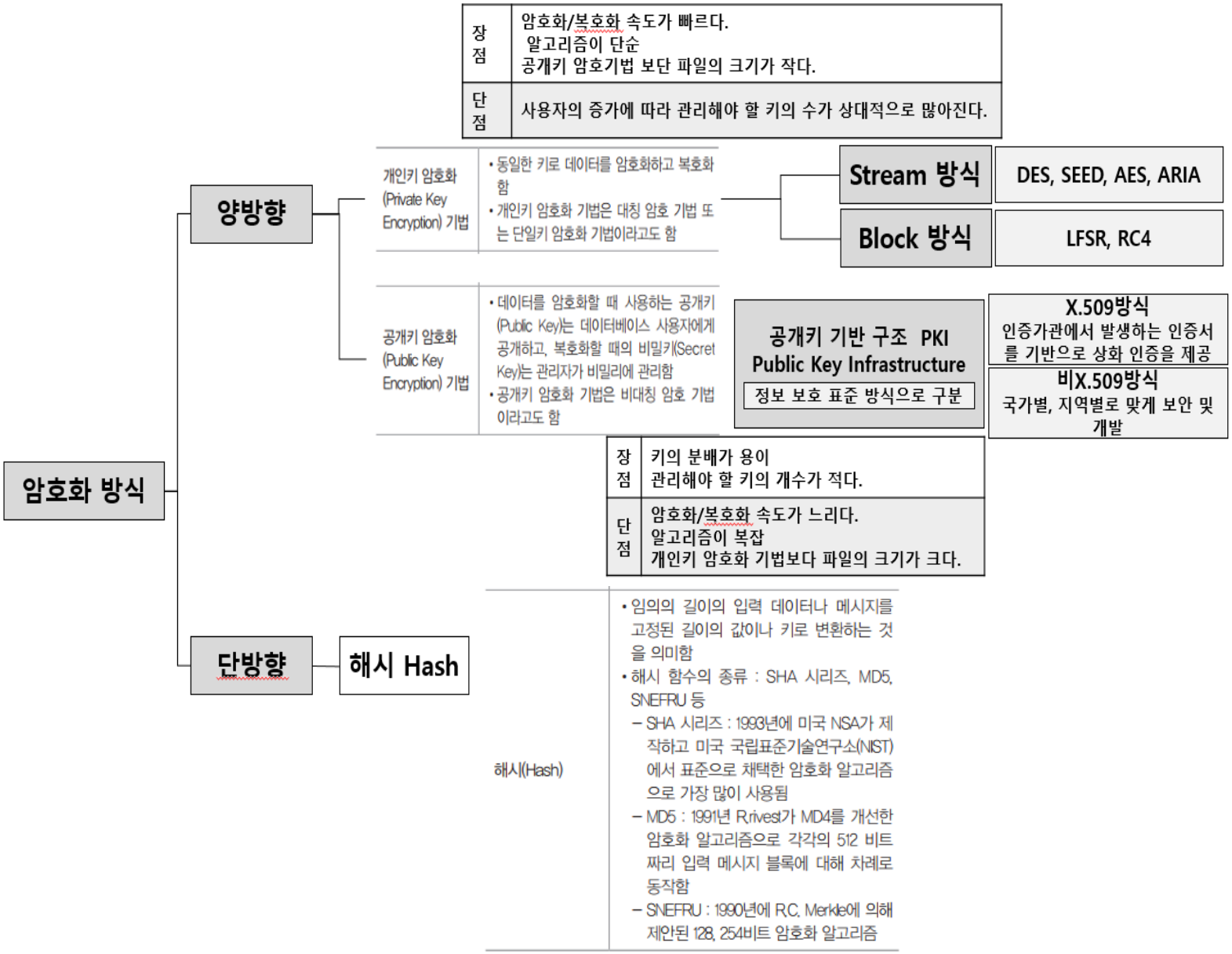
  C 언어의 문자열 함수: strcat(), strcpy(), sprintf()

  보안상 안전한 API 라도 자원에 대한 직접 연결이나 네트워크 소켓을 동한 직접 호출이 있는 경우

  보안에 위협을 줄 수 있는 인터페이스를 사용하는 경우
- 방지: 보안상 금지된 함수는 안전한 함수로 대체, 보안이 보장되는 인터페이스 사용

암호 알고리즘

패스워드, 주민번호, 은행계좌 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법



주요 암호화 알고리즘

SEED	• 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류
ARIA(Academy, Research Institute, Agency)	• 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류
DES(Data Encryption Standard)	• 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘 • 블록 크기는 64비트이며, 키 길이는 56비트
AES(Advanced Encryption Standard)	• 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘 • 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류
RSA(Rivest Shamir Adleman)	• 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘 • 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호화 알고리즘