소프트웨어 개발 보안 구축

SW 개발 보안

- SW 개발 보안은 해킹 등 사이버 공격의 원인인 보안 취약점을 SW개발 단계에서 미리 제거하고 SW 개발에 따른 생명주기(Software Development Life Cycle – SDLC)별 로 단계적으로 수행하고 개발 과정에서 보안 업무를 수행하며 안전한 보안 요소를 만족하는 소프트웨어를 개발 및 운영하기 위한 목적으로 수행하는 개발 방법
- 보안 관련 기관
  - ✓ 행정 안전부
  - ✓ KISA(한국인터넷진흥원)
  - ✓ 발주기관
  - ✓ 사업자
  - ✓ 감리법인
- 소프트웨어 개발 직무 별 보안 활동
  - ✓ Project Manager: 조직 구성원들에게 응용 프로그램 보안 영향을 이해시킴
  - ✓ Requirement Specifier: Architect가 고려해야 할 보안 관련 비지니스 요구사항을 설명
  - ✓ Architect: 보안 오류가 발생하지 않도록 보안 기술 문제를 충분히 이해
  - ✓ Designer: 특정 기술에 대해 보안 요구사항의 만족성 여부를 확인
  - ✓ Implementer: 구조화 된 소프트웨어 개발 환경에서 프로그램을 원할히 구현할 수 있도록 시큐어 코딩 표준을 준수하여 개발
  - ✓ Test Analyst: 소프트웨어 개발 요구 사항과 구현 결과를 반복적으로 확인
  - ✓ Security Auditor: 소프트웨어 개발 프로젝트의 현재 상태의 보안을 보장

SW 개발 보안

- Secure SDLC 방법론
  - ✓ CLASP: SDLC의 초기 단계에서 보안을 강화하기 위해 개발된 방법론
  - ✓ SDL: Microsoft에서 안전한 소프트웨어 개발을 위해 기존의 SDLC를 개선한 방법론
  - ✓ Seven Touchpoints: 소프트웨어 보안의 모범 사례를 SDLC에 통합한 방법론
- SDLC 단계별 보안 활동
  - ✓ 요구사항 분석 단계: 보안 항목에 해당하는 요구 사항을 식별하는 작업을 수행
  - ✓ 설계 단계: 식별된 보안 요구사항들을 소프트웨어 설계서에 반영하고 보안 설계서를 작성함
  - ✓ 구현 단계: 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며 설계서에 따라 보안 요구 사항들을 구현
  - ✓ 테스트 단계: 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작하는지 점검
  - ✓ 유지보수 단계: 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안 사고들을 식별하고 사고 발생 시 이를 해결하고 보안 패치를 실시

SW 개발 보안

- 사회 공학(Social Engineering): 컴퓨터 보안에 있어서 인간 상호 작용의 깊은 신뢰를 바탕으로 사람들을 속여 정상 보안 절차를 깨트리기 위한 비 기술적 시스템 침입 수단
- 보안 아키텍쳐 – ITU-T X.805
  - 보안 아키텍처는 정보 시스템의 무결성(Integrity), 가용성(Availability), 기밀성(Confidentiality)을 확보하기 위해 보안 요소 및 보안 체계를 식별하고 이들 간의 관계를 정의한 구조
  - 보안 아키텍처를 보안 계층(Security Layers), 보안 영역(Security Areas), 보안 요소(Security Elements)의 3개 레이어로 구분하여 설명
- SW 개발 보안의 요소
  - ✓ 기밀성(confidentiality): 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용
  - ✓ 무결성(integrity): 시스템 내의 정보는 오직 인가된 사용자만 수정
  - ✓ 가용성(availability): 인가 받은 사용자는 언제라도 사용 가능
  - ✓ 인증(authentication): 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위
  - ✓ 부인 방지(Non-repudiation): 데이터를 송·수신한 자가 송·수신 사실을 부인할 수 없도록 송·수신 증거를 제공

Secure Coding

- 안전한 소프트웨어를 개발하여 각종 보안 위협으로부터 예방하고 대응하고자 하며 정보 시스템 개발 시 보안성을 고려하고 보안 취약점을 사전에 제거하기 위하여 시큐어 코딩을 사용
- 소프트웨어 개발 보안 측면의 시큐어 코딩의 목적
  - ✓ 보안 취약점과 결함 방지: 최근 사이버 공격의 진화에 따라 사전에 정보처리시스템의 보안취약점을 사전에 대응하고 SQL injection 취약점, Zero Day Attack 공격, 침입 차단 시스템(TMS System) 등 보안 장비의 우회 등과 같은 보안 취약점을 사전에 제거하여 개발
  - ✓ 안전한 대 고객 서비스 확대: 대부분의 대 고객 서비스가 ICT신기술을 통하여 인터넷을 통해 제공되면서 대 고객 서 비스의 보안취약점을 지속적으로 진단하여 제거에 효율적 관리 방안을 마련
  - ✓ 안정성 및 신뢰성 확보: 대 고객 서비스의 신뢰성을 기반으로 하는 안정성에 기반한 보안 확보를 위해 정보 시스템의 기초 단계부터 설계 개념 및 시큐어 코드의 수준에서의 대응조치를 제안하여 대 고객 서비스의 보안성을 강화
- 보안 프레임워크
  - ✓ 보안 프레임워크(Security Framework)는 안전한 정보 시스템 환경을 유지하고 보안 수준을 향상시키기 위한 체계
  - ✓ ISO 27001은 정보보안 관리를 위한 국제 표준으로, 일종의 보안 인증이자 가장 대표적인 보안 프레임워크로 영국의 BSI(British Standards Institute)가 제정한 BS 7799를 기반으로 구성되어 있으며 조직에 대한 정보보안 관리 규격이 정의되어 있어 실제 심사/인증용으로 사용됨

세션 통제

- 세션은 서버와 클라이언트의 연결을 의미하고 세션 통제는 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것
- 세션 통제는 소프트웨어 개발 과정 중 요구 사항 분석 및 설계 단계에서 진단해야 하는 보안 점검 내용
- 세션 통제의 보안 약점
  - ✓ 불충분한 세션 관리
    - ❑ 일정한 규칙이 존재하는 세션ID가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점
    - ❑ 세션 ID(Session ID): 서버가 클라이언트들 을 구분하기 위해 부여하는 키(Key)로 클라이언트가 서버에 요청을 보낼 때마다 세션 ID를 통해 인증이 수행됨
  - ✓ 잘못된 세션에 의한 정보 노출
    - ❑ 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점
    - ❑ 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.

세션 통제

- 세션 설계 시 고려사항
  - ✓ 시스템의 모든 페이지에서 로그아웃이 가능하도록 UI 구성
  - ✓ 로그아웃 요청 시 할당된 세션이 완전히 제거되도록 함
  - ✓ 세션 타임아웃은 중요도 높을 때는 2~5분, 낮을 때는 15~30분
  - ✓ 이전 세션이 종료되지 않으면 새 세션이 생성되지 못하도록 설계
  - ✓ 중복 로그인을 허용하지 않는 경우 클라이언트의 중복 접근에 대한 세션 관리 정책을 수립
  - ✓ 패스워드 변경 시 활성화된 세션을 삭제하고 재할당
- 세션ID의 관리방법
  - ✓ 안전한 서버에서 최소 128비트 길이로 생성
  - ✓ 예측이 불가능하도록 안전한 난수 알고리즘 적용
  - ✓ URL Rewrite(URL에 세션 ID를 포함하는 것)기능을 사용하지 않는 방향으로 설계
  - ✓ 로그인시 로그인 이전의 세션 ID를 삭제하고 재할당
  - ✓ 장기간 접속하고 있는 세션ID는 주기적으로 재할당되도록 설계

입력 데이터 검증 및 표현

- 입력 데이터 검증 및 표현은 입력 데이터로 인해 발생 하는 문제들을 예방하기 위해 구현 단계에서 검증해야 하는 보안 점검 항목
  - ✓ SQL Injection
    - 응용 프로그램에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작하는 보안 약점
    - 관리자 인증을 우회하기도 함
    - 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지
  - ✓ 경로 조작 및 자원 삽입
    - 데이터 입출력 경로를 조작하여 서버 자원 을 수정·삭제할 수 있는 보안 약점
    - 사용자 입력 값을 식별자로 사용하는 경우 경로 순회 공격을 막는 필터를 사용하여 방지
  - ✓ 크로스사이트 스크립팅(XSS)
    - 웹 페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점
    - HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 ‘<’, ‘>’, ‘&’ 등의 문자를 다른 문자로 치환함으로써 방지

입력 데이터 검증 및 표현

- ✓ 운영체제 명령어 삽입
  - 외부 입력 값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점
  - 웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력 값을 검증없이 내부 명령어로 사용하지 않음으로써 방지
- ✓ 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 방지
- ✓ 신뢰되지 않는 URL 주소로 자동 접속 연결
  - 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점
  - 연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지
- ✓ 메모리 버퍼 오버플로우
  - 버퍼 오버런(buffer overrun)은 연속된 메모리 공간을 사용하는 프로그램에서 할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 해서 발생하는 보안 약점
  - 경계 검사로 버퍼 오버플로를 방지
- ✓ 감사 추적(Audit Trails)
  - 데이터 처리 과정에서의 오류나 외부의 불법적인 침입을 파악하기 위해 정보 시스템 내·외부의 모든 활동을 기록하고 분석하는 것

보안 기능

- ❖ 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목
- ❖ 적절한 인증없이 중요 기능 허용
  - ✓ 보안검사를 우회하여 인증과정 없이 중요한 정보 또는 기능에 접근 및 변경이 가능
  - ✓ 중요 정보나 기능을 수행하는 페이지에서는 재 인증 기능을 수행하도록 하여 방지
- ❖ 부적절한 인가
  - ✓ 접근제어 기능이 없는 실행 경로를 통해 정보 또는 권한을 탈취할 수 있음
  - ✓ 모든 실행 경로에 대해 접근제어 검사를 수행하고 사용자에게는 반드시 필요한 접근 권한만을 부여하여 방지
- ❖ 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 방지
- ✓ 취약한 암호화 알고리즘 사용
  - ✓ 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요 정보를 탈취할 수 있음
  - ✓ 안전한 암호화 알고리즘을 이용하고 업무 관련 내용이나 개인정보 등에 대해서는 IT 보안인증사무국이 안정성을 확인한 암호 모듈을 이용함으로써 방지

보안 기능

- 중요 정보 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 방지
- 하드 코드 된 비밀번호
  - ✓ 소스코드 유출 시 내부에 하드 코드 된 패스워드를 이용하여 관리자 권한을 탈취할 수 있음
  - ✓ 패스워드는 암호화하여 별도의 파일에 저장하고 디폴트 패스워드나 디폴트 키의 사용을 피함으로써 방지

시간 및 상태

- 동시 수행을 지원하는 병렬 처리 시스템이나 다수의 프로세스가 동작하는 환경에서 시간과 실행 상태를 관리하여 시스템이 원활하게 동작되도록 하기 위한 보안 검증 항목
- 시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점
  - ✓ TOCTOU 경쟁 조건
    - 검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 발생하는 보안 약점
    - 코드 내에 동기화 구문을 사용하여 해당 자원에는 한번에 하나의 프로세스만 접근 가능하도록 구성함으로써 방지
  - ✓ 종료되지 않는 반복문 또는 재귀 함수
    - 반복문이나 재귀 함수에서 종료 조건을 정의하지 않았거나 논리 구조상 종료될 수 없는 경우 발생하는 보안 약점
    - 모든 반복문이나 재귀 함수의 수행 횟수를 제한하는 설정을 추가하거나 종료 조건을 점검하여 반복 또는 호출의 종료 여부를 확인함으로써 방지

에러 처리

- 소프트웨어 실행 중 발생할 수 있는 오류(Error)들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목
- 에러 처리의 미비로 인한 코딩이 유발하는 보안 약점
  - ✓ 오류 메시지를 통한 정보노출
    - 오류 발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점
    - 예외처리 구문에 예외의 이름이나 스택 트레이스를 출력하도록 코딩한 경우 해커는 소프트웨어의 내부구조를 쉽게 파악
    - 오류 발생 시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 사전에 준비된 메시지만 출력되도록 함으로써 방지
  - ✓ 오류 상황 대응 부재
    - 소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점
    - 오류가 발생할 수 있는 부분에 예외처리 구문을 작성하고 제어문을 활용하여 오류가 악용되지 않도록 코딩함으로써 방지
  - ✓ 부적절한 예외처리
    - 함수의 반환값 또는 오류들을 세분화하여 처리하지 않고 광범위하게 묶어 한번에 처 리하거나, 누락된 예외가 존재할 때 발생하는 보안 약점
    - 모든 함수의 반환값이 의도대로 출력되는지 확인하고 세분화된 예외처리를 수행함으로써 방지

코드 오류

- 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형(Type) 변환, 자원 반환 등의 오류를 예방하기 위한 보안 점검 항목
- 코드 오류로 발생할 수 있는 보안 약점
  - ✓ 널 포인터(Null Pointer) 역 참조
    - 널 포인터가 가리키는 메모리에 어떠한 값을 사용할려고 할 때 발생하는 보안 약점
    - 널 포인터(Null Pointer): 널(Null)은 정보 부재를 의미하며 포인터(Pointer)는 메모리의 위치를 가리키는 요소이며 널 포인터(Null Pointer)는 포인터에 널(Null)이 저장되어 어떠한 곳도 가리키지 못하는 상태
    - 널(Null)이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지
  - ✓ 부적절한 자원 해제
    - 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점
    - 프로그램 내에 자원 반환 코드가 누락되었는지 확인하고 오류로 인해 함수가 중간에 종료되었을 때 예외 처리에 관계없이 자원이 반환되도록 코딩함으로써 방지
  - ✓ 해제된 자원 사용
    - 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점이다.
    - 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화해서 방지

코드 오류

- 코드 오류로 발생할 수 있는 보안 약점
  - ✓ 초기화되지 않은 변수 사용
    - 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점
    - 변수 선언 시 할당된 메모리를 초기화함으로써 방지
  - ✓ Stack Guard
    - 메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장해 두었다가 그 값이 변경되었을 경우 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술
    - 컴파일러 옵션으로 지정할 수 있는 방어 기법 중 하나로 스택에 프로그램 실행 시 canary 이놈을 심어두고 BOF 공격할 때 canary의 값을 확인

캡슐화

- 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목
- 캡슐화로 인해 발생할 수 있는 보안 약점
  - ✓ 잘못된 세션에 의한 정보 노출
    - Multi-Thread 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점
    - 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지
  - ✓ 제거되지 않고 남은 디버그 코드
    - 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점
    - 소프트웨어를 배포하기 전에 코드 검사를 통해 남아있는 디버그 코드를 삭제함으로써 방지
  - ✓ 시스템 데이터 정보 노출
    - 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩했을 때 발생하는 보안 약점
    - 시스템 메시지를 통해 노출되는 메시지는 최소한의 정보만을 제공함으로써 방지

캡슐화

- 캡슐화로 인해 발생할 수 있는 보안 약점
  - ✓ Public 메소드로부터 반환된 Private 배열
    - 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점
    - Private 배열을 별도의 메소드를 통해 조작하거나 동일한 형태의 복제본으로 반환받은 후 값을 전달하는 방식으로 방지
  - ✓ Private 배열에 Public 데이터 할당
    - Private 배열에 Public으로 선언된 데이터 또는 메소드의 파라미터를 저장할 때 발생하는 보안 약점
    - Public으로 선언된 데이터를 Private 배열에 저장할 때, 레퍼런스가 아닌 값을 직접 저장 함으로써 방지

API 오용

- 소프트웨어 구현 단계에서 API를 잘못 사용하거나 보안에 취약한 API를 사용하지 않도록 하기 위한 보안 검증 항목
- API 오용으로 발생할 수 있는 보안 약점
  - ✓ DNS Lookup에 의존한 보안 결정
    - 도메인 명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점
    - DNS 검색을 통해 도메인 이름을 비교하지 않고 IP 주소를 직접 입력하여 접근함으로써 방지
  - ✓ 취약한 API 사용
    - 보안 문제로 사용이 금지된 API를 사용하거나 잘못된 방식으로 API를 사용했을 때 발생하는 보안 약점
    - 보안 문제로 금지된 대표적인 API에는 C언어의 문자열 함수 strcat( ), strcpy( ), sprintf( ) 등
    - 보안 문제로 금지된 함수는 안전한 함수로 대체하고 API의 매뉴얼을 참고하여 보안이 보장되는 인터페이스를 사용함으로써 방지

암호화 알고리즘

- 패스워드, 주민번호, 은행계좌와 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미
- 암호화 알고리즘은 해시(Hash)를 사용하는 단방향 암호화 방식과 개인키 및 공개키로 분류되는 양방향 암호화 방식이 있음
  - ✓ 개인키 암호화(Private Key Encryption) 기법
    - 동일한 키로 데이터를 암호화하고 복호화
    - 대칭 암호화 기법 또는 단일키 암호화 기법이라고도 함
    - 암호화/복호화 속도가 빠르지만 관리해야 할 키의 수가 많음
    - 종류
      - ❑ 스트림 암호화 방식: 평문과 동일한 길이의 스트림을 생성하며 비트 단위로 암호화 하는 방식으로 LFSR, RC4가 대표적인 알고리즘
      - ❑ 블록 암호화 방식: 한번에 하나의 데이터 블록을 암호화 하는 방식으로 DES, SEED, AES, ARIA 가 대표적인 알고리즘
  - ✓ 공개키 암호화(Public Key Encryption) 기법
    - 데이터를 암호화 할 때 사용하는 공개키(Public Key)는 사용자에게 공개하고 복호화 할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리
    - 비대칭 암호 기법 이라고도 함
    - 관리해야 할 키의 개수가 적지만 암호화/복호화 속도가 느림
    - RSA 알고리즘이 대표적

암호화 알고리즘

- 암호화 알고리즘
  - ✓ 양방향 알고리즘 – 개인키 암호화 와 공개키 암호화
    - SEED
      - 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘
      - 블록 크기는 128비트이며 키 길이에 따라 128, 256 으로 분류
    - ARIA(Academy, Research Institute, Agency)
      - 2004년 국가정보원과 산학연협회가 개발 한 블록 암호화 알고리즘
      - 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256 으로 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, 256 으로 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

암호화 알고리즘

- ❖ 암호화 알고리즘
  - ✓ 해시(Hash)
    - ● 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미
    - ● 해시 함수의 종류 : SHA 시리즈, MD5, SNEFRU 등
      - ▪ SHA 시리즈: 1993년에 미국 NSA가 제작하고 미국 국립표준기술연구소(NIST)에서 표준으로 채택한 암호화 알고리즘으로 가장 많이 사용되고 있는데 SHA-0, SHA-1, SHA-2(SHA-224, SHA-256, SHA-384, SHA-512 등) 로 발전
      - ▪ MD5: 1991년 R.rivest가 MD4를 개선한 암호화 알고리즘으로 각각의 512 비트 짜리 입력 메시지 블록에 대해 128비트 키를 적용
      - ▪ N-NASH: 1989년 일본의 전신전화주식회사(NTT)에서 발표한 암호화 해시 함수로 블록의 크기와 키의 길이가 모두 128 bit
      - ▪ SNEFRU: 1990년에 R.C. Merkle에 의해 제안된 128, 254비트 암호화 알고리즘으로 32bit 프로세서에서 구현현을 용이하게 할 목적으로 개발

서비스 거부 공격

- DoS(Denial of Service): 시스템을 악의적으로 공격해 해당 시스템의 자원을 부족하게 하여 원래 의도된 용도로 사용하지 못하게 하는 공격
  - ✓ Ping of Death: ping 명령을 사용할 때 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 전송해서 공격 대상의 네트워크를 마비시키는 서비스 거부 공격으로 큰 패킷은 분할되어 전송되기 때문에 수신 측은 분할되어 전송된 패킷을 재조립하고 각각의 분할된 패킷에 ICMP Ping 메시지에 대한 응답을 처리하느라고 시스템이 다운됨
  - ✓ Smurf Attack(SMURFING): 인터넷 프로토콜의 브로드캐스트 나 인터넷 운용 측면을 이용하여 인터넷 망을 공격하는 행위로 스머프 프로그램은 다른 네트워크 주소(스머프 – 공격 대상)로부터 생성된 것처럼 보이는 네트워크 패킷을 만들어 네트워크 내의 모든 IP 주소들에 ICMP Ping Message를 전송하는 방식으로 엄청난 양의 데이터를 한 사이트에서 집중적으로 보낸 것 처럼 해서 수신된 곳에서 응답 메시지를 송신된 곳으로 집중적으로 전송하게 해서 네트워크 불능 상태를 만드는 것인데 이를 무력화시키는 방법 중의 하나는 각 네트워크 라우터에서 브로드캐스트 주소를 사용할 수 없도록 해서 방비하는 것
  - ✓ SYN Flooding Attack: TCP는 신뢰성 있는 전송을 위해서 3-way handshaking을 사용해서 동기화 한 후 데이터를 전송하는데 가상의 클라이언트가 이 과정을 의도적으로 중단시켜 서버가 무한 대기 상태에 빠지도록 하는 공격 방법으로 수신지의 SYN에 대한 대기 시간을 줄이거나 침입 차단 시스템을 이용해서 방비

서비스 거부 공격

- DoS(Denial of Service)
  - ✓ Tear Drop Attack: 공격 대상 컴퓨터에 헤더가 조작된 일련의 IP 패킷 조각들을 전송함으로써 컴퓨터의 OS를 다운시키는 공격으로 보통 큰 패킷을 분할해서 전송할 때 분할 순서를 조작해서 재조립 할 때 오류로 인한 과부하가 발생하도록 해서 서비스를 중단시키는 공격 방법으로 패킷의 순서가 다를 때 이 패킷을 폐기하도록 해서 방비
  - ✓ Local Area Network Denial Attack(LAND Attack): 공격자가 패킷의 출발지 주소나 포트를 임의로 변경하여 출발지와 목적지 주소(또는 포트)를 동일하게 함으로써 공격 대상 컴퓨터의 실행 속도를 느리게 하거나 마비시켜 서비스 거부 상태에 빠지도록 하는 공격방법으로 수신되는 패킷 중 출발지 주소와 목적지 주소가 동일한 패킷들을 차단함으로써 공격을 회피
  - ✓ DDoS(Distributed Denial of Service): 여러 곳의 분산된 공격 지점에서 하나의 서버에 대해 분산 DoS 공격을 수행하는 것
    - ❑ 분산 서비스 공격용 툴(Agent의 역할을 수행하도록 하는 프로그램으로 데몬이라고도 함)
      - Trinn00: 초창기 데몬으로 UDP Flooding 공격을 수행
      - TFN(Tribe Flood Network): UDP Flooding 뿐 만 아니라 TCP SYN Flood 공격, ICMP 응답 요청, 스머핑 공격 등을 수행함
      - TFN2K: TFN의 확장판
      - Stacheldraht: 공격자, 마스터, 에이전트가 쉽게 노출되지 않도록 암호화된 통신을 수행하고 자동으로 업데이트되는 툴

네트워크 침해 공격

- 네트워크 침해 공격 유형
  - ✓ Smishing: 각종 행사 안내, 경품 안내 등의 문자 메시지(SMS) 등을 이용해 사용자의 개인 신용 정보를 빼내는 공격 기법으로 소액 결제를 하게 만들거나 앱을 설치하도록 해서 사용자 정보를 빼내가는 방식
  - ✓ 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 탈취하는 공격 기법
  - ✓ 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
    - ❑ 악성코드가 포함된 이메일을 지속적으로 발송해서 클릭하도록 하는 형태
    - ❑ Stuxnet 과 같이 악성코드가 담긴 이동식 디스크 등으로 전파하는 형태
    - ❑ 악성코드에 감염된 P2P 사이트에 접속하면 악성 코드에 감염되도록 하는 형태
  - ✓ 무작위 대입 공격 (Brute Force Attack): 암호화된 문서의 암호키를 찾아내기 위해 적용 가능한 모든 값을 대입하여 공격하는 방식
  - ✓ 큐싱(Qshing): QR Code(Quick Response Code) 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융 사기 기법의 하나로 QR 코드와 개인정보 및 금융정보를 '낚는다 (Fishing)'는 의미의 합성어

네트워크 침해 공격

- 네트워크 침해 공격 유형
  - ✓ SQL 삽입(Injection) 공격: 입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작하는 공격 기법으로 전문 스캐너 프로그램 혹은 봇넷 등을 이용해 웹사이트를 무차별적으로 공격하는 과정에서 취약한 사이트가 발견되면 데이터베이스 등의 데이터를 조작하는 방법
  - ✓ XSS(Cross Site Scripting): 웹 페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 공격 기법으로 사용자가 특정 게시물이나 이메일의 링크를 클릭하면 악성 스크립트가 실행되어 페이지가 깨지거나 사용자의 컴퓨터에 있는 로그인 정보나 개인 정보, 내부 자료 등이 해커에게 전달됨
  - ✓ Sniffing: 네트워크의 중간에서 남의 패킷 정보를 도청하는 해킹 유형의 하나로 수동적 공격에 해당함

정보 보안 침해

- Back Door
  - ✓ 시스템 설계자나 관리자에 의해 고의로 남겨진 시스템의 보안 헛점으로 응용 프로그램이나 운영체제에 삽입된 프로그램 코드
  - ✓ 시스템 접근에 대한 사용자 인증 등 정상적인 절차를 거치지 않고 응용 프로그램 또는 시스템에 접근 할 수 있도록 함
  - ✓ 경우에 따라서는 현장 서비스 기술자나 시스템 공급자의 유지보수 프로그래머가 사용할 목적으로 특수 계정을 허용하는 코드를 운영체제나 응용프로그램에 넣을 수도 있음
  - ✓ 멀웨어(Malware) – 바이러스/백도어/스파이웨어/트로이 목마
    - 악의적인 사용자가 시스템의 보안 허점을 응용하여 고의로 만들어진 프로그램
    - 목적은 시스템에 대한 사용자 인증 등 정상적은 절차를 거치지 않고 응용프로그램 또는 시스템에 접근할 수 있는 프로그램
- C&C 서버: 해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버
- 좀비(Zombie) PC: 악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터로 C&C(Command & Control) 서버의 제어를 받아 주로 DDoS 공격 등에 이용
- 봇넷(Botnet): 악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태
- Worm: 네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 있음
- 트로이 목마(Trojan Horse): 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로 그램이 동작할 때 활성화되어 부작용을 일으키는 것

정보 보안 침해

- Ransomware
  - ✓ 사용자의 동의 없이 컴퓨터에 설치되어 컴퓨터 시스템을 감염시켜 접근을 제한하고 내부 파일을 인질로 잡아 일종의 몸값을 요구하는 악성 소프트웨어의 한 종류
  - ✓ 컴퓨터로의 접근이 제한되기 때문에 제한을 없애려면 해당 악성 프로그램을 개발한 자에게 지불을 강요받게 되는데 이때 암호화된 랜섬웨어가 있는 반면, 어떤 것은 시스템을 단순하게 잠그고 컴퓨터 사용자가 지불하게 만들기 위해 안내문구를 출력
  - ✓ 최근 전 세계적인 랜섬웨어를 통한 대량 해킹은 인터넷 세계의 싸이버 아마겟돈으로 불림
  - ✓ 랜섬웨어는 몸값을 뜻하는 Ransom과 Software(소프트웨어)가 더해진 합성어
  - ✓ 일반적으로 윈도우 운영체제가 설치된 PC에서 가장 많이 발생하지만 모바일 환경에서도 발생하며, 맥 OS도 감염될 수 있음
- 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 의미
- 키로거 공격(Key Logger Attack): 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격

보안 통신 규약

- IPsec
  - ✓ IP 계층에서 통신 세션의 각 IP 패킷을 암호화하고 인증하는 안전한 IP 통신을 위한 프로토콜로 통신 세션의 개별 IP 패킷을 인증하고 암호화함으로써 처리
  - ✓ IPsec은 세션의 시작에서 에이전트들 사이에서 상호 인증을 확립하거나 세션을 맺는 중에 사용될 암호화 키의 협상을 위한 프로토콜을 포함
- SSL
  - ✓ 전송 계층 보안(Transport Layer Security, TLS, 과거 명칭: 보안 소켓 레이어/Secure Sockets Layer, SSL)은 컴퓨터 네트워크에 통신 보안을 제공하기 위해 설계된 암호 규약
  - ✓ '트랜스포트 레이어 보안'이라는 이름은 '보안 소켓 레이어'가 표준화 되면서 바뀐 이름
  - ✓ 인터넷 같이 TCP/IP 네트워크를 사용하는 통신에 적용되며, 통신 과정에서 전송 계층 종단 간 보안과 데이터 무결성을 확보
  - ✓ 웹 브라우징, 전자 메일, 인스턴트 메신저, voice-over-IP (VoIP) 같은 응용 부분에 적용
- S-HTTP
  - ✓ 웹 상에서 네트워크 트래픽을 암호화하는 주요 방법 중 하나
  - ✓ 웹 상에서 네트워크 트래픽(메시지)을 암호화하는 것으로 전자서명을 포함
- 데이터 익명화
  - ✓ 익명화란 개인 식별 정보를 삭제하거나 수정하는 데이터 처리 기술
  - ✓ 이 방법을 사용하면 어떤 개인과도 연관 지을 수 없는 익명화된 데이터를 얻을 수 있음

Server Authentication

- Authentication(인증)
  - ✓ 다중 사용자 컴퓨터 시스템이나 네트워크 시스템에서 로그인을 요청한 사용자의 정보를 확인하고 접근 권한을 검증하는 보안 절차
  - ✓ 인증에는 네트워크를 통해 컴퓨터에 접속하는 사용자의 등록 여부를 확인하는 것과 전송된 메시지의 위·변조 여부를 확인하는 것이 있음
  - ✓ 유형
    - 지식 기반 인증(Something You Know)
      - 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것
      - 관리 비용이 저렴하나 사용자가 인증 정보를 기억하지 못하면 본인이라도 인증을 하지 못함
      - 고정된 비밀번호, 패스 프라이즈, I-PIN 이 대표적
    - 소유 기반 인증 (Something You Have)
      - 사용자가 소유하고 있는 것을 기반으로 인증을 수행하는 것
      - 소유물이 쉽게 도용될 수 있으므로 지식 기반 인증과 함께 사용
      - 신분증, 메모리 카드, 스마트 카드, OTP(One Time Password) 등
    - 생체 기반 인증(Something You Are): 사용자의 고유한 생체 정보를 기반으로 인증을 수행하는 것으로 사용이 쉽고 도난의 위험도 적으며 위조가 어려운 방식으로 지문, 홍채/망막, 얼굴, 음성, 정맥 등
    - 행위 기반 인증 (Something You Do): 사용자의 행동 정보를 이용해 인증 수행하는 것으로 서명이나 동작을 이용
    - 위치 기반 인증 (Somewhere You Are): 인증을 시도하는 위치의 적절성을 통해 인증을 수행하는 것으로 GPS 나 IP 주소를 이용

Server Authentication

- Authorization: 리소스에 대한 접근 권한 및 정책을 지정하는 기능으로 컴퓨터 시스템은 어떤 (인증된) 리소스 수요자가 리소스에 대한 요청을 하면, 저장된 접근 제어 규칙들을 적용해 요청을 허가할지 거부할지를 결정하는데 이와 관련된 권한 처리
- 보안 서버
  - ✓ 개인 정보를 암호화하여 송수신 할 수 있는 기능을 갖춘 서버
  - ✓ 서버에 SSL(Secure Socket Layer) 인증서를 설치하여 전송 정보를 암호화 해서 송수신
  - ✓ 서버에 암호화 응용 프로그램을 설치하고 전송 정보를 암호화해서 송수신

보안 아키텍쳐

- 보안 아키텍쳐
  - ✓ 정보 시스템의 무결성, 가용성, 기밀성을 확보하기 위해 보안 요소 및 보안 체계를 식별하고 이들 간의 관계를 정의한 구조
  - ✓ 관리적, 물리적, 기술적 보안 개념의 수립, 보안 관리 능력의 향상, 일관된 보안 수준의 유지를 기재할 수 있음
  - ✓ 보안 수준에 변화가 생겨도 기존 보안 아키텍쳐의 수정없이 지원이 가능해야 하며 보안 요구사항의 변화나 추가를 수용할 수 있어야 함
- 보안 프레임워크
  - ✓ 안전한 정보 시스템 환경을 유지하고 보안 수준을 향상 시키기 위한 체계
  - ✓ ISO 27001
    - ❑ 정보 보안 관리를 위한 국체 표준으로 보안 인증 이자 가장 대표적인 보안 프레임워크
    - ❑ 영국의 BSI(British Standards Institute)가 제정한 BS7799를 기반으로 구성
    - ❑ 조직에 대한 정보 보안 관리 규격이 정의되어 있어 실제 심사/인증 용으로 사용

로그 분석

- Log
  - ✓ 시스템 사용에 대한 모든 내역을 기록해 놓은 것
  - ✓ 로그를 이용하면 시스템 침해 사고 발생 시 해킹 흔적이나 공격 기법을 파악할 수 있고 로그 정보를 정기적으로 분석하면 시스템에 대한 침입 흔적이나 취약점을 확인할 수 있음
- 리눅스 로그
  - ✓ 리눅스에서는 var/log 디렉토리에 여러가지 로그 파일을 생성
  - ✓ 로그 파일 종류
    - ❑ 커널 로그: 커널에 관련된 내용으로 관리자에게 알리기 위해 파일로 저장하지 않고 지정된 장치에 표시
    - ❑ 부팅 로그: 부팅 시 나타나는 메시지들을 기록
    - ❑ 크론 로그: 작업 스케줄러인 crond의 작업 내역을 기록
    - ❑ 시스템 로그: 커널에서 실시간으로 보내온 메시지를 기록
    - ❑ 보안 로그: 시스템의 접속에 대한 로그를 기록
    - ❑ FTP 로그: FTP로 접속하는 사용자에 대한 로그를 기록
    - ❑ 메일 로그: 송수신 메일에 대한 로그를 기록

로그 분석

- 윈도우 로그
  - ✓ 이벤트 로그 형식으로 시스템의 로그를 관리하고 이벤트 뷰어를 통해 확인 가능
  - ✓ 이벤트 뷰어의 로그 종류
    - ❑ 응용 프로그램
    - ❑ 보안
    - ❑ 시스템
    - ❑ Setup
    - ❑ Forwarded Events

보안 솔루션

- 보안 솔루션: 접근, 통제, 침입 차단 및 탐지 등을 수행하여 외부로부터 불법적인 침입을 막는 기술 및 시스템
- Fire Wall(방화벽)
  - ✓ 외부로부터 내부 망을 보호하기 위한 네트워크 구성요소 중의 하나로서 외부의 불법 침입으로부터 내부의 정보자산을 보호하고 외부로부터 불법정보 유입을 차단하기 위한 정책과 하드웨어 및 소프트웨어의 총칭
  - ✓ 필요성
    - 위협에 취약한 서비스에 대한 보호
    - 호스트 시스템에 대한 엑세스 제어
    - 보안의 집중
    - 확장된 프라이버시
    - 네트워크 사용에 대한 로깅과 통계자료
    - 네트워크 엑세스 정책 구현
  - ✓ Access Control List: 컴퓨터 파일 시스템과 관련된 액세스 제어 목록은 개체에 첨부 된 권한 목록으로 ACL은 오브젝트에 대한 액세스 권한이 부여 된 사용자 또는 시스템 프로세스와 지정된 오브젝트에 허용되는 조작을 지정
  - ✓ White List: 식별된 일부 실체들이 특정 권한, 서비스, 이동, 접근, 인식에 대해 명시적으로 허가하는 목록이며, 이에 대한 과정은 화이트 리스트라고 하며 반의어는 블랙리스트
  - ✓ 프록시 서버는 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해 주는 컴퓨터 시스템이나 응용 프로그램을 가리키는 것으로 서버와 클라이언트 사이에 중계기로서 대리로 통신을 수행하는 것을 가리켜 '프록시', 그 중계 기능을 하는 것을 프록시 서버라고 부름

보안 솔루션

- 침입 탐지 시스템(IDS; Intrusion Detection System): 해커 침입 패턴에 대한 추적과 유해 정보 감시를 위해 컴퓨터 시스템의 비 정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템으로, 오용 탐지(Misuse Detection), 이상 탐지(Anomaly Detection) 등의 기능을 수행하는 보안 솔루션
- 침입 방지 시스템(IPS; Intrusion Prevention System)
  - ✓ 방화벽과 침입 탐지 시스템을 결합한 것
  - ✓ 비정상적인 트래픽을 능동적으로 차단하고 격리하는 등의 방어 조치를 취하는 보안 솔루션
  - ✓ 침입 탐지 기능으로 패킷을 하나씩 검사한 후 비정상적인 패킷이 탐지되면 방화벽 기능으로 해당 패킷을 차단
- 데이터 유출 방지(DLP; Data Leakage/Loss Prevention)
  - ✓ 내부 정보의 외부 유출을 방지하는 보안 솔루션
  - ✓ 사내 직원이 사용하는 PC와 네트워크상의 모든 정보를 검색하고 메일, 메신저, 웹하드, 네트워크 프린터 등의 사용자 행위를 탐지·통제해 외부로의 유출을 사전에 막는 방식
- 웹 방화벽(Web Firewall): 일반 방화벽이 탐지하지 못하는 SQL 삽입 공격, XSS 등의 웹 기반 공격을 방어할 목적으로 만들어진 웹 서버에 특화된 방화벽으로 웹 관련 공격을 감시하고 공격이 웹 서버에 도달하기 전에 이를 차단해 주는 보안 솔루션

보안 솔루션

- VPN
  - ✓ 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션
  - ✓ 암호화된 규격을 통해 인터넷 망을 전용선의 사설망을 구축한 것처럼 이용하므로 비용 부담을 줄일 뿐만 아니라 원격지의 지사, 영업소, 이동 근무자가 지역적인 제한 없이 업무를 수행
- NAC(Network Access Control)
  - ✓ 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후 일관된 보안 관리 기능을 제 공하는 보안 솔루션
  - ✓ 내부 PC의 소프트웨어 사용 현황을 관리하여 불법적인 소프트웨어 설치를 방지
  - ✓ 일괄적인 배포 관리 기능을 이용해 백신이나 보안 패치 등의 설치 및 업그레이드를 수행
- ESM(Enterprise Security Management)
  - ✓ 다양한 장비에서 발생하는 로그 및 보안 이벤트를 통합하여 관리하는 보안 솔루션
  - ✓ 방화벽, IDS, IPS, 웹 방화벽, VPN 등에서 발생한 로그 및 보안 이벤트를 통합하여 관리함으로써 비용 및 자원을 절약
  - ✓ 보안 솔루션 간의 상호 연동을 통해 종합적인 보완 관리 체계를 수립

취약점 분석 및 평가

- 사이버 위협으로부터 정보 시스템의 취약점을 분석 및 평가 한 후 개선하는 일련의 과정
- 취약점 분석 및 평가 범위 및 항목
  - ✓ 범위는 정보 시스템과 정보 시스템 자산에 직간접적으로 관여된 물리적, 관리적, 기술적 분야
  - ✓ 평가의 기본 항목은 상, 중, 하 3단계로 중요도를 분리
  - ✓ 중요도가 상인 항목은 필수적으로 점검하고 하인 항목은 선택적으로 점검
- 취약점 분석 및 평가 수행 절차 및 방법
  - ✓ 취약점 분석 및 평가 계획 수립
  - ✓ 취약점 분석 및 평가 대상 선별
  - ✓ 취약점 분석 수행
  - ✓ 취약점 평가 수행

소프트웨어 개발 보안 활동 관련 법령

- 개인 정보 보호 관련 법령
  - ✓ 개인정보 보호법
  - ✓ 정보 통신망 이용촉진 및 정보보호 등에 관한 법률
  - ✓ 신용정보의 이용 및 보호에 관한 법률
  - ✓ 위치정보의 보호 및 이용 등에 관한 법률
  - ✓ 표준 개인정보 보호 지침
  - ✓ 개인정보의 안전성 확보 조치 기준
  - ✓ 개인정보 영향평가에 관한 고시
- IT 기술 관련 규정
  - ✓ RFID 프라이버시 보호 가이드라인
  - ✓ 위치정보의 보호 및 이용 등에 관한 법률
  - ✓ 위치정보의 관리적, 기술적 보호조치 권고 해설서
  - ✓ 바이오 정보 보호 가이드라인
  - ✓ 뉴미디어 서비스 개인정보 보호 가이드라인