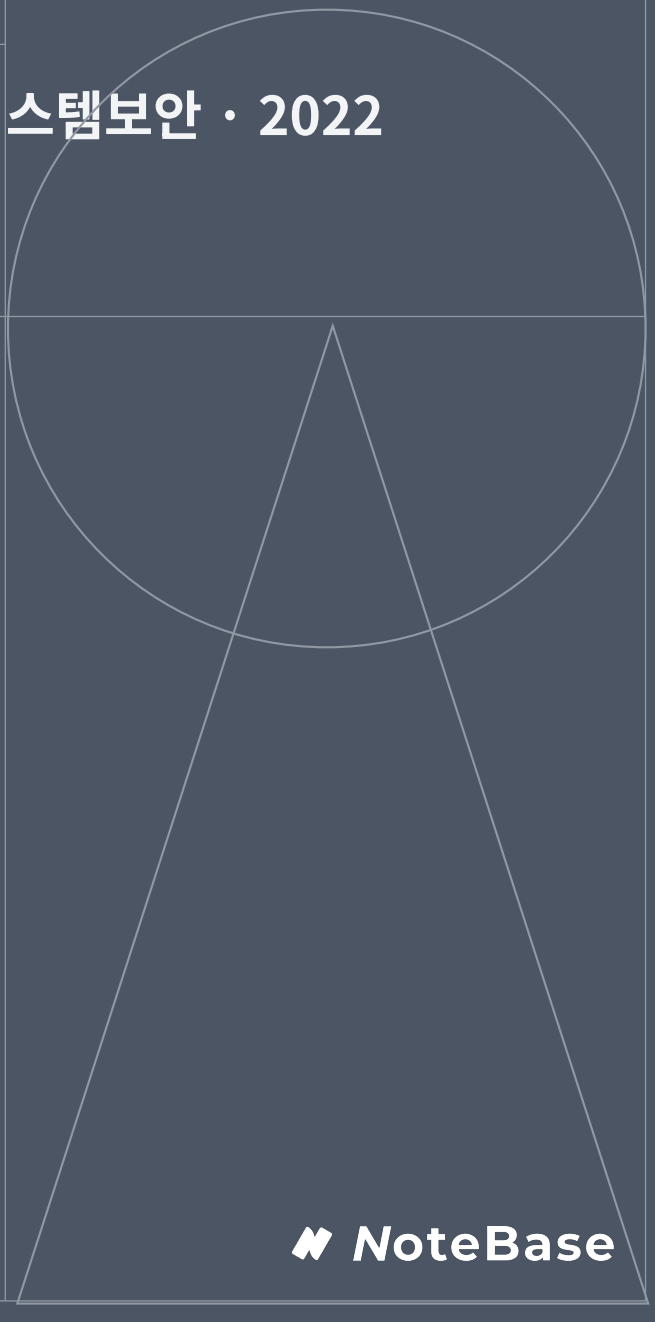
기출OX 해설집

공무원/국가직/9급 · 정보시스템보안 · 2022



NoteBase

기출OX 해설집

공무원/국가직/9급 · 정보시스템보안 · 2022

공무원/국가직/9급 · 정보시스템보안 · 2022

문 1. CPU에서 실행되는 머신 코드가 위치하는 시스템 메모리 영역은?

① 스택

② 힙

③ 텍스트

④ 데이터

정답 3번

간단한 해설

CPU가 실행할 **머신 코드(명령어)**가 저장되는 곳은 **텍스트 영역**이다.

개념 구분

텍스트 영역: 프로그램의 **실행 명령어인 머신 코드**를 저장한다. CPU는 이 영역의 명령어를 가져와 해석하고 실행하므로, 문제에서 묻는 메모리 영역은 텍스트 영역이다.

다른 영역과의 구분: 데이터 영역은 초기값이 있는 전역·정적 변수 등을, 힙 영역은 실행 중 동적으로 할당한 자료를, 스택 영역은 함수 호출에 따른 지역 변수·복귀 주소 등을 저장한다. 이들은 주로 **명령어가 처리할 자료나 함수 실행에 필요한 정보**를 보관한다는 점에서 텍스트 영역과 구별된다.

NoteBase

문 2. 사용자와 운영체제 사이에서 중간자 역할을 수행하며, 명령어 해석 기능, 프로그래밍 기능, 사용자 환경 설정 기능을 제공하는 것은?

① 데몬

② 커널

③ 프로세스

④ 셸

정답 4번

간단한 해설

셸은 사용자의 명령을 해석하여 실행하는 인터페이스이다. 여러 명령을 묶는 셸 스크립트와 환경 변수 설정도 지원하므로, 지문의 세 기능에 부합한다.

개념 구분

셸: **사용자의 명령을 해석하고 실행**하는 프로그램으로, 사용자와 운영체제를 연결한다. 명령을 직접 입력하여 실행할 수 있고, 변수·조건문·반복문 등을 이용한 셸 스크립트로 작업을 자동화할 수도 있다. 환경 변수나 초기 설정 파일을 이용하여 명령 검색 경로 등 사용자 환경을 설정한다.

커널과의 구분: 커널은 CPU·메모리·입출력 장치 등 **시스템 자원을 관리**하는 운영체제의 핵심 부분이다. 셸이 사용자의 명령을 받아 해석한다면, 커널은 프로그램 실행에 필요한 자원과 서비스를 제공한다. 따라서 명령어 해석과 사용자 환경 설정이라는 단서는 셸을 가리킨다.

프로세스·데몬과의 구분: 프로세스는 **실행 중인 프로그램**이며, 데몬은 보통 백그라운드에서 서비스 작업을 수행하는 프로세스이다. 실행 중인 셸도 프로세스이지만, '프로세스'라는 말 자체가 명령어 해석 기능을 뜻하지는 않는다. 데몬 역시 사용자 명령을 해석하는 인터페이스를 가리키는 명칭이 아니다.

NoteBase

문 3. (가)에 들어갈 용어로 옳은 것은?

보안 운영체제는 기존의 운영체제에 내재된 보안상의 결함으로 인한 각종 침해로부터 시스템을 보호하기 위하여 기존의 운영체제 내에 보안 기능을 통합시킨 (가)을/를 추가로 이식한 운영체제이다. 이를 통해 사용자의 모든 접근 행위가 안전하게 통제된다. 이것은 하드웨어, 운영체제 및 기타 시스템 요소 간의 보안 인터페이스를 제공한다.

① 보안 커널
② 접근 제어
③ 신뢰 플랫폼 모듈
④ 하이퍼바이저

정답 1번

간단한 해설

운영체제에 통합되어 **모든 접근을 보안 정책에 따라 중재**하는 핵심 구성 요소가 보안 커널이다. 접근 제어는 이 구성 요소가 수행하는 기능을 가리킨다.

빈칸별 해설

(가) 보안 커널: 사용자나 프로세스가 파일·메모리 같은 자원에 접근할 때, 보안 정책에 따라 허용 여부를 판단하고 집행하는 핵심 구성 요소이다. 모든 접근을 검사한다는 '참조 모니터' 개념을 하드웨어·펌웨어·소프트웨어로 구현한다. 접근이 보안 검사를 우회하지 못하도록 **모든 접근을 빠짐없이 중재**해야 하므로, 지문의 '사용자의 모든 접근 행위가 안전하게 통제된다'는 설명과 연결된다. 예를 들어 프로세스가 파일을 읽으려 할 때 권한을 검사하여 허용된 읽기만 수행하게 한다. 또한 검사 기능 자체가 변조되지 않도록 보호되어야 하며, 올바르게 작동하는지 검증할 수 있어야 한다. 접근 제어는 접근을 허용하거나 제한하는 기능이고, 보안 커널은 그 기능을 시스템 내부에서 집행하는 구현 요소라는 차이가 있다.

따라서 (가)는 운영체제 내부에 통합되어 시스템 요소 사이의 접근을 통제하는 보안 커널이며, ①에 대응한다.

NoteBase

문 4. 보안 모델의 하나인 만리장성(Chinese Wall) 모델에 대한 설명으로 옳은 것은?

① 높은 보안 수준에서 낮은 보안 수준으로 정보가 흐르는 것을 방지하기 위한 기밀성 보장 모델이다.

② 비즈니스 입장에서 직무 분리 개념을 적용하여 이해가 충돌하는 회사 간에 정보의 흐름이 일어나지 않도록 접근을 통제한다.

③ 무결성 중심의 상업적 모델로 내·외부의 일관성을 유지하고 비인가자에 의한 불법적인 수정을 방지한다.

④ 무결성을 위한 모델로 비인가자의 데이터 변형을 방지하기 위한 것이다.

정답 2번

간단한 해설

만리장성 모델은 **이해가 충돌하는 회사 사이의 정보 유출**을 막도록 접근을 통제한다. 한 회사의 비밀정보를 열람한 사용자가 경쟁 회사의 비밀정보까지 열람하지 못하게 하므로, **이전 접근 이력**에 따라 접근 가능 범위가 달라진다.

선지별 해설

① X: 높은 보안 수준의 정보가 낮은 보안 수준으로 흘러 기밀이 유출되는 것을 막는 것은 **벨–라파둘라(Bell–LaPadula) 모델**의 설명이다. 만리장성 모델은 보안 등급의 높낮이보다 회사 간 이해충돌 여부와 사용자의 이전 접근 이력을 기준으로 접근을 제한한다.

② O: 만리장성 모델은 비즈니스에서 **이해충돌을 방지하도록 업무와 정보 접근을 분리**하는 모델이다. 경쟁 회사들을 같은 이해충돌 집합으로 묶고, 사용자가 그중 한 회사의 비밀정보에 접근하면 같은 집합의 다른 회사 비밀정보에는 접근하지 못하게 한다. 예를 들어 서로 경쟁하는 A사와 B사가 같은 집합에 속한다면, A사의 비밀자료를 읽은 분석가는 B사의 비밀자료를 읽을 수 없다. 이처럼 이전 접근 이력에 따라 접근을 제한하여 경쟁 회사 사이에 정보가 전달되는 것을 막는다.

③ X: 상업적 환경에서 데이터의 무결성과 내·외부 일관성을 유지하는 것은 **클라크–윌슨(Clark–Wilson) 모델**의 설명이다. 내부 일관성은 시스템 안의 데이터가 정해진 규칙에 맞는 것이고, 외부 일관성은 데이터가 실제 업무 상태와 일치하는 것이다. 이 모델은 검증된 거래 절차를 통해 데이터를 변경하고 직무를 분리하여 부당한 수정을 방지한다. 만리장성 모델의 핵심은 이러한 데이터 변경의 적정성보다 이해충돌 관계에 있는 회사 간 정보 흐름을 차단하는 데 있다.

④ X: 데이터의 부당한 변형을 막는 무결성 중심 모델의 대표적인 예는 **비바(Biba) 모델**이다. 무결성은 데이터가 허가 없이 또는 부적절하게 변경되지 않도록 하는 성질이다. 만리장성 모델은 데이터 수정 방지 자체를 주목적으로 삼는 것이 아니라, 이해충돌을 일으키는 비밀정보 접근과 정보 전달을 제한한다.

NoteBase

문 5. 윈도시스템의 NTFS에서 폴더 및 파일 접근 권한에 대한 설명으로 옳은 것은?

① 그룹 A와 그룹 B에 속한 사용자가 그룹 A에서는 읽기 권한을 할당받고 그룹 B에서는 쓰기 권한을 할당받았다면, 사용자에게 읽기와 쓰기 권한이 모두 주어진다.

② 파일이 포함된 폴더 권한이 파일 권한보다 우선한다.

③ 권한을 중첩해서 적용할 수 있으며, 허용 설정이 거부 설정보다 우선한다.

④ 폴더 접근 권한은 그룹에게만 부여된다.

정답 1번

간단한 해설

NTFS에서는 사용자가 속한 여러 그룹의 **허용 권한이 누적**된다. 적용되는 거부 권한이 없다면 그룹 A의 읽기 권한과 그룹 B의 쓰기 권한을 모두 갖는다.

선지별 해설

① O: NTFS의 접근 권한은 사용자 본인과 소속 그룹에 부여된 권한을 종합하여 결정한다. 여러 그룹에서 부여받은 **허용 권한은 누적**되므로, 별도의 거부 권한이 없는 제시된 상황에서는 그룹 A의 읽기 권한과 그룹 B의 쓰기 권한이 함께 적용된다.

② X: 상위 폴더의 권한이 파일의 권한보다 무조건 우선하는 것은 아니다. 파일은 상위 폴더의 권한을 상속받을 수 있지만, **파일에 직접 설정한 명시적 권한은 상속된 권한보다 우선**한다. 따라서 파일에는 폴더와 다른 접근 권한을 설정할 수 있다.

③ X: 권한을 중첩하여 적용할 수 있다는 부분은 맞지만, 허용 설정이 거부 설정보다 우선한다는 부분이 틀렸다. 같은 수준의 명시적 권한끼리 또는 같은 상속 단계의 권한끼리 충돌하면 **거부가 허용보다 우선**한다. 다만 명시적 권한은 상속된 권한보다 먼저 적용되므로, 파일에 직접 설정한 허용 권한은 상위 폴더에서 상속된 거부 권한보다 우선할 수 있다.

④ X: 폴더 접근 권한은 **개별 사용자와 그룹 모두**에게 부여할 수 있다. 그룹에 권한을 부여하면 그 그룹의 구성원에게 함께 적용되므로 여러 사용자의 권한을 한꺼번에 관리할 수 있지만, 그룹을 통해서만 부여해야 하는 것은 아니다.

NoteBase

문 6. 다음에서 설명하는 전자우편 보안 기술은?

◯ 종단 사용자에게 투명한 인증 기술을 제공하도록 설계되었다.
◯ 전자우편 메시지는 전자우편 발신지 관리 도메인의 개인키에 의해 서명된다.
◯ 서명은 메시지 내용 전체와 메시지 헤더의 일부를 대상으로 한다.
◯ 수신 측의 MDA(Mail Delivery Agent)는 DNS를 통해 해당 공개키에 접근하여 서명을 검증할 수 있다.

① PGP
② PEM
③ MIME
④ DKIM

정답 4번

간단한 해설

발신 도메인의 개인키로 전자우편에 서명하고, 수신 측이 **DNS에서 얻은 공개키로 검증**하는 방식은 DKIM의 특징이다.

공통 해설

DKIM(DomainKeys Identified Mail)은 전자우편 메시지에 도메인의 전자서명을 붙여, 해당 도메인과 메시지의 연결을 검증하는 인증 기술이다. 발신 측은 도메인의 개인키로 서명을 생성하고, 수신 측은 **DNS에 공개된 대응 공개키**를 조회하여 서명을 검증한다. 따라서 여러 전자우편 관련 기술 중에서 도메인 기반 서명과 DNS 공개키 조회를 함께 사용하는 기술을 찾으면 DKIM에 해당한다.

서명에는 메시지 본문의 해시값과 선택한 헤더 필드가 반영된다. 본문 길이를 별도로 제한하지 않으면 본문 전체가 해시 대상이 되므로, 지문의 본문 전체와 헤더 일부를 대상으로 한다는 설명과 연결된다. 수신 측은 본문 해시값과 서명을 확인하여 서명 대상 내용의 변경 여부를 검사할 수 있다. 이 처리를 메일 서버 측에서 수행할 수 있어, 종단 사용자가 직접 서명·검증 작업을 하지 않아도 된다는 의미에서 사용자에게 투명하다.

NoteBase

문 7. 다음에서 설명하는 윈도시스템 인증 구성 요소는?

사용자의 계정과 패스워드가 일치하는 경우 해당 사용자에게 고유의 SID(Security Identifier)를 부여하며, 파일이나 폴더에 대한 접근 허용 여부를 결정하고 이에 대한 감사 메시지를 생성한다.

① SAM

② LSA

③ SRM

④ SPI

정답 3번

간단한 해설

파일·폴더의 **접근 허용 여부를 판단하고 감사 이벤트를 생성**하는 구성 요소는 SRM(Security Reference Monitor, 보안 참조 모니터)이다. 사용자의 신원을 확인한 뒤 실제 자원 사용 권한을 검사하는 기능에 해당한다.

공통 해설

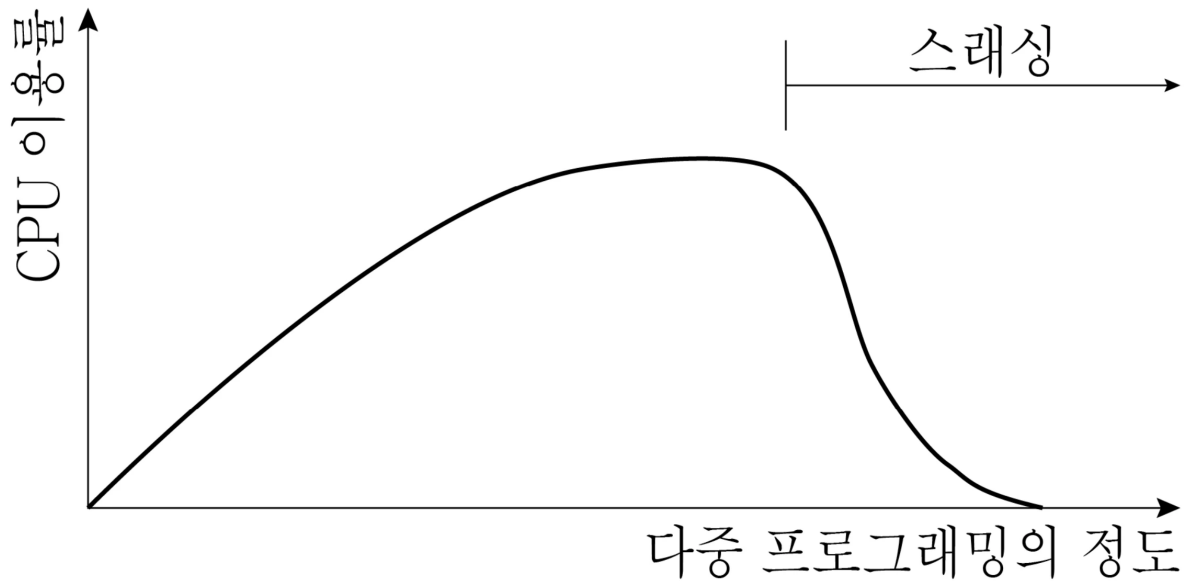
SRM(Security Reference Monitor, 보안 참조 모니터)은 윈도 커널에서 보안 정책을 집행하는 구성 요소이다. 파일이나 폴더에 대한 접근 요청이 들어오면 **요청한 작업을 수행할 권한이 있는지 검사**하고, 감사 정책에 따라 보안 감사 이벤트를 생성한다. 지문의 '접근 허용 여부를 결정'하고 '감사 메시지를 생성'한다는 설명이 이 기능에 해당한다.

SID는 사용자나 그룹을 식별하는 값으로, 인증 후 사용되는 접근 토큰에 포함된다. SRM은 접근 토큰의 사용자·그룹 정보와 대상 자원의 접근 권한 설정을 대조하여 접근을 판단한다. 예를 들어 파일에 읽기 권한만 있는 사용자가 쓰기를 요청하면, 로그인에 성공했더라도 해당 쓰기 요청은 허용되지 않는다. 이처럼 **인증 성공과 개별 자원의 접근 허용은 별개의 판단**이며, 이 문제는 신원 확인 이후의 접근 통제와 감사를 담당하는 구성 요소를 묻는다.

NoteBase

문 8. (가), (나)에 들어갈 용어를 바르게 연결한 것은?

그림과 같이 다중 프로그램의 정도에 따른 CPU 이용률은 처음에는 비례해서 증가하지만 다중 프로그래밍의 정도가 어느 정도 이상으로 커지면 스래싱(thrashing)이 일어나게 되어 CPU 이용률은 급격히 떨어진다. 스래싱은 (가) 페이지 교체 알고리즘을 사용하면 제한할 수 있다. 또한 이 현상을 방지하기 위해서는 각 프로세스에게 할당되는 최소한의 (나) 개수를 보장해야 한다.



① (가): 전역, (나): 프레임

② (가): 전역, (나): 페이지

③ (가): 지역, (나): 프레임

④ (가): 지역, (나): 페이지

정답 3번

간단한 해설

지역 페이지 교체는 다른 프로세스의 프레임을 빼앗지 않아 스래싱이 다른 프로세스로 번지는 것을 제한한다. 스래싱을 방지하려면 각 프로세스가 실행에 필요한 페이지를 유지할 수 있도록 충분한 **프레임**을 보장해야 한다.

NoteBase

빈칸별 해설

스래싱은 페이지 부재가 지나치게 자주 발생하여 실제 명령 실행보다 페이지를 메모리로 가져오고 내보내는 데 많은 시간을 쓰는 현상이다. 그림에서 가로축은 다중 프로그래밍의 정도, 세로축은 CPU 이용률이다. 동시에 실행하는 프로세스가 늘어나면 CPU 이용률이 높아지지만, 메모리가 부족해 스래싱이 발생하는 구간에서는 페이지 입출력 대기가 늘어 CPU 이용률이 급격히 떨어진다.

(가) 지역: 지역 페이지 교체는 **해당 프로세스에 할당된 프레임 안에서만** 교체 대상을 고른다. 따라서 스래싱이 발생한 프로세스가 다른 프로세스의 프레임까지 빼앗아 연쇄적으로 페이지 부재를 늘리는 것을 제한한다. 반면 전역 페이지 교체는 다른 프로세스의 프레임도 교체 대상으로 삼으므로 이러한 영향이 퍼질 수 있다. 다만 지역 교체만으로 해당 프로세스 자체의 프레임 부족까지 해결되는 것은 아니다.

(나) 프레임: 페이지는 가상 메모리를 나눈 단위이고, 프레임은 그 페이지를 담는 **물리 메모리의 공간 단위**이다. 할당된 프레임이 부족하면 곧 다시 참조할 페이지도 내보내야 하므로 페이지 부재가 반복된다. 이를 막으려면 각 프로세스가 현재 실행에 필요한 페이지들을 메모리에 유지할 수 있을 만큼의 프레임을 보장해야 한다.

따라서 (가)는 지역, (나)는 프레임이며, ③의 조합과 일치한다.

NoteBase

문 9. S/MIME에 대한 설명으로 옳지 않은 것은?

① 서명된 데이터(signed data)의 디지털 서명은 메시지 다이제스트를 서명자의 개인키로 암호화한 것으로, 서명과 함께 메시지 내용은 base64로 부호화된다.

② 서명과 봉인된 데이터(signed and enveloped data)는 서명만 하거나 암호화만 한 개체들이 중첩된 것으로, 암호화한 데이터를 서명하거나 서명한 데이터를 암호화할 수 있다.

③ 디지털 서명만을 base64로 부호화한 클리어 서명 데이터(clear-signed data)의 경우, S/MIME 기능을 갖추지 않은 수신자도 서명을 검증하고 메시지의 내용을 볼 수 있도록 지원한다.

④ X.509의 버전 3을 따르는 공개키 인증서를 사용한다.

정답 3번

간단한 해설

클리어 서명은 본문과 서명을 분리하므로 S/MIME 기능이 없어도 **메시지 내용은 읽을 수 있다**. 그러나 **서명 검증에는 S/MIME 처리 기능이 필요**하며, base64 부호화만으로 검증 기능이 제공되지는 않는다.

선지별 해설

① O: 메시지 다이제스트는 메시지에 해시 함수를 적용한 요약값이다. 서명자는 개인키로 서명을 생성하고, 수신자는 서명자의 공개키로 이를 검증한다. '다이제스트를 개인키로 암호화한다'는 표현은 RSA 서명의 원리를 간략히 설명한 것이다. 서명된 데이터 형식은 본문과 서명을 함께 묶어 base64로 부호화한다. 이때 **base64는 전송용 부호화**이므로 메시지의 기밀성을 보장하는 암호화와 구별한다.

② O: 서명된 개체와 암호화된 개체를 **중첩하여 두 기능을 함께 적용**할 수 있다. 먼저 서명한 뒤 그 결과를 암호화하면 본문과 서명이 함께 암호화된다. 반대로 먼저 암호화한 뒤 서명하면 암호문이 서명의 대상이 된다. 따라서 두 적용 순서가 모두 가능하다.

③ X: 클리어 서명은 MIME 본문과 분리된 서명을 별도 부분으로 구성하고, 서명 부분을 base64로 부호화하는 방식이다. 일반 MIME 수신 프로그램은 본문을 표시할 수 있지만, 서명의 유효성을 확인하려면 서명 처리와 공개키를 이용한 검증이 필요하다. 따라서 S/MIME 기능이 없는 수신자도 **본문은 읽을 수 있으나 서명을 검증할 수는 없다**. 선지의 '서명을 검증하고 메시지의 내용을 볼 수 있다'를 '메시지의 내용을 볼 수 있다'로 고친다.

④ O: S/MIME은 **X.509 버전 3 공개키 인증서**를 사용한다. 인증서는 소유자 정보와 공개키를 연결하고 인증기관의 서명으로 그 연결을 확인할 수 있게 한다. 수신자는 인증서의 유효성을 확인한 뒤 서명자의 공개키를 서명 검증에 사용한다.

NoteBase

문 10. HTTP 버전 1.1에서 정의된 요청 메시지의 메소드에 대한 설명으로 옳은 것은?

① GET이 요청하는 웹페이지의 위치는 헤더 라인 안에 명시된다.

② 서버가 보내온 쿠키를 저장했다가 반환하는 GET의 경우, 쿠키 정보는 메시지 몸체(body)에 포함된다.

③ HEAD는 서버로부터 웹페이지 자체가 아닌 웹페이지에 대한 일부 정보를 요청하기 위한 것이다.

④ PUT은 요구 메시지가 서버에 의해 제대로 처리되는가를 검사하기 위한 에코 반환 용도로 사용된다.

정답 3번

간단한 해설

HEAD는 GET과 달리 **응답 몸체 없이 헤더 정보만** 받아 웹페이지의 내용 유형이나 최종 수정 시각 등을 확인하는 메소드이다. 웹페이지 본문을 내려받지 않고도 해당 페이지에 관한 정보를 얻을 수 있다.

선지별 해설

① X: 요청하는 웹페이지의 경로는 헤더 라인이 아니라 **첫 줄인 요청 라인**의 요청 대상에 명시된다. 예를 들어 'GET /index.html HTTP/1.1'에서 '/index.html'이 요청할 페이지의 경로이다. Host 헤더는 대상 서버의 호스트와 필요한 경우 포트를 나타내므로 요청 라인과 구분해야 한다.

② X: 저장한 쿠키를 서버에 다시 보낼 때는 메시지 몸체가 아니라 **요청 헤더의 Cookie 필드**에 넣는다. 서버는 응답의 Set-Cookie 헤더로 쿠키를 전달하고, 클라이언트는 이를 저장한 뒤 전송 조건에 맞는 요청의 Cookie 헤더로 돌려보낸다. GET 요청에서도 같은 방식이다.

③ O: HEAD는 GET과 같은 대상의 정보를 요청하되 **응답에 몸체를 포함하지 않는다**. 따라서 웹페이지 본문 대신 응답 헤더를 통해 페이지의 메타데이터를 확인한다. 예를 들어 Content-Type이 있으면 내용 유형을, Last-Modified가 있으면 최종 수정 시각을 알 수 있다.

④ X: 요청 메시지를 되돌려 받아 상대편에 도달한 내용을 확인하는 에코·진단용 메소드는 **TRACE**이다. PUT은 요청 몸체에 보낸 내용으로 지정한 대상 자원의 상태를 생성하거나 대체하도록 요청하는 메소드이다.

NoteBase

문 11. 리눅스 시스템의 /etc/shadow 파일 내용에서 패스워드의 최종 변경일에 해당하는 것은?

root: 6L9~중략~ruKuT0: 15917: 0: 99999: 7: 5: 16070:

① 15917

② 99999

③ 7

④ 16070

정답 1번

간단한 해설

/etc/shadow에서 패스워드 최종 변경일은 콜론으로 구분한 **세 번째 필드**에 기록된다. 제시된 15917은 1970년 1월 1일을 기준으로 최종 변경일까지 경과한 일수이다.

핵심 풀이

/etc/shadow의 각 행은 콜론(:)으로 필드를 구분한다. 앞부분은 '사용자명:패스워드 해시:최종 변경일' 순서이므로, root와 패스워드 해시 다음에 오는 **15917이 최종 변경일**이다. 이 값은 연월일을 직접 적은 것이 아니라 1970년 1월 1일 00:00 UTC부터 해당 날짜까지 경과한 일수이다.

다른 숫자는 위치와 의미가 다르다. 다섯 번째 필드의 99999는 패스워드를 변경해야 할 때까지의 최대 사용 기간이고, 여섯 번째 필드의 7은 패스워드 만료 며칠 전부터 경고할지를 나타낸다. 여덟 번째 필드의 16070은 계정 만료일이다. 16070도 기준일부터 경과한 일수로 날짜를 나타내지만, **패스워드 최종 변경일과 계정 만료일**은 서로 다른 필드이다.

NoteBase

문 12. PKI에 대한 설명으로 옳지 않은 것은?

① 인증기관(CA), 등록기관(RA), 키 분배 센터(KDC)로 구성된다.

② 이용자는 공개키를 이용하기 전에 CA의 인증서 폐기 목록(CRL)을 조사해서 해당 인증서의 유효성을 확인할 필요가 있다.

③ CA의 공개키에 대해 다른 CA가 디지털 서명을 하는 것으로 그 CA의 공개키를 검증할 수 있다.

④ CA의 개인키가 노출된 경우에는 그 사실을 CRL을 사용해서 이용자에게 통지할 필요가 있다.

정답 1번

간단한 해설

KDC는 **대칭키 기반 인증에서 키를 분배하는 기관**으로, PKI의 기본 구성요소가 아니다. PKI는 CA의 인증서 발급, RA의 등록 심사, 저장소의 인증서·폐기 정보 제공 등을 통해 공개키의 신뢰를 관리한다.

선지별 해설

① X: PKI는 공개키와 그 소유자의 관계를 인증서로 확인하는 기반이다. CA는 인증서를 발급하고, RA는 인증서 신청자의 신원과 등록 정보를 확인하며, 저장소는 인증서와 CRL 등을 제공한다. **KDC는 PKI의 기본 구성요소가 아니라** Kerberos 등 대칭키 기반 인증에서 세션키와 티켓을 제공하는 기관이다.

② O: 인증서는 유효기간이 남아 있어도 개인키 노출 등의 사유로 폐기될 수 있다. 따라서 공개키를 사용하기 전에 CRL을 조회하여 **해당 인증서가 폐기되었는지** 확인할 필요가 있다. CRL에 없다는 사실만으로 모든 유효성 검사가 끝나는 것은 아니며, 인증서의 서명·유효기간·인증 경로 등도 확인한다.

③ O: 다른 CA가 해당 CA의 공개키와 신원 정보를 담은 인증서에 디지털 서명을 하면, 그 인증서를 통해 공개키의 소유 관계를 확인할 수 있다. 이를 **교차 인증**이라고 한다. 다만 서명한 CA까지 이어지는 신뢰할 수 있는 인증 경로가 있어야 그 서명을 신뢰의 근거로 사용할 수 있다.

④ O: CA의 개인키가 노출되면 공격자가 그 CA를 사칭하여 인증서에 서명할 수 있으므로, 관련 CA 인증서를 폐기하고 이용자에게 알려야 한다. 예를 들어 하위 CA의 키가 노출되면 그 인증서를 발급한 상위 CA가 CRL에 폐기 사실을 반영한다. CRL에는 **CA 키 손상(cACompromise)**이라는 폐기 사유가 있다. 다만 신뢰의 출발점인 루트 CA의 키가 노출되면 CRL 통지만으로 충분하지 않으며, 해당 CA를 신뢰기점에서 제거하는 조치가 필요하다.

NoteBase

문 13. 인증을 받은 사용자가 여러 정보 시스템에 재인증 절차 없이 반복해서 접근할 수 있도록 해주는 것은?

① OTP

② SSO

③ Challenge & Response

④ CAPTCHA

정답 2번

간단한 해설

SSO(Single Sign-On)는 **한 번의 인증으로 여러 연계 시스템을 이용**하게 하는 통합 인증 방식이다. 최초 인증 결과를 활용하므로 시스템마다 사용자가 인증 절차를 반복할 필요가 없다.

개념 구분

SSO(Single Sign-On, 싱글 사인온)는 한 시스템에서 확인한 사용자 인증 결과를 다른 연계 시스템에서도 신뢰하여 이용하는 방식이다. 예를 들어 회사 포털에 로그인한 뒤 연계된 메일과 업무 시스템을 각각 다시 로그인하지 않고 이용할 수 있다. 문제의 **여러 정보 시스템에 재인증 없이 접근**한다는 조건이 이에 해당한다.

OTP는 **한 번만 사용하는 비밀번호**로 인증하는 방식이며, 여러 시스템의 로그인을 연계하는 기능 자체를 뜻하지 않는다.

Challenge & Response는 서버가 제시한 요청값에 대해 올바른 응답값을 제시하는지 확인하는 인증 방식이다. 이는 사용자를 검증하는 방법으로, 여러 시스템에서 최초 인증 결과를 활용하는 SSO와 구별된다.

CAPTCHA는 응답자가 **자동화 프로그램인지 사람인지** 구별하기 위한 검사이다. 여러 정보 시스템에 대한 통합 로그인 기능은 아니다.

NoteBase

문 14. 공격 대상이 되는 서버에서 먼저 공격자 PC로 연결하게 하여 방화벽 보안 정책을 우회하는 공격은?

① 리버스 텔넷

② 쿠키 변조

③ 명령 삽입

④ 파일 업로드

정답 1번

간단한 해설

리버스 텔넷은 **공격 대상 서버가 공격자 PC로 먼저 연결**하도록 하는 방식이다. 외부에서 시작하는 접속은 차단하지만 내부에서 시작하는 접속은 허용하는 방화벽 정책을 이용한다.

핵심 풀이

리버스 텔넷: 공격자가 서버로 직접 접속하는 방향을 뒤집어, **서버에서 공격자 PC로 연결을 시작**하게 한다. 이렇게 맺어진 연결을 통해 공격자가 서버에 명령을 보내고 실행 결과를 받을 수 있다. 문제의 '서버에서 먼저 공격자 PC로 연결'이라는 설명이 이 방식에 해당한다.

방화벽 우회 원리: 외부에서 내부로 들어오는 새 연결은 차단하더라도, 내부에서 외부로 시작한 연결과 그에 따른 응답은 허용하는 정책이 있을 수 있다. 리버스 텔넷은 이러한 **내부 발신 연결의 허용**을 이용해 통신 경로를 만든다. 따라서 연결 방향만 바꾸면 모든 방화벽을 우회하는 것은 아니며, 해당 외부 연결이 허용되어야 한다.

다른 공격과의 구분: 쿠키 변조는 쿠키 값을 바꾸어 인증·권한 등의 처리에 영향을 주는 공격이고, 명령 삽입은 입력값에 운영체제 명령을 끼워 넣어 실행시키는 공격이다. 파일 업로드 공격은 업로드 기능의 검증 미비를 이용해 악성 파일 등을 서버에 올리는 공격이다. 명령 삽입이나 파일 업로드가 리버스 텔넷을 실행시키는 수단이 될 수는 있지만, 이 문제에서 묻는 것은 **서버가 외부로 먼저 연결하는 방식**이다.

NoteBase

문 15. (가) ~ (다) 안에 들어갈 용어를 바르게 연결한 것은?

함수 P가 함수 Q를 호출하기 위해, P는 Q에 전달할 인수를 스택에 넣는다. Q를 호출하는 call 명령어를 수행하면 (가)가 스택에 저장된다. Q는 (나)를 스택에 넣는다. 프레임 포인터 레지스터 값을 현재의 포인터 값으로 설정하고, 스택 포인터를 아래로 움직여 (다)를 저장할 공간을 할당하고, Q의 코드를 수행한다. 컴파일러의 최적화 기능에 따라 실제 배치는 차이가 있을 수 있으나, 저장된 (가)와 (나)에 겹쳐 쓰기는 스택 버퍼 오버플로 공격의 핵심이다.

① (가): 반환 주소, (나): P의 스택 프레임 포인터, (다): 지역 변수

② (가): 반환 주소, (나): Q의 스택 프레임 포인터, (다): 지역 변수

③ (가): P의 스택 프레임 포인터, (나): 지역 변수, (다): 반환 주소

④ (가): Q의 스택 프레임 포인터, (나): 지역 변수, (다): 반환 주소

정답 1번

간단한 해설

Q가 자신의 스택 프레임을 만들기 전에는 프레임 포인터 레지스터에 **호출자 P의 프레임 포인터**가 남아 있다. 이를 먼저 보존해야 Q의 실행이 끝난 뒤 P의 프레임을 다시 사용할 수 있다.

NoteBase

빈칸별 해설

지문은 호출자의 프레임 포인터를 보존한 뒤 새 프레임을 설정하는 전형적인 스택 구성을 전제로 한다. 컴파일러 최적화에 따라 실제 배치는 달라질 수 있다.

(가) 반환 주소: call 명령어는 Q의 실행이 끝난 뒤 P에서 실행을 이어 갈 위치를 스택에 저장한다. 이 위치는 **call 명령어 바로 다음 명령어의 주소**이다. 반환 주소는 실행을 재개할 위치이고, 프레임 포인터는 스택 프레임을 참조하는 기준 주소이므로 역할이 다르다.

(나) P의 스택 프레임 포인터: Q에 진입한 직후에는 프레임 포인터 레지스터가 아직 P의 프레임을 가리킨다. Q는 **자신의 프레임 포인터를 설정하기 전에 P의 값을 보존**한다. 이후 Q가 종료될 때 저장해 둔 값을 복원하여 P의 프레임을 다시 참조할 수 있게 한다. 따라서 이 시점에 저장하는 값은 Q의 프레임 포인터가 아니다.

(다) 지역 변수: 지문처럼 스택이 낮은 주소 방향으로 자라는 구성에서는 스택 포인터를 아래로 움직여 Q의 지역 변수를 위한 공간을 확보한다. P가 호출 전에 넣은 인수와 달리, 이 공간은 **Q 내부에서 사용할 지역 변수**를 위한 것이다. 이 영역의 버퍼에 경계를 넘어 쓰면 저장된 프레임 포인터나 반환 주소가 훼손되어 프레임 복원이나 실행 흐름에 영향을 줄 수 있다.

(가)는 반환 주소, (나)는 P의 스택 프레임 포인터, (다)는 지역 변수이므로 ①의 조합과 일치한다.

NoteBase

문 16. FTP에 대한 설명으로 옳지 않은 것은?

① 데이터 연결 시 평문으로 데이터를 전송한다.

② 데이터 연결은 송·수신 모두 지정된 포트 20번을 통해서만 가능하다.

③ 사용자 계정의 패스워드는 암호화되지 않은 상태로 전달된다.

④ FTP를 이용하여 클라이언트는 서버의 파일을 읽고 서버에 파일을 저장할 수 있을 뿐만 아니라 서버의 파일 목록을 볼 수도 있다.

정답 2번

간단한 해설

FTP의 데이터 연결은 **20번 포트로만 제한되지 않는다**. 능동 모드에서는 서버의 기본 데이터 포트가 20번이지만, 수동 모드에서는 서버가 알려 준 별도 포트로 클라이언트가 접속한다.

선지별 해설

① O: 보안 확장을 적용하지 않은 기본 FTP는 데이터 연결에서 파일 내용을 **암호화하지 않고 전송**한다. 따라서 전송 경로에서 패킷을 가로채면 파일 내용이 노출될 수 있다.

② X: FTP는 명령과 응답을 주고받는 제어 연결과 파일 등을 전송하는 데이터 연결을 분리한다. 제어 연결의 서버 기본 포트는 21번이다. 능동 모드에서는 서버가 기본적으로 자신의 20번 포트에서 클라이언트가 지정한 포트로 데이터 연결을 시작한다. 수동 모드에서는 **클라이언트가 서버가 알려 준 별도 데이터 포트로 연결**한다. 따라서 데이터 송·수신이 양쪽 모두 20번 포트로만 이루어진다는 설명은 잘못이다.

③ O: 기본 FTP는 사용자 인증에 쓰는 패스워드를 **제어 연결의 PASS 명령에 평문으로 실어 전송**한다. 로그인 과정이 있다는 사실이 전송 중 패스워드의 암호화를 의미하지는 않는다.

④ O: FTP는 원격 파일 전송과 목록 조회를 지원한다. 필요한 접근 권한이 있으면 클라이언트는 RETR 명령으로 서버의 파일을 내려받고, STOR 명령으로 서버에 파일을 저장하며, LIST 명령으로 파일 목록을 조회할 수 있다.

NoteBase

문 17. SQL 뷰에 대한 설명으로 옳지 않은 것은?

① 사용자가 뷰를 통해서만 데이터에 접근하게 함으로써 기본 테이블에 대한 보안성을 높일 수 있다.

② 뷰가 정의된 기본 테이블이 확장되거나 뷰가 속해 있는 데이터베이스에 테이블이 늘어난다고 하더라도 기존의 뷰를 사용하는 프로그램이나 사용자는 영향을 받지 않는다.

③ 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 사용자 권한 관리가 용이하다.

④ 대부분의 경우 삽입, 삭제, 갱신 연산에 많은 제한이 따르며 질의문이 복잡해지는 단점이 있다.

정답 4번

간단한 해설

④의 '질의문이 복잡해진다'는 설명이 잘못되었다. 뷰는 복잡한 조회 내용을 미리 정의하여 **사용자가 작성하는 질의문을 단순하게** 한다. 삽입·삭제·갱신의 제약은 뷰의 정의에 따라 발생할 수 있다.

선지별 해설

① O: 뷰는 기본 테이블을 조회하는 질의로 정의되는 가상 테이블이다. 사용자에게 기본 테이블의 직접 접근 권한 대신 **허용된 행과 열만 보여 주는 뷰의 접근 권한**을 부여하면 민감한 데이터의 노출을 줄일 수 있다.

② O: 뷰는 기본 테이블의 구조와 사용자 프로그램 사이에 일정한 조회 형태를 제공하므로 **논리적 데이터 독립성**을 높인다. 예를 들어 뷰에서 사용하는 열을 그대로 둔 채 기본 테이블에 다른 열을 추가하거나 별도의 테이블을 추가해도 기존 뷰를 사용하는 프로그램은 그대로 동작할 수 있다. 다만 뷰가 참조하는 열을 삭제하는 등 뷰 정의를 깨뜨리는 변경까지 영향이 없다는 뜻은 아니다.

③ O: 사용자나 업무에 필요한 데이터만 뷰로 정의하고 **뷰 단위로 권한을 부여**하면 접근 범위를 관리하기 쉽다. 예를 들어 직원 테이블에서 급여 열을 제외한 뷰를 만들어 일반 사용자에게 그 뷰의 조회 권한만 줄 수 있다.

④ X: 집계나 그룹화 등을 포함한 뷰는 뷰의 변경을 기본 테이블의 어느 행에 반영할지 명확하지 않아 삽입·삭제·갱신이 제한될 수 있다. 반면 일정한 조건을 충족하는 단순 뷰에서는 이러한 연산이 가능하다. 질의문에 관해서는 복잡한 조인이나 조건을 뷰 정의에 담아 두고 뷰 이름으로 조회하므로, 사용자 질의문이 **복잡해지는 것이 아니라 단순해진다**. 예를 들어 여러 테이블의 조인 결과를 뷰로 정의하면 사용자는 조회할 때마다 그 조인 조건을 다시 작성할 필요가 없다.

NoteBase

문 18. 리눅스에서 제공하는 특수 권한에 대한 설명으로 옳지 않은 것은?

① 숫자로 나타내면 접근 권한의 맨 앞자리에 Set-UID는 4, Set-GID는 2, Sticky-Bit는 1로 표현된다.

② Set-UID를 설정하면 소유자 실행 권한 자리에, Set-GID를 설정하면 그룹 실행 권한 자리에 s 혹은 S가 표시된다.

③ Sticky-Bit는 공유를 목적으로 파일에 설정하는 특수 권한으로, 설정 시 소유자 실행 권한 자리에 t 혹은 T가 표시된다.

④ Set-UID가 설정된 파일이 실행되는 동안에는 파일을 실행한 사용자의 권한이 아니라 파일 소유자의 권한이 적용된다.

정답 3번

간단한 해설

③은 Sticky-Bit의 설정 대상과 표시 위치를 잘못 설명했다. 리눅스에서는 주로 **공유 디렉터리 안의 파일 삭제·이름 변경을 제한**하는 데 사용하며, t 또는 T는 **기타 사용자 실행 권한 자리**에 표시된다.

선지별 해설

① O: 권한을 네 자리 8진수로 표시할 때 맨 앞자리는 특수 권한을 나타낸다. **Set-UID는 4, Set-GID는 2, Sticky-Bit는 1**이며, 여러 특수 권한을 함께 설정하면 해당 값을 더한다. 예를 들어 4755는 일반 권한 755에 Set-UID를 추가한 표시이다.

② O: Set-UID는 소유자 실행 권한 자리, Set-GID는 그룹 실행 권한 자리에 표시된다. 해당 특수 권한과 실행 권한이 모두 있으면 **소문자 s**, 특수 권한은 있으나 해당 실행 권한이 없으면 **대문자 S**로 나타낸다.

③ X: Sticky-Bit는 리눅스에서 주로 공유 디렉터리에 설정하여, 그 안의 파일을 다른 사용자가 함부로 삭제하거나 이름을 바꾸지 못하게 한다. 일반 사용자는 해당 파일이나 디렉터리의 소유자여야 이러한 작업을 할 수 있으며, root와 같은 특권 사용자도 작업할 수 있다. 일반 파일에 공유 권한을 부여하는 기능이 아니다. 표시 위치도 소유자가 아니라 **기타 사용자 실행 권한 자리**이다. Sticky-Bit가 설정된 상태에서 기타 사용자 실행 권한이 있으면 t, 없으면 T로 표시된다. 대표적으로 /tmp와 같은 공유 디렉터리에 사용된다.

④ O: Set-UID가 적용되는 실행 파일을 실행하면 프로세스의 **유효 사용자 ID가 파일 소유자의 사용자 ID**로 설정된다. 따라서 파일 소유자의 권한으로 동작할 수 있다. 실행한 사용자를 나타내는 실제 사용자 ID가 바뀌는 것은 아니므로, 이를 실행자의 신원 자체가 파일 소유자로 바뀌는 것으로 이해해서는 안 된다.

NoteBase

문 19. 다음에서 설명하는 웹 취약점 점검 방법과 해당 취약점을 바르게 연결한 것은?

(가) “../”를 이용해서 임의의 경로가 포함된 값으로 웹페이지 파라미터를 변조한 후 해당 경로의 파일 내용이 표시되는지 확인
(나) 사용자 입력값을 전달받는 게시판, 자료실 등에 <script>alert()</script>와 같은 스크립트를 입력한 후 실행 여부 확인
(다) 인증 후 정상적으로 세션이 발행된 페이지의 정보를 취득하고 일정 시간 후에 재전송했을 때 정상 처리가 되는지 확인

① (가): 경로 추적, (나): CSRF, (다): 불충분한 인증 및 인가
② (가): 경로 추적, (나): XSS, (다): 불충분한 세션 관리
③ (가): 디렉터리 인덱싱, (나): XSS, (다): 불충분한 인증 및 인가
④ (가): 디렉터리 인덱싱, (나): CSRF, (다): 불충분한 세션 관리

정답 2번

간단한 해설

(다)는 인증에 성공한 뒤 발급된 세션을 시간이 지난 후에도 재사용할 수 있는지 점검한다. 핵심은 최초의 신원 확인이 아니라 **세션의 유효기간과 만료 처리**이므로 불충분한 세션 관리에 해당한다.

NoteBase

대응 관계 해설

(가) 경로 추적: '../'는 상위 디렉터리로 이동하는 경로 표현이다. 파일 경로를 받는 파라미터에 이를 넣어 변조한 뒤 허용 범위 밖의 파일 내용까지 읽을 수 있는지 확인하므로, **경로를 조작하여 파일에 접근하는** 취약점을 점검한다. 디렉터리 인덱싱은 웹 서버가 디렉터리 안의 파일·폴더 목록을 노출하는 것으로, 경로를 변조하여 특정 파일의 내용을 읽는 점검과 구별된다.

(나) XSS: 사용자 입력에 포함된 스크립트가 웹페이지에 반영되어 **브라우저에서 코드로 실행되는지** 확인한다. XSS는 신뢰할 수 없는 입력이 실행 가능한 스크립트로 처리되는 취약점이다. 반면 CSRF는 인증된 사용자가 의도하지 않은 요청을 보내도록 유도하여 그 사용자의 권한으로 작업이 처리되게 하는 공격으로, 입력한 스크립트 자체의 실행 여부를 확인하는 점검과 구별된다.

(다) 불충분한 세션 관리: 세션은 인증된 사용자의 상태를 여러 요청에 걸쳐 유지하는 수단이다. 인증 후 발급된 세션을 이용한 요청을 일정 시간 뒤 재전송하는 것은 세션의 유효기간과 만료 처리가 적절한지 확인하는 점검이다. **만료되어야 할 시간이 지난 뒤에도 기존 세션으로 요청이 처리된다면** 세션 관리가 불충분하다. 불충분한 인증 및 인가는 사용자의 신원 확인이나 자원·기능별 접근 권한 검사가 제대로 이루어지는지에 초점을 둔다.

(가)는 경로 추적, (나)는 XSS, (다)는 불충분한 세션 관리로 연결되며, 이 대응은 ②와 일치한다.

NoteBase

문 20. 보안 요구 조건을 명세화하고 평가 기준을 정의하기 위한 ISO 표준인 공통 기준(CC)에서는 요구 조건을 기능적 요구 조건과 보증 요구 조건으로 나누고 있다. 기능적 요구 조건에 해당하지 않는 것은?

① 식별과 인증

② 암호 지원

③ 보안 감사

④ 취약점 평가

정답 4번

간단한 해설

취약점 평가는 제품이 제공하는 보안 기능이 아니라, 공격에 악용될 수 있는 약점이 있는지 확인하는 **보증 요구 조건**에 속한다.

개념 구분

기능적 요구 조건: 제품이 **어떤 보안 기능을 제공해야 하는지**를 규정한다. 식별과 인증(FIA)은 사용자를 구별하고 주장한 신원이 맞는지 확인하는 기능, 암호 지원(FCS)은 암호키 관리와 암호 연산 기능, 보안 감사(FAU)는 보안 관련 사건을 기록하고 감사 기록을 검토·보호하는 기능에 관한 요구 조건이다.

보증 요구 조건: 보안 기능이 요구대로 구현되어 신뢰할 수 있는지를 확인하기 위한 요구 조건이다. 취약점 평가(AVA)는 제품의 개발·운용 과정에서 생긴 약점이 **공격에 악용될 수 있는지**를 분석하고 침투 시험 등으로 확인한다. 따라서 취약점 평가는 기능적 요구 조건이 아닌 보증 요구 조건으로 분류한다. 보안 감사가 제품 자체의 기록·검토 기능을 다루는 데 비해, 취약점 평가는 제품의 안전성에 대한 신뢰를 확보하는 평가 활동을 다룬다는 차이가 있다.

NoteBase