

| 과정명 | |
|------|-------|
| 10차시 | 접근 제어 |

<1> 접근 제어의 개요

[1] 접근 제어 정의

- 접근 제어는 자원이 어떻게 접근되는지를 제어하여 인가되지 않은 수정이나 노출로부터 보호하는 것으로, 사용자가 컴퓨터 사용을 위해 사용자 이름과 패스워드를 입력하는 것도 접근 제어의 예시로 볼 수 있음

| | |
|----|---|
| 접근 | <ul style="list-style-type: none"> ◆ 주체와 객체 사이의 흐름 |
| 주체 | <ul style="list-style-type: none"> ◆ 사용자 또는 사용자가 수행하는 시스템 내에서 활성화된 요소 ◆ 연산, 프로그램, 프로세스 등 ◆ 작업 수행을 위해 객체 혹은 객체 안의 데이터에 대한 접근 요청 |
| 객체 | <ul style="list-style-type: none"> ◆ 수동적인 요소 ◆ 컴퓨터, 데이터베이스, 파일, 컴퓨터 프로그램 등 |

[2] 접근 제어 실행 과정 및 원칙

- 접근 제어 시 어떠한 원칙들을 어떻게 적용할 것인지를 보안 정책에 기술하는 것이 바람직함
- 접근 제어에서는 직무분리, 최소 권한, 알 필요성, 고장 시 안전 초기화, 완전한 중재의 다섯 가지 원칙을 준수함

(1) 직무 분리

- 중요한 정보나 민감한 정보를 다루는 기능을 분리하여 접근 권한의 남용을 막거나 최소화하는 것이 목표
- 예) 시스템 관리자가 자신이 관리하는 서버에서 발생하는 활동을 스스로 모니터링 하는 대신 객관적인 제3자에게 모니터링 하도록 하는 것

(2) 최소 권한

- 어떤 사람에게도 주어진 업무를 수행 하는데 필요 이상의 권한을 부여해서는 안됨
- 최소 권한 원칙 하에서는 기본 권한이 아닌 최소한의 권한이 주어져야 함
- 예) 보안팀이 내부 웹 사이트에 문서와 자료를 저장하지만 보안팀 구성원이라 해도 보안 사건 담당이 아닌 직원은 사건 처리 파일에 접근할 수 없음

(3) 알 필요성

- 주체가 일을 수행하기 위해 객체에 접근할 필요가 없다면, 주체는 객체에 접근 권한을 가져서는 안됨
- 예) 주체가 객체에 정보를 첨부하되, 객체 내의 정보를 수정할 필요가 없다면 쓰기 권한을 받아서는 안되고 첨부 권한만을 받아야 함

(4) 고장 시 안전 초기화

- 주체나 객체 생성 시 권한을 초기화하는 방법에 관한 것이자 주체가 주어진 일을 완료하지 못하고 실패하더라도 주체가 행한 변화를 초기 상태로 되돌림으로써 시스템을 안전한 상태로 유지해야한다는 것을 의미함

(5) 완전한 중재

- 주체의 객체에 대한 접근은 처음뿐만이 아닌 항상 검사되어야 함
- 많은 시스템에서 서비스 구현을 용이하게 하기 위해 캐싱 기법을 사용하여 위배되는 경향이 있는데, 캐싱은 시스템의 효율성을 향상시킬 수 있으나 보안성에 문제를 가져올 수 있음

- 접근 제어는 식별-인증-인가-책임추적성의 실행과정을 거침

(1) 식별

- 주체가 인증 서비스에 스스로를 확인시키기 위하여 자신의 신원 정보를 제공하는 활동
- 신원 정보는 물리적 시설 또는 컴퓨터 정보 시스템에 접근을 가능하게 하는 물리적인 객체, 지식 또는 사람이 가지고 있는 특성으로 개인 식별 번호, 소지한 물품, 신체 특성 또는 이들의 조합이 신원 정보가 될 수 있음
- 신원 정보의 예시 : 사용자명, 계정 번호, 메모리 카드, 생체 인식 정보 등

(2) 인증

- 주체가 자신의 신원을 증명하기 위해 행하는 검증 활동
- 대표적인 예시로는 사용자가 아이디와 패스워드를 가지고 인증하는 것

(3) 인가

- 인증된 주체가 자원에 접근하여 요청한 업무를 수행할 수 있는가를 판단하여 허용하는 것
- 주체가 가지고 있는 보안 수준 및 알 필요성 등 보안 정책을 참조하여 판단
- 접근 제어 목록, 보안 등급 등을 사용

(4) 책임추적성

- 시스템에 접근한 주체가 시스템에 어떤 행위를 하고 있는지를 기록함으로써 문제 발생 시 원인 및 책임 소재를 파악하기 위해 필요함
- 감사, 디지털 포렌식 등의 방법을 사용

<2> 접근 제어 모델 및 방식

[1] 접근 제어 모델

- 접근 제어 모델은 프레임워크로서 주체가 어떻게 객체에 접근하는지를 설명하며, 주요 접근 제어 모델로는 임의적, 강제적, 역할 기반 접근 제어 모델이 있음
- 각 접근 제어 모델들은 주로 운영 체제의 코어 또는 코널에서 구현되어 응용 프로그램을 제어하는데 사용되며 기업은 한 가지 모델만을 사용하기도 하고 여러 모델들을 결합하여 사용하기도 함

(1) 임의적 접근 제어

- 소유자의 임의성에 기반을 두기 때문에 소유자가 자신의 자원에 접근할 수 있는 주체 지정 가능
- 오늘날 대부분의 운영체제는 임의적 접근 제어를 기반으로 구현
- 전통적인 유닉스 시스템에서 사용하는 사용자/그룹 계정, 읽기-쓰기-실행-권한도 이에 해당

| | |
|----|---|
| 장점 | <ul style="list-style-type: none">◆ 객체별 세분화된 접근 제어◆ 특정 주체가 다른 주체에 대해 임의적으로 접근 제어◆ 매우 유연한 접근 제어 서비스 제공 가능 |
| 단점 | <ul style="list-style-type: none">◆ 시스템 전체 차원의 일관성 있는 접근 제어가 부족◆ 높은 권한을 가진 사용자가 다른 사용자에게 임의적으로 접근을 허용할 수 있음◆ 멀웨어, 바이러스, 웜, 루트킷, 트로이 목마 공격에 취약 |

(2) 강제적 접근 제어

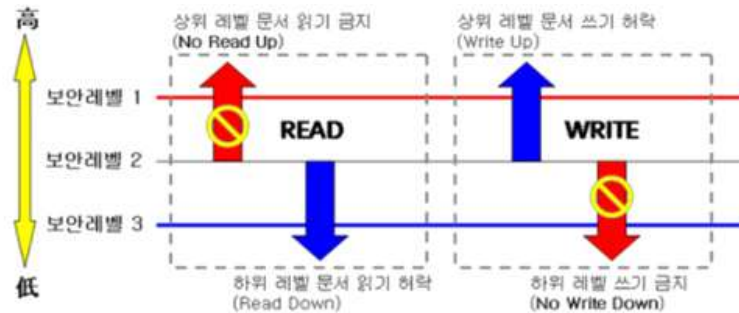
- 사용자와 데이터 소유자들에게는 해당 파일들에 접근할 수 있는 것을 결정하는데 자유로운 권한이 없으며, 관리자에 의해 시스템 전체적으로 적용되는 메커니즘에 따라 결정됨.
- 이러한 룰은 시스템의 모든 주체와 객체가 보안 레이블을 가져야 함이 전제가 됨

| | |
|----|--|
| 장점 | <ul style="list-style-type: none">◆ 중앙 집중형 보안 관리로 관리자 허용 개체만 접근할 수 있도록 강제로 통제하기 때문에 모든 객체에 대 한 관리가 용이하며 매우 엄격한 보안을 제공함 |
| 단점 | <ul style="list-style-type: none">◆ 매우 제한적인 사용자 기능◆ 많은 관리적 부담 요구◆ 많은 비용 소모◆ 모든 접근에 대한 확인으로 성능 저하◆ 상업적인 환경에 부적합 |

- 보안 레이블은 주체와 객체에 대한 접근 허용 여부를 결정짓는데 사용
- 보안 레이블에서 범주는 정보를 분류하는 기준으로 부서, 프로젝트, 관리 수준 등이 사용되며, 보통 알 필요성 규칙이 적용됨
- 보안 레이블에 할당된 비밀 취급 허가 수준과 알 필요성 수준을 객체의 보안 레이블에 할당된 분류 수준 및 범주와 비교하여 접근 허용 여부를 결정하게 됨
- 강제적 접근 제어 모델의 예시로는 벨-라파둘라 모델, 비바 모델, 클락 윌슨 모델, 만리장성 모델(브루어-나쉬 모델)이 있음

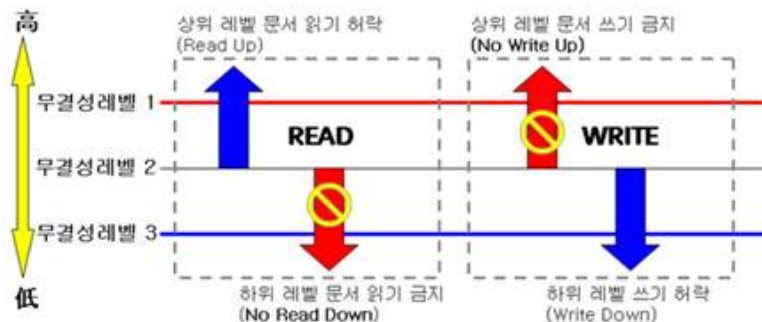
1) 벨-라파둘라 모델

- 1960년대, 미 육군에서 근무하던 벨-라파둘라가 메인 프레임 사용 환경에서 정보 유출 발생 차단을 위해 고안해 낸 강제적 접근 제어 모델
- 처음으로 제시된 수학적 보안 모델
- 높은 보안 수준에서 낮은 보안 수준으로 정보가 흐르는 것을 방지



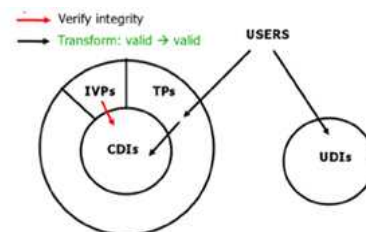
2) 비바 모델

- 1977년 비바가 개발한 데이터의 무결성을 위한 모델
- 데이터의 부적절한 변조 방지에만 목적을 두고 있음
- 보안 수준 대신 무결성 수준이라는 용어 사용



3) 클락 윌슨 모델

- 1978년 무결성 중심의 상업적 모델로 개발
- 무결성의 3가지 목표인 비인가자들의 데이터 변형 방지, 내/외부의 일관성 유지, 합법적인 사람에 의한 불법적인 수정 방지를 모두 만족한 모델



- CDI(Constrained Data Items) : 무결성이 극도로 요구되는 데이터
- UDI(Unconstrained Data Items) : 무결성이 그다지 중요하지 않은 데이터
- IVP(Integrity Verification Procedures) : 상주하면서 CDIs의 무결성을 체크
- TP(Transformation Procedures) : valid state → valid state

4) 만리장성 모델(브루어-나쉬 모델)

- 비즈니스 영역의 한 회사에 최근 일을 한 적이 있는 파트너는 동일한 영역에 있는 다른 회사의

자료에 접근해서는 안된다는 개념이 핵심인 접근 제어 모델

- 직무 분리를 접근 제어에 반영한 개념으로 상호 신뢰하지 않는 사용자들끼리 데이터를 이용하는 컴퓨터 시스템에서 데이터를 구분하는 방법을 제시함

(3) 역할 기반 접근 제어

- 기존 모델에서 '그룹'이라는 개념을 좀 더 정형화한 것
- 사용자(주체)에게 직접적으로 접근 권한을 할당하는 대신, 사용자들을 직무에 따라 역할로 그룹핑하고, 이 역할에 접근 권한을 부여함
- 임의적 접근 제어에 비해 유연성은 부족하나, 편리한 관리 능력을 제공하고 관리자에 의한 일관성 있는 접근 제어와 최소 권한, 직무분리 원칙을 충족시키기 용이함

[2] 참조 모니터

- 주체와 객체 사이의 모든 접근을 제어하는 추상적인 기계로 시스템의 보안 모델을 강제하는 역할
- 모든 주체의 객체로써 접근 요청은 참조 모니터에 의해 모니터링되며, 참조 모니터는 인가 데이터베이스를 참조하여 접근의 허용 여부를 결정함. 이러한 모든 과정은 기록되며 후에 감사 추적에 사용될 수 있음
- 참조 모니터는 개념을 수행하는 프로세스가 독립적이고 분리되어 부정하게 조작될 수 없어야 하며, 모든 접근 시도에 시행되어 회피하는 것이 불가능해야 하고 항상 모든 동작은 분석과 테스트를 통해 올바름을 확인할 수 있어야 한다는 요구사항을 만족해야 함



[3] 접근 제어 방식

- 접근 제어 방식은 중앙 집중화, 분산화, 혼합형으로 분류할 수 있음

(1) 중앙집중화 접근 제어

- 단일 개체가 전체 조직의 접근 제어를 수행하는 방식으로 AAA의 기능을 하나의 시스템에서 수행
- AAA 프로토콜의 예시로는 RADIUS, TACACS/TACACS+, Diameter 등이 있음

1) RADIUS

- 전화를 이용하여 원격 접속한 사용자를 인증하고 인가하는 프로토콜
- 클라이언트가 원격 사용자의 로그인 요청을 수신하여 RADIUS 서버로 전달하면 서버가 사용자 이름과 패스워드 값을 비교하여 인증
- 초기 RAS 접속제어를 표준화함

2) TACACS+

- 기본적으로는 RADIUS와 유사하나 전송 프로토콜로 TCP를 사용함
- 패킷 페이로드 전체를 암호화하여 RADIUS 프로토콜보다 안전성이 높음
- 기업 네트워크와 같이 보다 정교한 인증 단계와 복잡한 인가 활동을 위한 엄격한 통제가 필요한 곳에 적합함

3) Diameter

- AAA프로토콜의 기본만을 제공
- 다양한 서비스와의 연동을 위해 확장 가능한 프레임 워크 제공

(2) 분산화된 접근 제어

- 각 자원의 사용자가 자신의 자원에 대해서 또는 한 부서의 부서장이 자신 부서 안의 시스템에 대해 인증, 인가, 책임 추적을 직접 제어하는 방식
- 해당 조직의 특성에 맞춰 빠른 제어 가능
- 사용자의 접근과 권한을 위한 절차와 기준의 일관성 유지가 어려움
- 표준화가 부족하며 권한의 오버래핑, 보안의 구멍이 존재

(3) 혼합형 접근 제어

- 관리자가 중앙집중식으로 데이터베이스, 파일서버, 호스트 등의 정보 시스템에 대한 접근을 제어하면, 사용자는 분산 방식으로 자신 소유의 데이터에 대한 접근을 제어할 수 있는 방식

1) OTP

- 사용자가 사용할 때마다 매번 바뀌는 패스워드를 생성하는 장치
- 패스워드 재사용이 불가능하기 때문에 추측을 통한 해킹, 패킷 스피닝을 통한 재사용 공격, 다음 패스워드 예측이 불가능
- 사용자, OTP 단말기, 서비스 제공 서버, OTP 인증서버로 구성
- 시간 동기화 방식, 인증 횟수 기반 동기화 방식, 비동기화 방식이 존재

2) SSO

- 여러 개의 사이트에서 한 번의 로그인으로 여러 다른 사이트들을 자동으로 접속하는 방법
- 하나의 사용자 정보를 기반으로 여러 시스템에 하나의 통합 인증 사용
- 정당한 사용자 및 서버로 위장, 다양한 인증 정보의 노출, 인증 정보 재사용, 키 관리, 관리자/사용자 프로그램 지속적 연결 시 제3자 공격 노출 등의 보안 위협이 있음
- SSO 보안 기능 요구 사항으로는 상호 인증, 사용자 인증, 서버 인증, 데이터 보호, 검증필 암호 모듈 탑재, 올바른 알고리즘 운영, 안전한 키 관리, 안전한 세션 관리, 보안 검사 등이 있음

<3> 접근 제어 보안위협 및 대응책

[1] 패스워드 크래커

- 다양한 편법으로 패스워드를 풀어주는 프로그램
- 사전 공격, 무차별 공격, 패스워드 하이브리드 공격, 레인보우 테이블 공격 등이 있음

| | |
|---------------|---|
| 사전 공격 | <ul style="list-style-type: none"> ◆ 패스워드 사전 파일을 이용해 접속 계정을 알아내는 기법 ◆ 공격대상의 개인정보를 충분히 안다면 매우 효율적 |
| 무차별 공격 | <ul style="list-style-type: none"> ◆ 성공할 때까지 모든 조합의 경우의 수를 시도해 공격하는 기법 ◆ 워다이얼링 등에도 사용 |
| 패스워드 하이브리드 공격 | <ul style="list-style-type: none"> ◆ 사전 공격 + 무차별 공격 |
| 레인보우 테이블 공격 | <ul style="list-style-type: none"> ◆ 패스워드를 해시 처리하여 패스워드와 해시로 이루어진 체인을 많이 만들어 놓은 테이블을 가지고 대입하여 공격하는 방식 ◆ 사전 공격, 무차별 대입 공격 수행보다 훨씬 적은 시간 소요 |

- 패스워드 크래커의 공동적인 대응책으로는 부가적인 숫자를 패스워드에 덧붙인 후 암호화하여 저장하는 방법으로 사전공격에 대한 내성 향상, 패스워드가 평문으로 전송되지 않도록 하여 패스워드 패킷 스니핑 방지, 반복적인 트래픽 감시, 패스워드 입력횟수 제한의 임계치 적용 등이 있음

[2] 사회공학 공격

- 보안학적 측면에서 기술적인 방법이 아닌 기본적인 신뢰를 기반으로 사람을 속여 비밀 정보를 획득하는 기법

- 인간 기반 사회공학 공격과 컴퓨터 기반 접근으로 분류 가능

(1) 인간 기반 사회공학 공격

1) 직접적인 접근

- 조직의 높은 위치에 있는 사람으로 가장하여 정보 획득
- 동정심에 호소하여 무척 긴급한 상황에 도움이 필요한 것처럼 행동하여 정보 획득
- 가장된 인간관계를 이용해 조직내 개인정보 획득

2) 도청

- 도청 장치 설치
- 유선 전화선을 중간에 따기
- 유리나 벽의 진동을 레이저로 탐지하여 음성으로 변경

3) 어깨 너머로 훑쳐보기

- 작업 중인 사람의 뒤에 다가가 그 사람이 수행하는 업무 관련 정보나 패스워드 등을 알아내는 방법

4) 휴지통 뒤지기

- 공격하는 조직에 대한 정보를 수집하기 위해 해당 정보를 포함하는 쓰레기를 뒤져 정보를 찾아내는 방법

5) 테일게이팅, 피키배킹

- 부정 인증을 이용한 출입 방법의 하나, 의도를 가지지 않는 부정 출입을 테일게이팅, 의도를 가진 부정 출입을 피키배킹이라고 함

(2) 컴퓨터 기반 사회공학 공격

1) 피싱

- 수신자에게 이메일을 발송하여 위조된 사이트로 이동시킨 후 사이트 개편 등의 이유로 고객 정보를 요구

2) 파밍

- 위조 사이트 개설 후 원래 사이트 접속 시 위조된 사이트로 접근되도록 방향 재지정
- 피싱보다 속기 쉬워 피해를 당할 우려가 더 큼

3) 스미싱

- SMS를 통해 사용자에게 트로이목마, 악성코드를 설치하도록 유도
- 개인정보를 빼내거나 소액결제 등을 실행해 금전적 손해를 입힘

4) 은닉채널

- 개체가 허가되지 않는 방식으로 정보를 얻음
- 통신 대역폭을 제한하고 로그분석, 시스템 자원 분석 등을 수행

5) 스미싱

- 컴퓨터와 장치에서 방출되는 전기적 신호 가로채기

- 사회공학 공격은 정보수집, 관계형성, 공격, 실행의 순으로 진행됨

- 사회공학 공격 대응 전략은 정보 수집 단계의 대응, 공격 단계의 대응, 실행 단계의 대응으로 구분

- 사회공학 기법을 활용한 공격에 대응할 수 있는 전략은 정보 수집 단계의 대응, 공격 단계의 대응, 실행 단계의 대응으로 나눌 수 있음

| | |
|--------------|---|
| 정보 수집 단계의 대응 | ◆ 개인 신상 정보 관련 문서 관리, 온라인상의 개인 정보 관리 |
| 공격 단계의 대응 | ◆ 사회공학 공격 형태 인지, 배경조사 |
| 실행 단계의 대응 | ◆ 관련 기관에 신속하게 신고 ◆ 신고 후 기관의 도움을 받아 공격자가 요청한 사항을 파악 |