어플리케이션 보안의 기본

msdn
Microsoft® Developer Network

세션 선수 조건

- Microsoft Visual Basic® , Microsoft Visual C++® 또는 C#을 사용해서 개발해 본 경험

수준 200

목차

- 어플리케이션 보안의 중요성
- 안전한 어플리케이션 개발 기법
- 보안 기술
- 안전한 개발 지침

신뢰할 수 있는 컴퓨팅

"신뢰할 수 있는 컴퓨팅에는 다음과 같은 네 가지 기본 요소가 있습니다.

안정성이란 컴퓨터 시스템이 신뢰할 수 있으며, 필요할 때 즉시 사용이 가능하며, 예상되는 적절한 수준의 성능을 나타내는 것입니다.

보안성은 시스템이 공격을 받더라도 쉽게 복구될 수 있으며, 시스템 및 데이터의 기밀성, 무결성 및 가용성이 보호되는 것을 말합니다.

개인 정보 보호란 사용자가 자신의 개인 정보를 통제할 수 있으며 조직에서 각 사용자의 개인 정보를 성실하게 보호하는 것입니다.

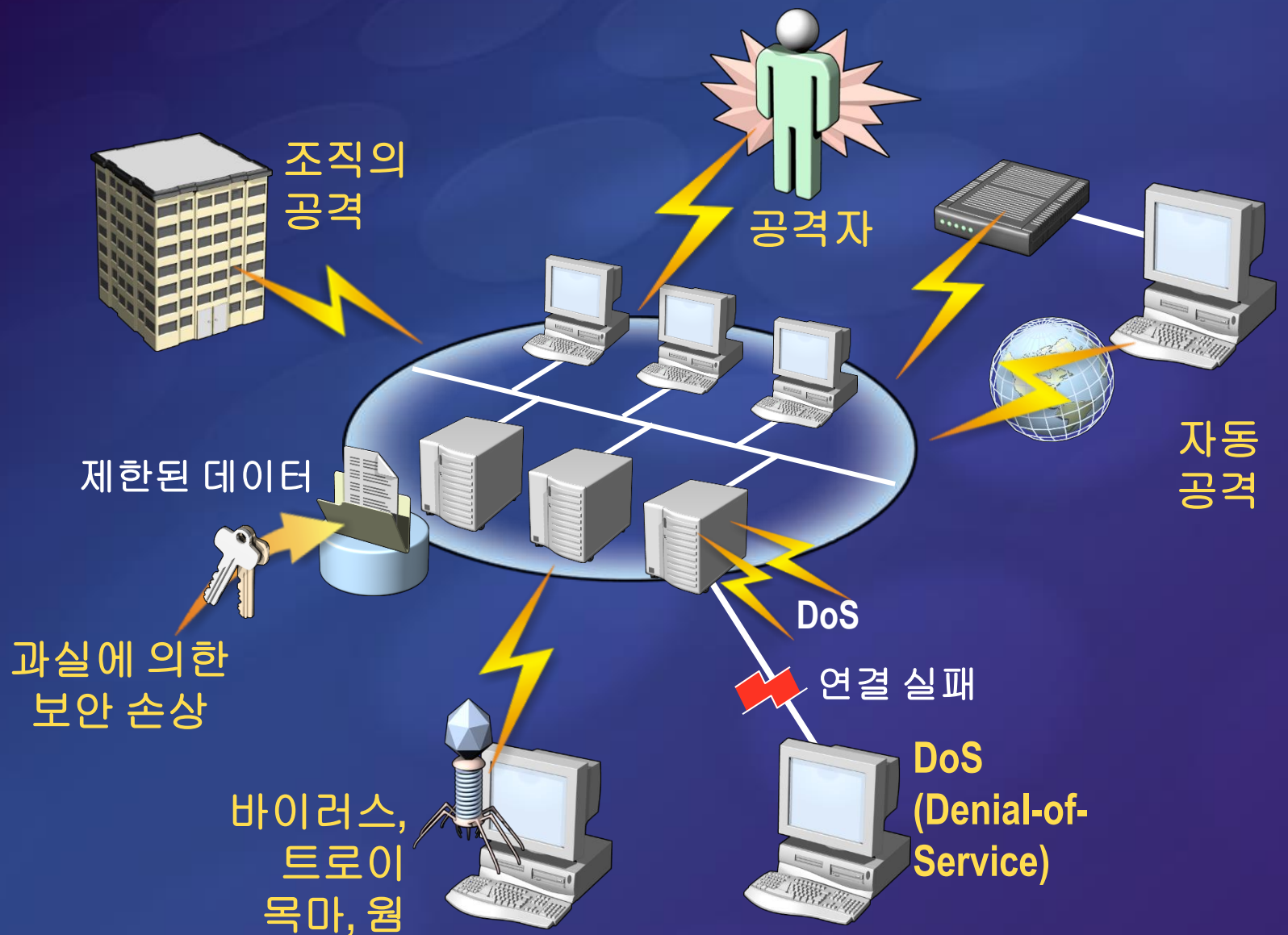
비즈니스 무결성은 고객에 대해 책임을 지고, 고객이 비즈니스 문제를 해결할 수 있는 적합한 솔루션을 찾도록 지원하며, 제품 또는 서비스 관련 문제를 해결해 주고 고객과 항상 의견을 주고 받는 회사와 관련된 개념입니다."

- Bill Gates

연결 시나리오 및 보안 고려 사항

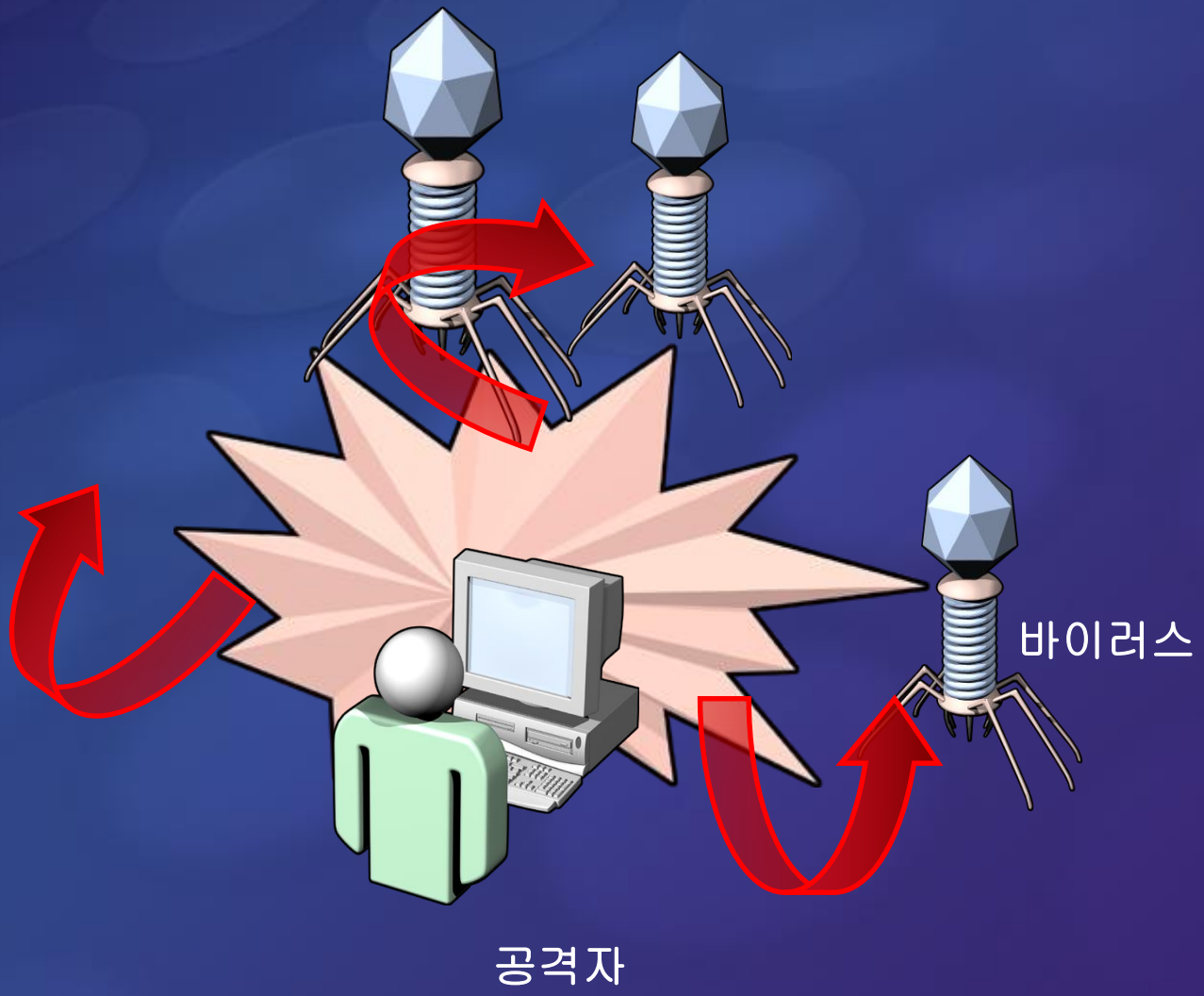
- 연결 시나리오:
  - 기존의 유선 연결
  - 이동이 잦은 직원
  - 공용 무선 네트워크
- 보안 고려 사항:
  - 어플리케이션의 인터넷 의존성
  - 비즈니스의 인터넷 의존성
  - 내부 보안 공격

일반적인 공격 유형


조직의
공격
공격자
자동
공격
제한된 데이터
과실에 의한
보안 손상
DoS
연결 실패
바이러스,
트로이
목마, 웜
DoS
(Denial-of-
Service)

보안 침입의 예

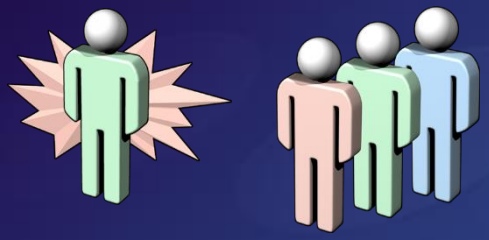
- CodeRed
- ILoveYou
- Nimda



취약한 보안에 의한 피해

- 지적재산권 침해
- 시스템 중단 시간
- 생산성 손실
- 비즈니스 평판 실추
- 고객 신뢰 하락
- 수익 손실로 인한 심각한 재정적 손해

보안 구현 시 직면하는 과제



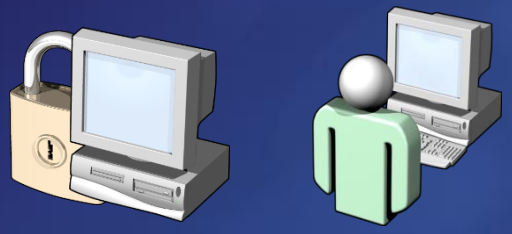
공격자 대 방어자

공격자는 단 하나의 취약점만 알아도 된다.

방어자는 모든 위치를 방어해야 한다.

공격자는 아무 때나 공격할 수 있다.

방어자에게는 시간과 비용이 제한되어 있습니다.



보안 대 사용 편의성

보안 시스템은 사용하기가 더 어렵습니다.

복잡하고 강력한 암호는 기억하기 어렵습니다.

사용자는 단순한 암호를 선호합니다.

뒤늦은 보안 조치

개발자 및 관리자는 보안은 어떠한 비즈니스 가치도 더하지 못한다고 생각합니다.

제품이 출시되기 전에 취약점을 해결하는 작업에는 많은 비용이 듭니다.

어플리케이션 보안에서 개발자의 역할

- 개발자의 임무:
  - 솔루션 아키텍트 및 시스템 관리자와 협력하여 어플리케이션 보안 구현
  - 보안에 기여:
    - 바람직한 어플리케이션 보안 개발 기법 적용
    - 보안이 취약한 지점을 파악하고 이를 방지하는 방법 습득
    - 안전한 프로그래밍 기술 사용

목차

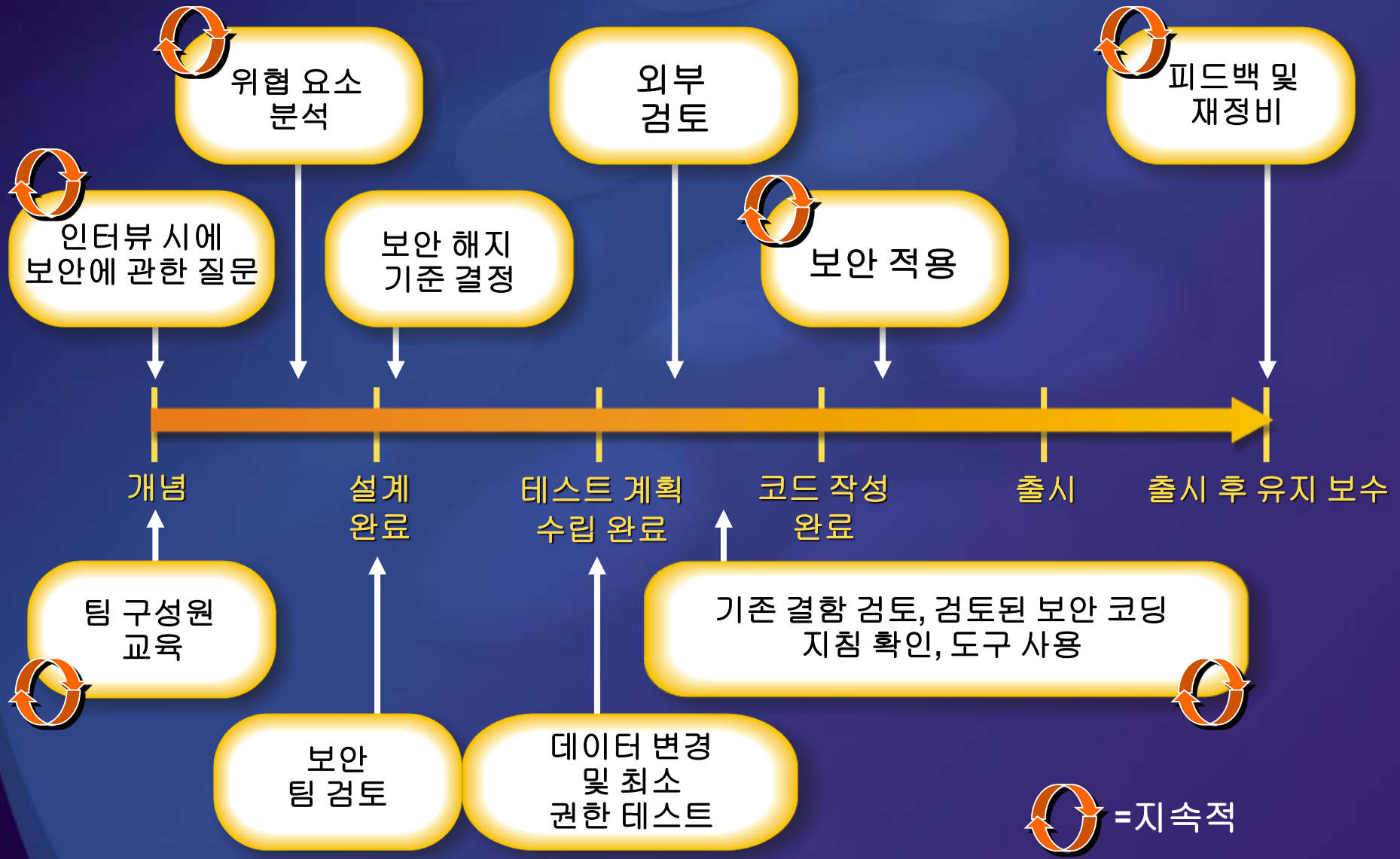
- 어플리케이션 보안의 중요성
- 안전한 어플리케이션 개발 기법
- 보안 기술
- 안전한 개발 지침

보안에 대한 총괄적인 접근

- 보안을 고려해야 하는 단계:
  - 프로젝트의 모든 단계
    - 설계
    - 개발
    - 배포
  - 모든 계층
    - 네트워크
    - 호스트
    - 어플리케이션

"보안 수준은 가장 약한 고리에 의해 결정된다."

프로젝트 라이프 사이클 전반에 걸쳐 보안 고려



SD³ 보안 프레임워크

SD³

보안을 고려한 설계

- 안전한 아키텍처 및 코드
- 위협 요소 분석
- 취약점 제거

보안을 고려한 기본 설정

- 공격 받을 수 있는 범위 축소
- 사용하지 않는 기능은 기본적으로 해제
- 최소한의 권한 사용

보안을 고려한 운용

- 보호: 탐지, 방어, 복구, 관리
- 프로세스: 배포 방법 안내, 아키텍처 지침
- 사용자: 교육

위협 모델링

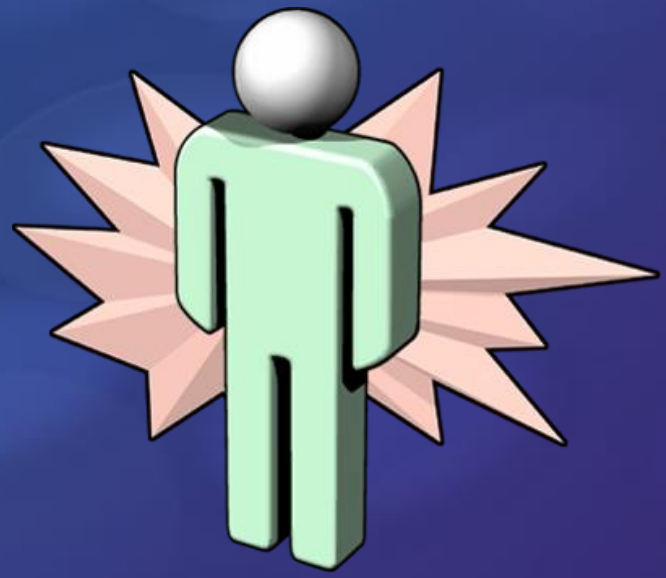
- 위협 모델링이란:
  - 어플리케이션을 보안을 기반으로 분석
  - 설계 과정의 중요한 부분
- 위협 모델링:
  - 어플리케이션 보안 비용 절감
  - 논리적이며 효율적인 프로세스 제공
  - 개발 팀 지원:
    - 어플리케이션에서 가장 취약한 지점 파악
    - 제거해야 할 위협 요소 및 그 해결 방법 결정

지속적인 교육

- 교육 내용:
  - 보안 기능의 작동 방식
  - 보안 기능을 이용한 보안 시스템의 구축
  - 결함 있는 코드를 식별하기 위해 보안 취약점 사전 숙지
  - 일반적인 보안 취약점을 방지하는 방법
  - 실수가 반복되지 않도록 방지하는 방법

입력 유효성 검사

- 버퍼 오버런
- SQL 삽입
- 교차 사이트 스크립팅



"모든 입력 내용은 위험하지 않다는 것이
입증되기 전까지는 위험하다!"

데모 1
버퍼 오버런

보안 검사 통과

보안을 강화할 수 있는 기법

기법	이점
위협 모델링 적용	• 보안 취약점 파악 • 어플리케이션 아키텍처 이해
개발 팀 교육	• 일반적인 보안 결함 방지 • 보안 기술의 올바른 적용

코드 검토	• 다음과 같은 코드 보호
	  ◦ 네트워크에 액세스하는 코드
	  ◦ 기본적으로 실행되는 코드
	  ◦ 인증되지 않은 프로토콜을 사용하는 코드
	  ◦ 높은 권한으로 실행되는 코드

도구 사용	• 취약점에 대한 더욱 일관된 테스트
인프라 솔루션 사용	• **SSL/TLS** 및 **IPSec**를 통한 보안 강화
구성 요소 솔루션 사용	• **CAPICOM** 및 **.NET Cryptography** 네임스페이스를 통해 더욱 강화된 보안
관리되는 코드로 마이그레이션	• 일반적인 취약점 방지

목차

- 어플리케이션 보안의 중요성
- 안전한 어플리케이션 개발 기법
- 보안 기술
- 안전한 개발 지침

보안 기술의 개요

- 개발자가 사용 및 적용해야 할 기술:
  - 암호화
  - 해싱
  - 디지털 서명
  - 디지털 인증서
  - 안전한 통신
  - 인증
  - 권한 부여
  - 방화벽
  - 감사
  - 서비스 팩 및 업데이트

암호화

- 암호화는 데이터를 인코딩하는 프로세스
  - 사용자의 ID나 데이터를 함부로 읽지 못하도록 보호
  - 데이터 변경 방지
  - 데이터를 보낸 상대방을 확인
- 암호화의 유형:
  - 비대칭
  - 대칭

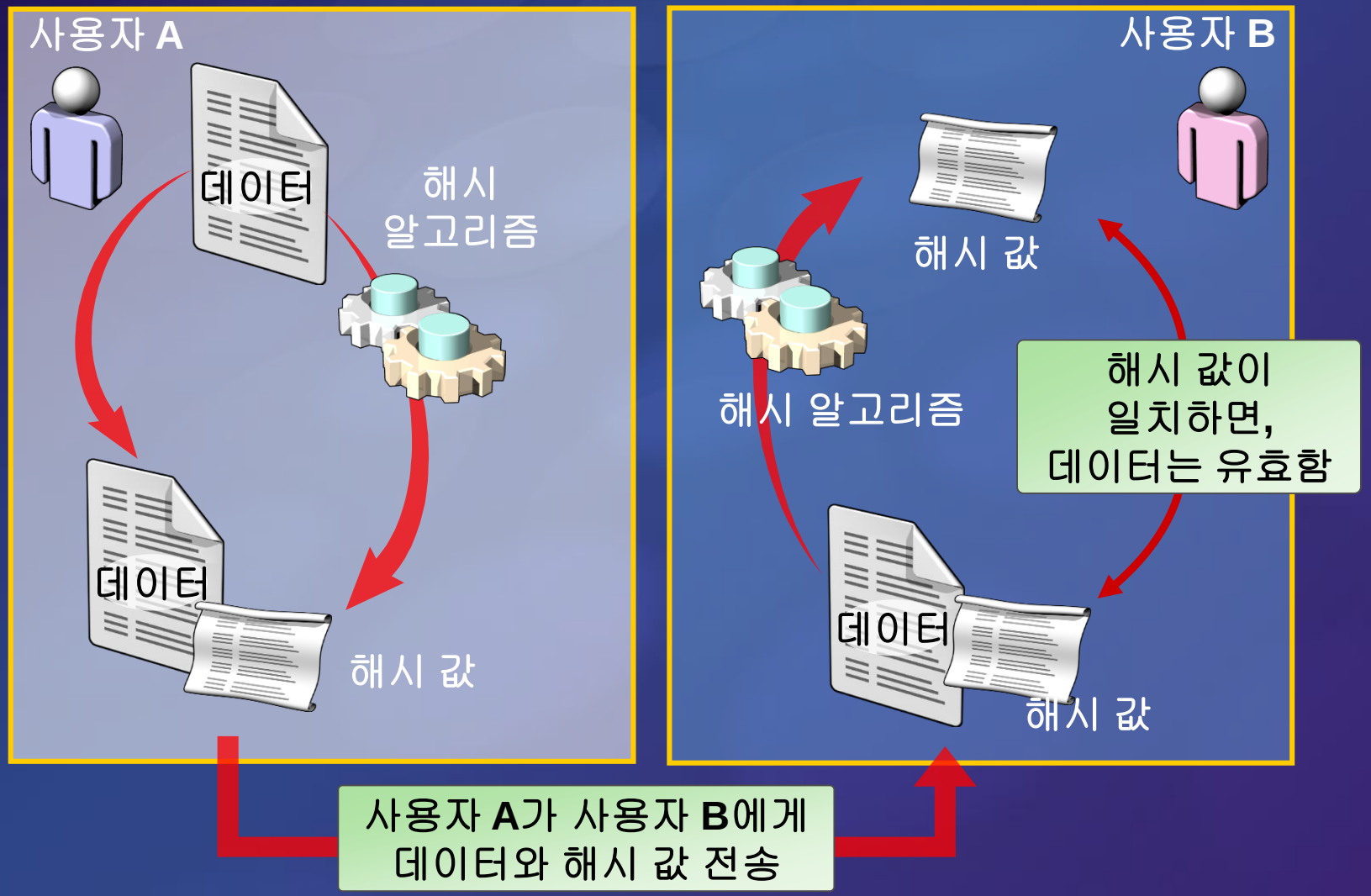
대칭 대 비대칭 암호화

알고리즘 유형	설명

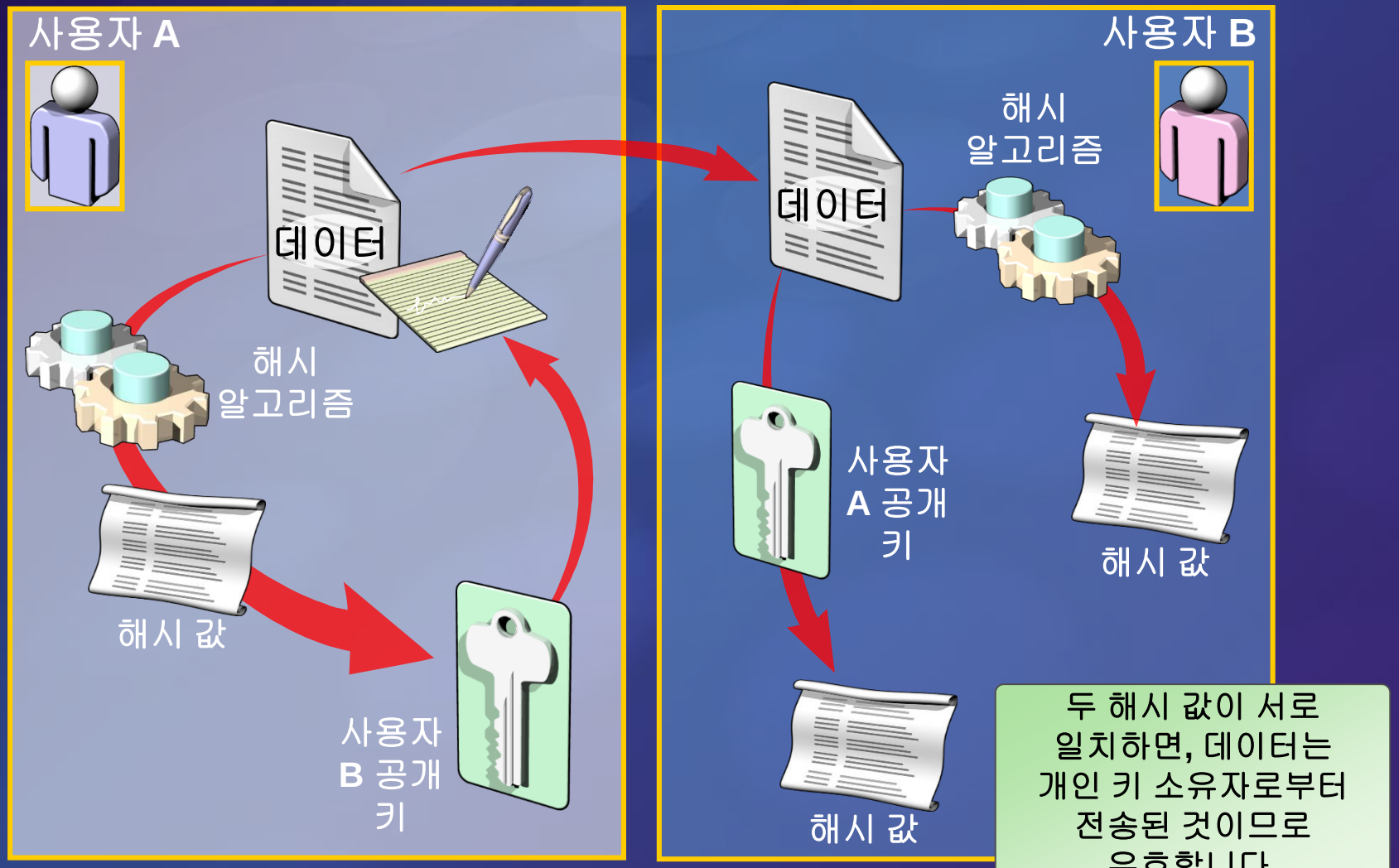
대칭	• 하나의 키 사용:
	  • 데이터 암호화
	  • 데이터 해독
	• 빠르고 효율적

비대칭	• 수학적으로 연관된 두 개의 키 사용:
	  • 공개 키로 데이터 암호화
	  • 개인 키로 데이터 해독
	• 대칭 암호화보다 더욱 안전
	• 대칭 암호화보다 느림

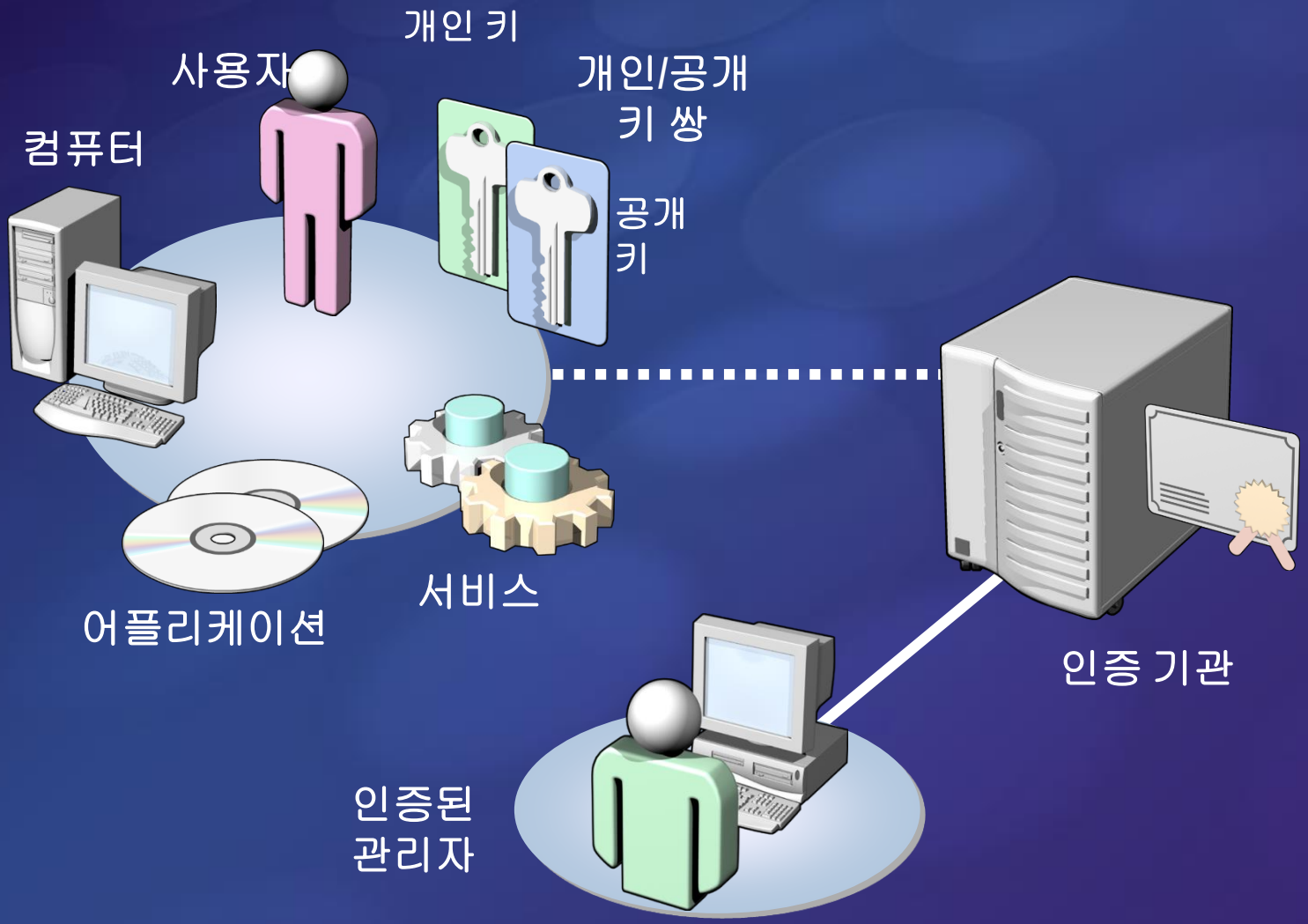
해시를 사용한 데이터 무결성 검사



디지털 서명

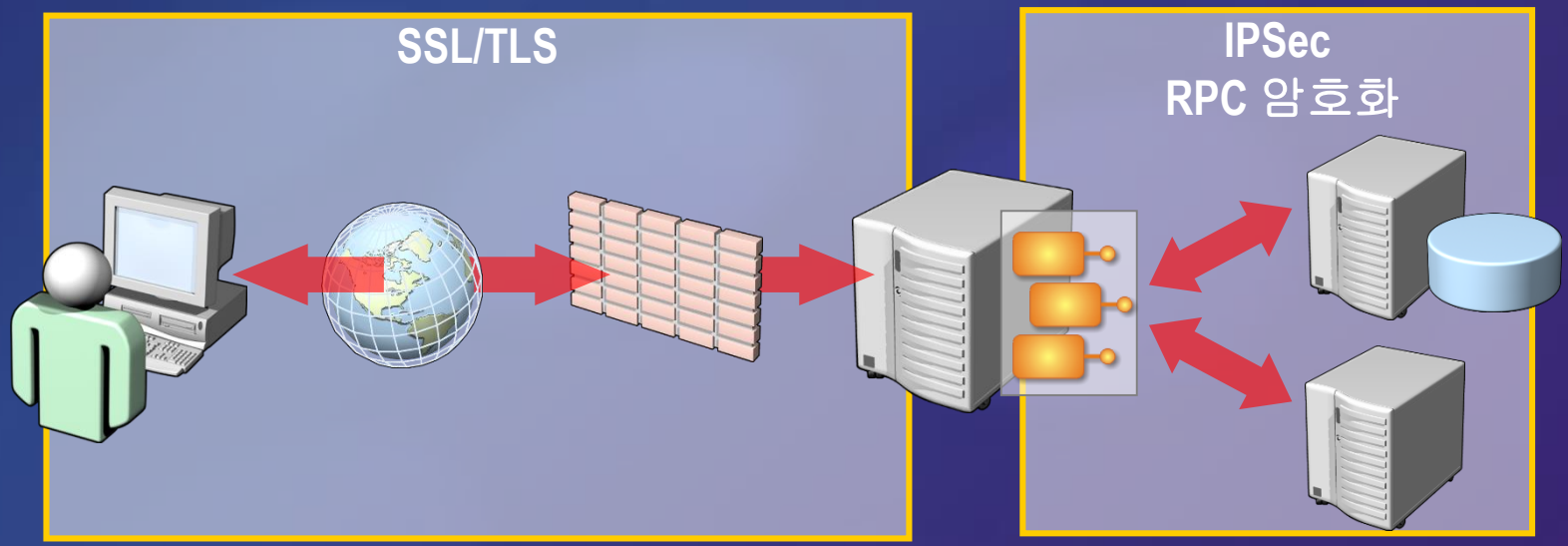

사용자 A
데이터
해시
알고리즘
해시 값
사용자
B 공개
키
사용자 B
데이터
해시
알고리즘
사용자
A 공개
키
해시 값
해시 값
두 해시 값이 서로
일치하면, 데이터는
개인 키 소유자로부터
전송된 것이므로
유효합니다.

디지털 인증서의 작동 방식

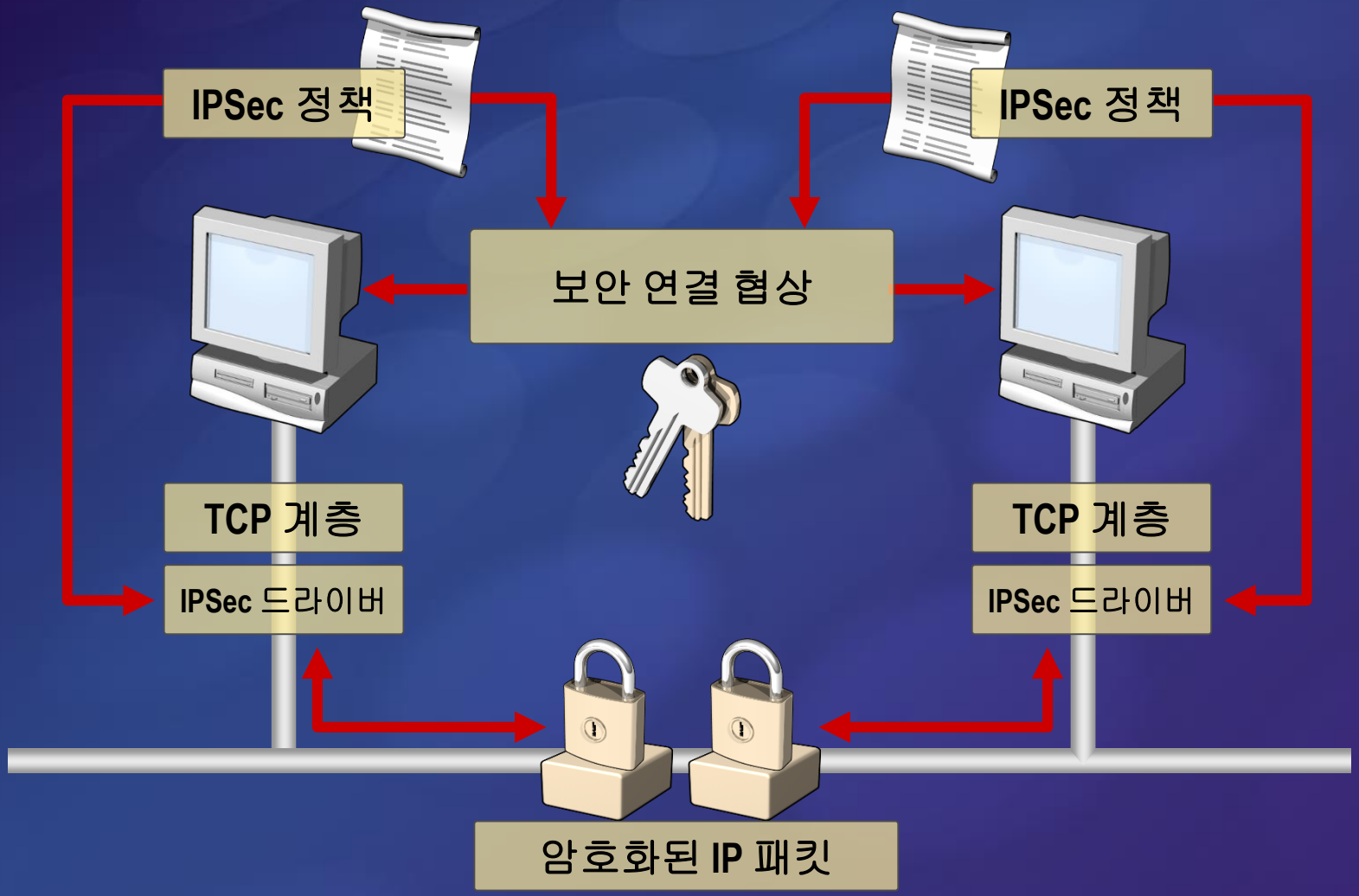


안전한 통신 기술

- 종류:
  - IPSec
  - SSL
  - TLS
  - RPC 암호화



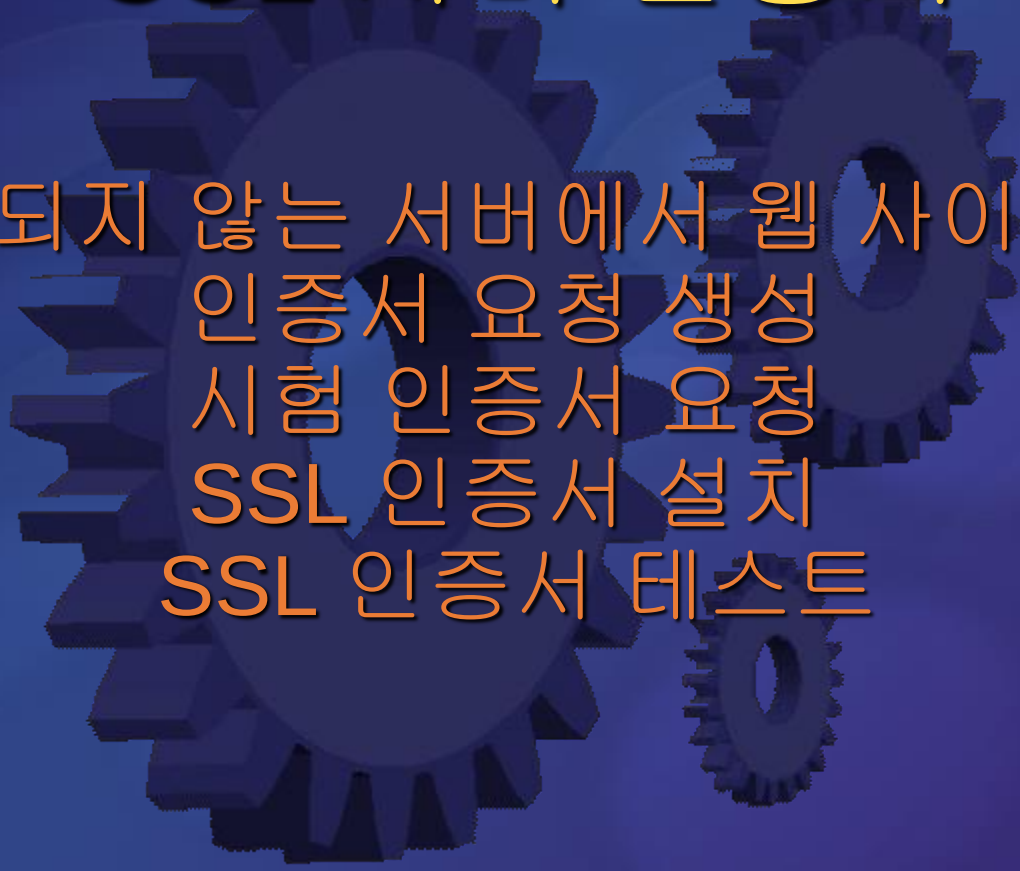
안전한 통신 IPSec의 작동 방식



안전한 통신 SSL의 작동 방식

1. 사용자는 HTTPS를 사용하여 보안 웹 서버를 탐색합니다.
2. 브라우저는 고유한 세션 키를 생성하며 루트 인증서에서 생성된 웹 서버의 공개 키를 사용하여 세션 키를 암호화합니다.
3. 웹 서버는 세션 키를 받아 서버의 개인 키를 사용하여 해독합니다.
4. 연결이 설정되면 브라우저와 웹 서버 간의 모든 통신은 보호됩니다.

데모 2
SSL 서버 인증서



보안이 되지 않는 서버에서 웹 사이트 보기
인증서 요청 생성
시험 인증서 요청
SSL 인증서 설치
SSL 인증서 테스트

인증 인증의 목적

- 다음 방법으로 사용자 ID 확인:
  - 자격 증명 수신
  - 자격 증명의 유효성 검사
- 어플리케이션이 호출자를 알 수 있도록 하여 통신 보호

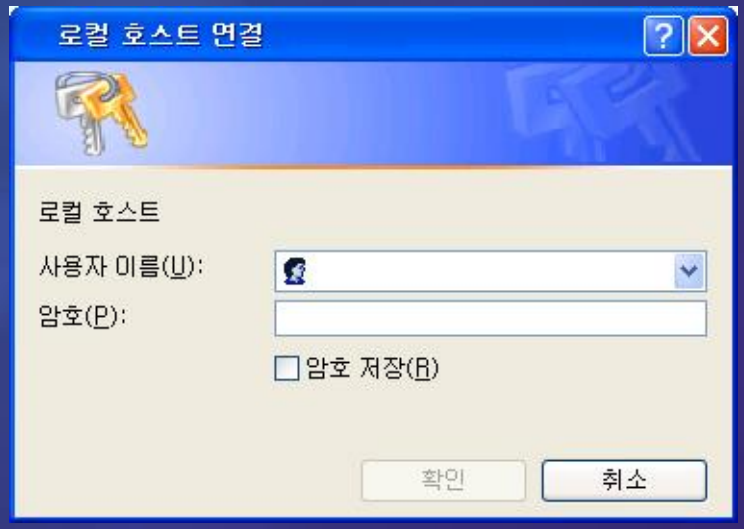
데이터 암호화만으로는 불충분!

인증 인증 방식

- 기본
- 다이제스트
- 디지털 서명 및 디지털 인증서
- 통합
  - Kerberos 버전 5 프로토콜
  - NTLM
- Microsoft Passport
- 생체 인식

인증 기본 인증

- 단순하지만 효과적
- 모든 주요 브라우저 및 서버에서 지원
- 프로그래밍하거나 설정하기 용이
- 사용자 자격 증명을 관리
- SSL/TLS 요구

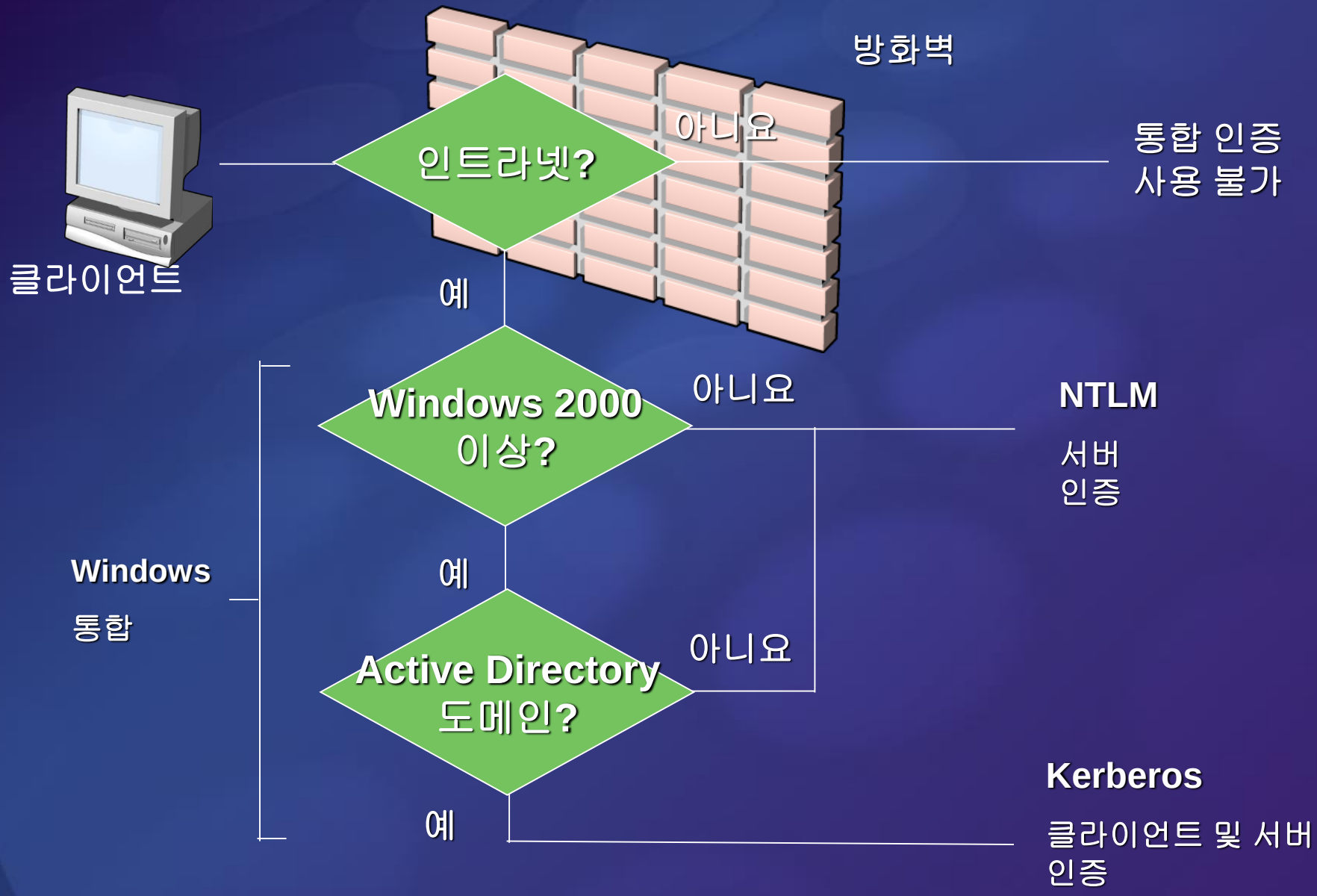


인증 다이제스트 인증의 작동 방식

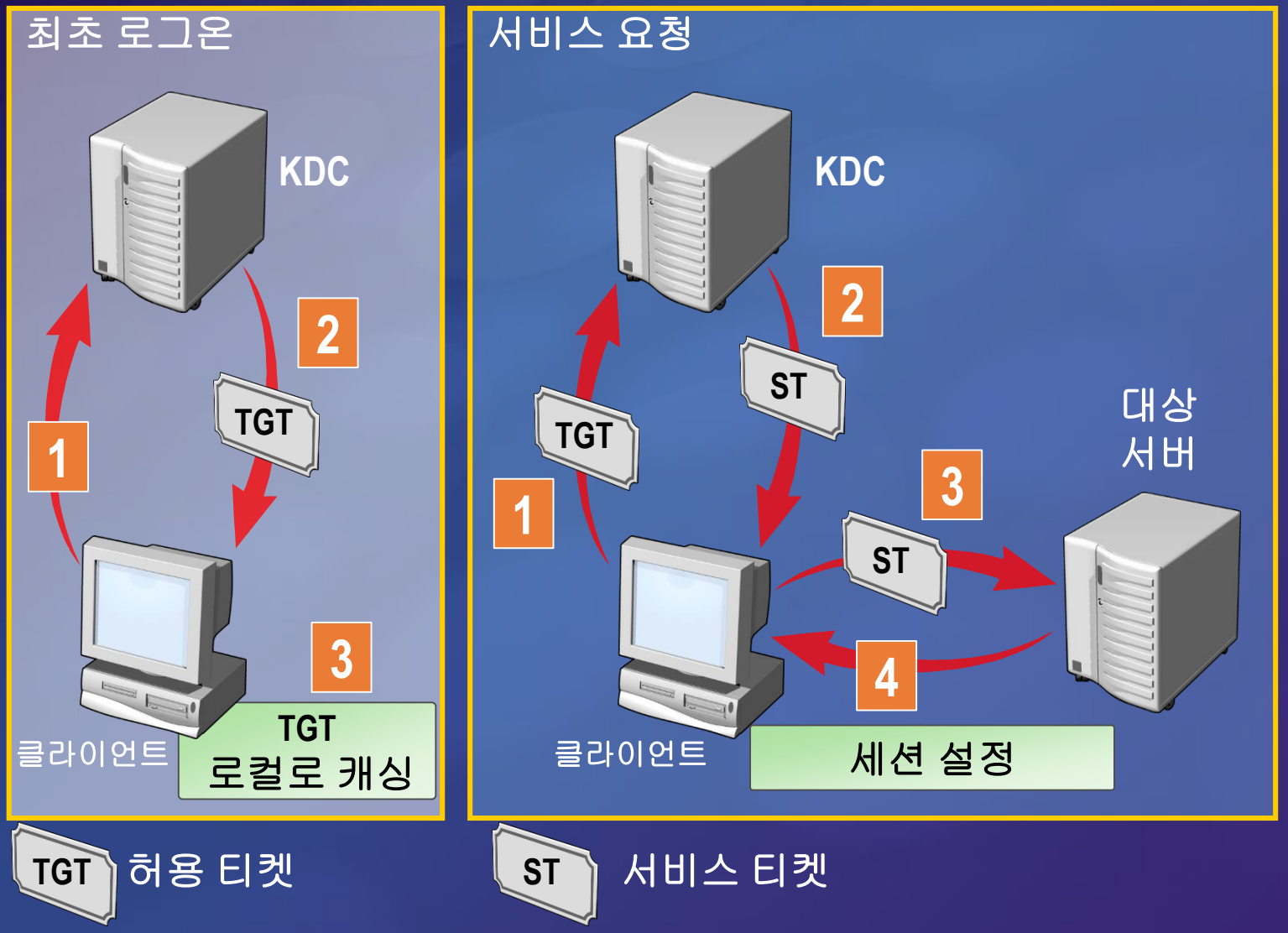
인증 클라이언트 디지털 인증서

- 웹 어플리케이션에서 사용
  - 서버에서는 X.509 서버 인증서와 함께 SSL/TLS를 사용하여 통신을 보호
  - 서버에서는 필요하면 클라이언트 X.509 인증서의 데이터를 사용하여 클라이언트를 인증
  - 인증 기관은 인증서를 발급. 해당 인증서의 루트 인증서는 서버가 보유
- 분산 어플리케이션에서 사용
  - 어플리케이션에서는 SSL/TLS 통신 채널 사용
  - 클라이언트 및 서버 어플리케이션에서는 인증서를 사용하여 인증
- 스마트 카드 형태로 배포 가능

인증 통합 인증이 필요한 경우



인증 Kerberos 버전 5 사용 방법



데모 3

IIS 인증 기술

익명 인증 사용
기본 인증 사용
통합 Windows 인증 사용

권한 부여 권한 부여란?

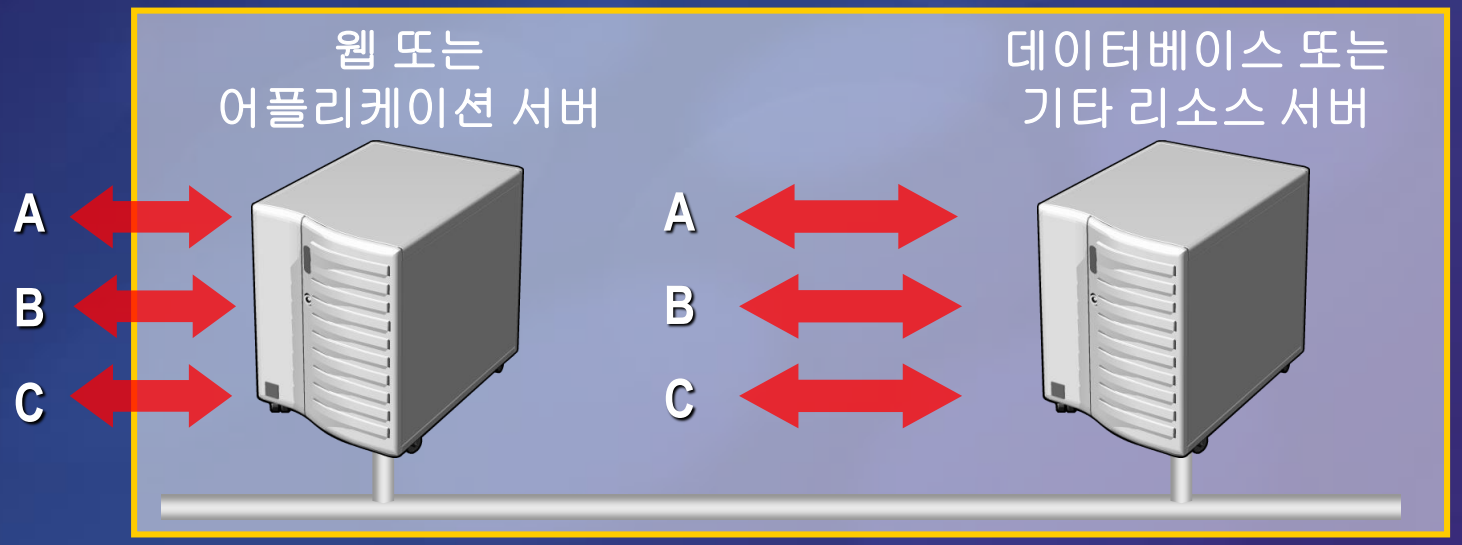
- 권한 부여:
  - 클라이언트 요청이 인증된 후에 이루어짐
  - 인증된 사용자가 특정 리소스에 액세스할 수 있는지 확인하는 과정
  - 파일, 폴더, 레지스트리 설정, 어플리케이션 등에 할당된 권한 검사
  - 역할을 기반으로 부여 가능
  - 코드를 기반으로 부여 가능

권한 부여 일반적인 권한 부여 기법

- IIS 웹 사용 권한(및 IP/DNS 제한)
- .NET 역할 기반 보안
- .NET 코드 액세스 보안
- NTFS ACL(액세스 제어 목록)
- SQL Server 로그인
- SQL Server 사용 권한

권한 부여 가장/위임 모델

- 클라이언트 ID를 사용하여 다운스트림 리소스에 액세스합니다.



권한 부여 신뢰할 수 있는 하위 시스템 모델

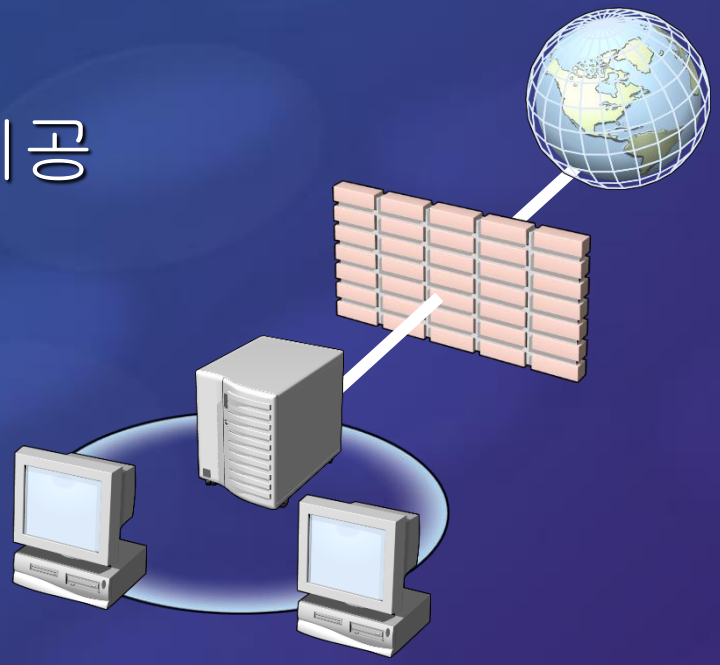
- 클라이언트가 역할에 매핑됩니다.
- 각 역할이 다운스트림 리소스에 액세스할 때 전용 Windows 서비스 계정이 사용됩니다.

데모 4
신뢰할 수 있는 하위 시스템 모델
권한 부여 기법

어플리케이션 검토
웹 서버에서 인증 설정
웹 서버에서 서비스 계정 만들기
데이터베이스 서버에서 권한 부여 설정

방화벽

- 방화벽의 기능:
  - 내부 클라이언트를 위한 안전한 인터넷 게이트웨이 제공
  - 패킷 필터링
  - 회선 수준 필터링
  - 어플리케이션 필터링
  - 감사
- 방화벽의 한계:
  - HTTP 또는 HTTPS를 통한 어플리케이션 수준 공격을 막을 수 없음



감사

- 감사의 추적 대상:
  - 리소스 액세스 및 사용
  - 성공하거나 실패한 로그온 시도
  - 어플리케이션 장애
- 감사의 이점:
  - 의심스러운 작업이나 침입을 관리자가 좀더 쉽게 탐지하도록 지원
  - 부인할 수 없는 법적인 분쟁에 대비하여 추적 가능
  - 보안 침해 조사

서비스 팩 및 업데이트

보안 업데이트	설명
핫픽스	• 하나의 문제 또는 몇 가지 문제를 처리 • QChain으로 결합 가능
보안 롤업 패키지	• 설치하기 쉽게 여러 핫픽스를 패키지로 만든 것
서비스 팩	• 주요 업데이트 제공 • 이전 업데이트의 누적 • 아직 발표되지 않은 픽스가 들어 있을 수 있음 • 변경된 기능이 들어 있을 수 있음

목차

- 어플리케이션 보안의 중요성
- 안전한 어플리케이션 개발 기법
- 보안 기술
- 안전한 개발 지침

순향적 보안 개발

- 보안 개선 노력을 전체 개발 과정과 통합
- 보안에 중점을 두고, 새로운 공격에 대비하여 코드 작성
- 교육 필요성 홍보
  - 팀 내 인식 강화
  - 자신이나 타인의 실수를 통해 배우기

SD3 보안 프레임워크 적용

보안을 고려한 설계	위협 모델 구축 코드 검토, 침입 테스트 최소한의 권한으로 코드 실행
보안을 고려한 기본 설정	공격 받을 수 있는 범위 최소화 보안을 염두에 두고 서비스를 활성화
보안을 고려한 운용	유용한 보안 정보 활용 보안 지침 작성 어플리케이션 보안 평가 도구 작성

Microsoft Java Virtual Machine 지원 종료 안내

- Java 지원 경고!
  - 이제 더 이상 MSJVM이 Windows XP SP1a 또는 Windows Server 2003에 포함되지 않습니다.

  Microsoft는 2004년 9월 30일에 지원을 종료합니다.
  - 이 날짜 이후에는 보안 픽스가 제작되지 않을 것입니다.
  - 이 날짜 이후에 발생하는 보안 문제로 인해 MSJVM이 제거될 수도 있습니다.
- 개발자가 수행해야 하는 작업
  - MSJVM 관련 어플리케이션 업데이트
  - 고객에게 업그레이드 제공
- 추가 정보:
  - http://www.microsoft.com/korea/java

세션 요약

- 어플리케이션 보안의 중요성
- 안전한 어플리케이션 개발 기법
- 보안 기술
- 안전한 개발 지침

다음 단계

1. 보안 관련 뉴스
   - 보안 게시판에 가입:
     http://www.microsoft.com/security/security_bulletins/alerts2.asp(영문)
   - 최신 Microsoft 보안 지침:
     http://www.microsoft.com/security/guidance/(영문)
2. 추가 보안 관련 교육
   - 교육 세미나 강좌 및 온라인 교육:
     http://www.microsoft.com/korea/seminar/security.asp
   - 실습 교육(해당 지역의 CTEC에서 제공):
     http://www.microsoft.com/korea/traincert/

추가 정보

- Microsoft 보안 사이트(모든 사용자)
  - http://www.microsoft.com/korea/security
- MSDN 보안 사이트(개발자)
  - http://msdn.microsoft.com/security(영문)
- TechNet 보안 사이트(IT 전문가)
  - http://www.microsoft.com/korea/technet/security

Microsoft®