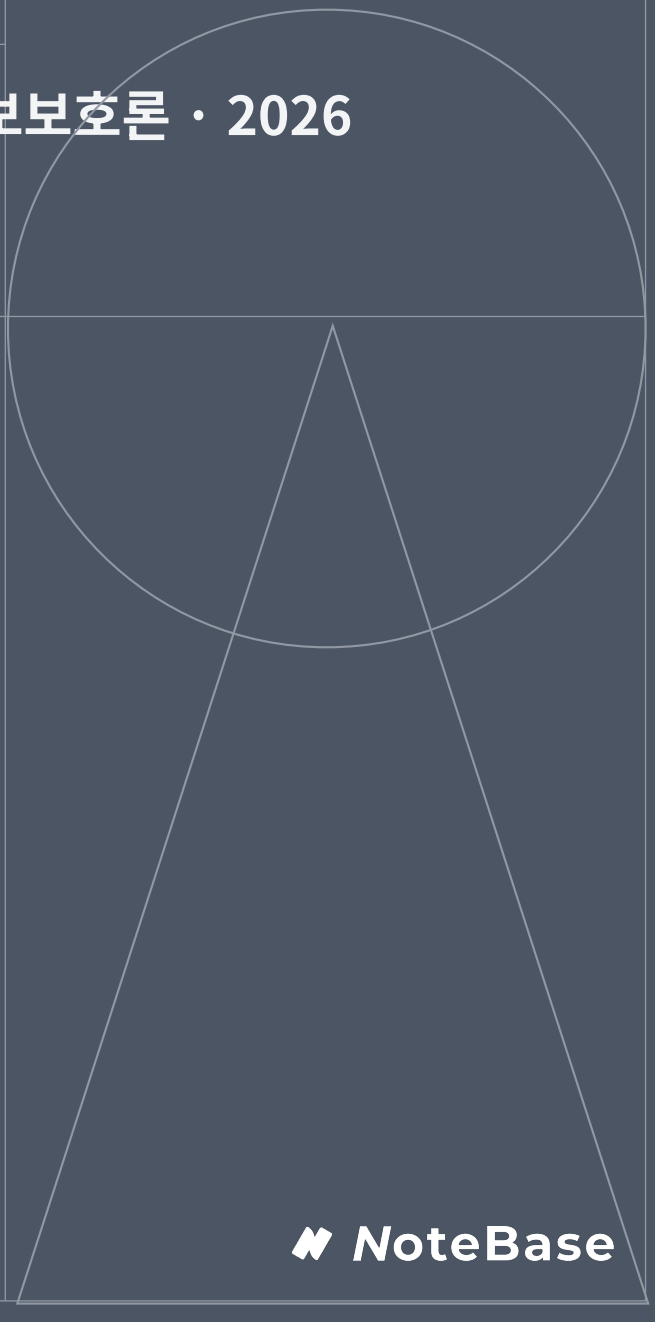
기출OX 해설집

공무원/경찰/경력채용 · 정보보호론 · 2026



NoteBase

기출OX 해설집

공무원/경찰/경력채용 · 정보보호론 · 2026

공무원/경찰/경력채용 · 정보보호론 · 2026

문 1. 다음 암호화 기술과 관련하여 ㉠~㉤에 들어갈 용어로 가장 적절한 것은?

· 앨리스가 밥에게 안전하게 메시지를 보내고자 한다.
· 메시지의 (㉠)을 보장하기 위하여, 앨리스는 밥의 (㉡)로 메시지를 암호화하여 전송한다. 그러면 공격자가 중간에 암호문을 탈취하더라도 공격자는 밥의 (㉢)를 가지고 있지 않기 때문에 암호문을 열어볼 수 없고, 결국 밥만 열어볼 수 있다.
· 발신자가 송신한 사실을 부인하지 못하게 하기 위하여, 앨리스의 (㉣)로 문서를 암호화하여 전송하도록 한다. 이후 수신측에서 해당 암호문이 앨리스의 (㉤)로 잘 복호화되면, 그것은 결국 앨리스가 그 문서를 작성했다는 의미가 된다.

① ㉠: 기밀성, ㉡: 공개키, ㉢: 개인키, ㉣: 개인키, ㉤: 비밀키
② ㉠: 무결성, ㉡: 개인키, ㉢: 공개키, ㉣: 비밀키, ㉤: 공개키
③ ㉠: 기밀성, ㉡: 공개키, ㉢: 개인키, ㉣: 개인키, ㉤: 공개키
④ ㉠: 무결성, ㉡: 개인키, ㉢: 공개키, ㉣: 공개키, ㉤: 비밀키

정답 3번

간단한 해설

기밀성 보호에는 **수신자의 공개키로 암호화하고 개인키로 복호화**한다. 전자서명은 발신자의 개인키로 생성하고 발신자의 공개키로 검증한다.

NoteBase

빈칸별 해설

㉠ 기밀성: 공격자가 암호문을 가로채더라도 내용을 읽지 못하고 수신자인 밥만 읽게 하려는 목적이다. 데이터의 변경 여부를 확인하는 무결성과 구별된다.

㉡ 공개키: 앨리스는 **수신자인 밥의 공개키**로 암호화한다. 공개키는 다른 사람에게 알려 줄 수 있으므로 앨리스가 이를 사용해 밥에게 보낼 암호문을 만들 수 있다.

㉢ 개인키: 밥은 자신의 공개키와 짝을 이루는 개인키로 복호화한다. 이 개인키를 밥이 안전하게 비밀로 유지하므로, 공격자는 공개키나 암호문을 알아도 메시지를 쉽게 읽을 수 없다.

㉣ 개인키: 발신 사실의 부인을 방지하려는 부분은 전자서명의 역할을 설명한다. 서명은 **발신자인 앨리스의 개인키**로 생성하여 해당 개인키 보유자와 문서를 연결한다.

㉤ 공개키: 수신자는 앨리스의 공개키로 서명을 검증한다. 이때 그 공개키가 실제 앨리스의 것인지 확인되어야 하며, 앨리스의 개인키도 안전하게 관리되어야 한다. 두 당사자가 공유하는 대칭암호의 비밀키를 사용하는 과정이 아니다.

지문의 개인키로 문서를 암호화하고 공개키로 복호화한다는 표현은 서명 생성과 검증을 단순화한 설명이다. 모든 전자서명 알고리즘이 문서 전체를 개인키로 암호화하는 것은 아니며, 서명 자체가 문서의 기밀성을 제공하지도 않는다.

따라서 ㉠~㉤은 기밀성, 공개키, 개인키, 개인키, 공개키이며, 이에 해당하는 선택지는 ③이다.

NoteBase

문 2. 패스워드 솔팅(salting) 기법에 관한 설명으로 가장 적절하지 않은 것은?

① 솔트(salt) 값은 동일한 사용자에 대해 패스워드 인증을 수행할 때마다 변경되는 값이다.

② 솔팅(salting)은 동일한 패스워드에 대해 해시된 패스워드가 중복되는 것을 방지한다.

③ 솔트(salt) 값은 패스워드와 함께 해시함수에 입력되어 해시된 패스워드를 구한다.

④ 해시된 패스워드를 저장한 파일을 탈취하면 해당 파일에 있는 사용자의 솔트(salt) 값을 알 수 있다.

정답 1번

간단한 해설

정답은 ①이다. 패스워드 인증에서는 **저장된 솔트를 다시 사용**하여 해시를 비교하므로, 솔트가 인증할 때마다 바뀐다는 설명이 부적절하다.

선지별 해설

① (X) 솔트는 패스워드의 해시를 새로 저장할 때 생성하여 함께 보관하는 값이다. 일반적인 로그인 검증에서는 **저장된 솔트를 다시 사용**해 입력 패스워드의 해시를 계산하고 기존 해시와 비교한다. 따라서 “인증을 수행할 때마다 변경”이 아니라 “저장된 해시를 검증할 때 같은 솔트를 사용”으로 고쳐야 하므로 정답은 ①이다.

② (O) 같은 패스워드라도 저장 항목마다 **서로 다른 솔트**를 결합하면 해시 입력이 달라진다. 이로써 같은 패스워드가 같은 저장 해시로 나타나는 현상을 막고, 미리 계산한 해시 목록을 여러 사용자에게 공통으로 적용하기 어렵게 한다. 솔팅이 수학적으로 모든 해시 충돌을 없앤다는 뜻은 아니다.

③ (O) 패스워드 해싱 과정은 **패스워드와 솔트를 함께 입력**으로 사용하여 저장용 해시를 만든다. 로그인 검증에서도 입력한 패스워드에 저장된 솔트와 동일한 알고리즘·비용 설정을 적용해야 비교할 수 있다.

④ (O) 일반적인 저장 구조에서는 **솔트와 해시를 함께 보관**하므로 해당 파일을 탈취한 공격자는 솔트도 알 수 있다. 솔트는 비밀 키가 아니며, 알려져도 사용자별 추측 계산을 따로 하게 만드는 효과가 있다. 솔트를 안다는 사실만으로 원래 패스워드가 바로 드러나는 것은 아니다.

NoteBase

문 3. 해시함수의 저항성에 관한 <보기 1>의 용어와 <보기 2>의 설명을 가장 적절하게 연결한 것은?

<보기 1>
(가) 역상 저항성
(나) 두 번째 역상 저항성
(다) 충돌 저항성

<보기 2>
㉠ h(x)=h(x')을 만족하는 임의의 두 입력값 x, x'을 찾는 것이 계산적으로 불가능하거나 적어도 무척 어려워야 한다.
㉡ 주어진 임의의 출력값 y에 대해 y=h(x)를 만족하는 입력값 x를 찾는 것이 계산적으로 불가능하거나 적어도 무척 어려워야 한다.
㉢ 주어진 입력값 x에 대해 h(x)=h(x'), x≠x'을 만족하는 다른 입력값 x'을 찾는 것이 계산적으로 불가능하거나 적어도 무척 어려워야 한다.

① (가): ㉠, (나): ㉡, (다): ㉢
② (가): ㉠, (나): ㉢, (다): ㉡
③ (가): ㉡, (나): ㉠, (다): ㉢
④ (가): ㉡, (나): ㉢, (다): ㉠

정답 4번

간단한 해설

역상 저항성은 **주어진 출력**에 맞는 입력을, 두 번째 역상 저항성은 **주어진 입력**과 같은 해시값을 갖는 다른 입력을 찾는 어려움이다. 충돌 저항성은 두 입력을 자유롭게 선택하되, 서로 다른 입력이 같은 해시값을 갖도록 하는 쌍을 찾는 어려움이다.

NoteBase

대응 관계 해설

(가) 역상 저항성 → ㉡. **출력값이 먼저 주어진다**는 점이 기준이다. 주어진 해시값 y에 대하여 $h(x) = y$를 만족하는 입력 x를 찾기 어려워야 한다. 출력에서 그 출력의 원인이 되는 입력을 역으로 찾는 문제이다.

(나) 두 번째 역상 저항성 → ㉢. **첫 번째 입력이 먼저 주어진다**는 점이 기준이다. 주어진 x와 다르면서 같은 해시값을 갖는 x', 즉 $h(x) = h(x')$이고 $x \neq x'$인 입력을 찾기 어려워야 한다. 이미 정해진 입력을 대신할 다른 입력을 찾는 문제이다.

(다) 충돌 저항성 → ㉠. 특정 입력이나 출력이 미리 고정되지 않은 상태에서 **같은 해시값을 갖는 서로 다른 두 입력**을 찾기 어려워야 한다. ㉠ 원문은 '두 입력값'이라고만 표현하지만, 충돌의 정의에는 $x \neq x'$라는 조건이 반드시 필요하다. 같은 입력을 두 번 사용하여 같은 해시값을 얻는 것은 충돌이 아니다.

따라서 연결은 **(가) ㉡, (나) ㉢, (다) ㉠**이므로 정답은 ④이다.

NoteBase

문 4. 송신자가 자신의 개인키로 메시지 다이제스트에 전자서명하여 수신자에게 보내는 과정에서 확보되는 보안 특성으로 가장 적절하지 않은 것은?

① 인증

② 기밀성

③ 무결성

④ 송신자 부인봉쇄

정답 2번

간단한 해설

전자서명은 인증·무결성·송신자 부인봉쇄를 제공하지만 **메시지 내용을 숨기지는 않는다**. 별도의 암호화가 없는 서명만으로 확보되지 않는 특성은 ② 기밀성이다.

개념 구분

전자서명은 송신자가 **개인키로 생성하고 대응하는 공개키로 검증**한다. 메시지 다이제스트에 대한 서명이 유효한지 확인하면 해당 메시지와 서명자의 연결 및 변경 여부를 검증할 수 있다.

① 인증은 공개키가 실제 송신자의 것임을 확인한 상태에서, 대응하는 개인키로 생성된 서명인지를 검증하는 기능이다.

③ 무결성은 수신 메시지의 다이제스트에 대해 서명을 검증하여 **무단 변경을 탐지**하는 기능이다.

④ 송신자 부인봉쇄는 유효한 서명을 제삼자에게 제시하여 송신자가 서명했다는 증거로 활용하는 기능이다. 이러한 보장은 안전한 알고리즘과 개인키 보호, 공개키와 서명자의 올바른 연결을 전제로 한다.

② 기밀성은 허가받지 않은 사람이 메시지 내용을 알 수 없도록 하는 성질이다. **다이제스트에 서명해도 메시지 내용은 숨겨지지 않는다**. 기밀성을 확보하려면 메시지에 별도의 암호화 등을 적용해야 하므로, 전자서명만으로 확보되는 특성이 아닌 것은 ②이다.

NoteBase

문 5. AES 암호 알고리즘에 관한 설명으로 가장 적절한 것은?

① 128비트 평문을 128비트 암호문으로 출력하는 Feistel 알고리즘이다.

② 12, 14, 16 라운드를 사용하며 라운드 횟수에 따라 키의 길이가 달라진다.

③ 128, 192, 256비트 키의 길이에 따라 각 라운드에 사용되는 라운드 키의 길이가 달라진다.

④ 라운드 시작 전에 AddRoundKey 연산을 수행하고, 마지막 라운드에서는 MixColumns 연산을 수행하지 않는다.

정답 4번

간단한 해설

AES는 첫 라운드 전에 AddRoundKey를 적용하고, **마지막 라운드의 MixColumns는 생략**한다. 키 길이에 따라 라운드 수는 달라지지만, 블록과 각 라운드 키의 크기는 모두 128비트이다.

선지별 해설

① X: AES의 평문·암호문 블록 크기가 각각 128비트인 것은 맞지만, 구조는 Feistel이 아니라 **치환·순열 네트워크(SPN)** 계열이다. 상태 전체에 바이트 치환, 행 이동, 열 혼합 등을 반복 적용하며, 블록을 좌우로 나누어 한쪽을 처리하는 전형적인 Feistel 구조와 구별된다.

② X: AES의 키 길이가 128·192·256비트일 때 라운드 수는 각각 **10·12·14회**이다. 선지의 12·14·16회가 틀렸으며, 키 길이에 따라 정해진 라운드 수를 적용한다.

③ X: 원래 키가 128·192·256비트 중 어느 길이든 AddRoundKey에 사용하는 **각 라운드 키는 128비트**이다. 128비트 상태와 비트별 XOR을 해야 하기 때문이다. 원래 키 길이에 따라 달라지는 것은 키 확장 과정과 필요한 라운드 키의 총개수이지, 각 라운드 키의 크기가 아니다.

④ O: 첫 번째 라운드 전에 초기 AddRoundKey로 상태와 키를 XOR한다. 이후 일반 라운드는 SubBytes → ShiftRows → MixColumns → AddRoundKey를 수행하지만, **마지막 라운드에서는 MixColumns를 생략**하여 SubBytes → ShiftRows → AddRoundKey를 수행한다.

NoteBase

문 6. 타원 곡선 암호 알고리즘에 관한 설명으로 가장 적절한 것은?

① 타원 곡선 상의 점 G와 G를 x배한 xG가 주어진다.

② xG를 구하는 연산은 상수 x값을 G점의 각 좌표값들에 곱하는 연산을 의미한다.

③ 타원 곡선 암호 시스템은 타원 곡선 자체를 구하기 어려움을 이용하는 기법이다.

④ p가 소수일 때, GF(p) 상의 타원 곡선은 $y^2+xy=x^3+ax^2+b$ 형태를 갖는다.

정답 1번

간단한 해설

타원 곡선 암호는 공개된 점 G와 xG에서 **비밀 정수 x를 구하기 어려운 성질**을 이용한다. 스칼라배는 좌표별 곱셈이 아니라 곡선의 점 덧셈을 반복하는 연산이다.

선지별 해설

① (O) 기준점 G와 스칼라배한 점 $Q = xG$가 주어졌을 때, 비밀 정수 x를 구하는 것이 타원 곡선 이산로그 문제이다. 공개키를 $Q = xG$, 개인키를 x로 두는 기본 관계를 설명한다.

② (X) xG는 좌표마다 x를 곱하는 연산이 아니라 **타원 곡선의 점 덧셈을 반복하는 스칼라배**이다. 예를 들어 $2G = G + G$이며, 이 덧셈은 곡선 위 점들에 정의된 군 연산이다.

③ (X) 곡선의 방정식과 기준점은 공개된 매개변수이다. 보안의 근거는 곡선 자체를 찾기 어렵다는 것이 아니라, 적절한 곡선과 충분한 크기의 부분군에서 **G와 xG로부터 x를 구하기 어렵다**는 데 있다.

④ (X) 소수체와 이진체에서 통상 사용하는 곡선 방정식을 혼동한 설명이다. $p > 3$인 소수체 $GF(p)$의 표준적인 짧은 바이어슈트라스 형식은 $y^2 = x^3 + ax + b$이며, 4a^3+27b^2≡0±od p를 만족해야 한다. 제시된 $y^2 + xy = x^3 + ax^2 + b$는 이진체 $GF(2^m)$의 비초특이 곡선을 설명할 때 쓰는 대표 형식이다. 따라서 이를 소수체 곡선의 일반적인 형식으로 제시한 부분을 고쳐야 한다.

기준점과 그 스칼라배의 관계를 설명한 ①이 가장 적절하다.

NoteBase

문 7. SET(Secure Electronic Transaction) 전자상거래 보안기술에 관한 설명으로 가장 적절하지 않은 것은?

① 전자봉투와 이중서명 기술을 사용한다.

② 사용자의 지불정보는 상점에게 숨기고 주문정보는 은행에게 숨기는 기능을 제공하여 사용자의 프라이버시를 보호한다.

③ 카드사용자는 비밀키를 사용해 지불정보를 암호화하고, 비밀키는 은행이 운영하는 지불 게이트웨이의 공개키로 암호화한다.

④ 카드사용자가 구매정보와 지불정보를 각각 해시한 후, 두 해시값을 합한 것을 다시 해시하여 이를 카드사용자의 공개키로 암호화한다.

정답 4번

간단한 해설

정답은 ④이다. SET의 이중서명은 **카드사용자의 개인 서명키**로 생성하고 공개키로 검증하므로, 공개키로 생성한다는 설명이 부적절하다.

선지별 해설

① (O) SET은 전자봉투로 지불정보와 암호화 키를 보호하고, 이중서명으로 **주문정보와 지불정보를 하나의 거래로 연결**한다. 이중서명은 두 정보의 관련성과 무결성을 확인하는 역할을 하며, 정보의 기밀성은 암호화와 필요한 정보만 전달하는 구조로 보호한다.

② (O) 상점은 주문을 처리하는 데 필요한 주문정보를 보고, 은행 측 지불 게이트웨이는 결제를 위한 지불정보를 본다. **상점에는 카드 지불정보를, 은행에는 주문 상세를 숨기면서** 각 정보가 같은 거래에 속함을 확인할 수 있게 한다. 거래 식별자나 결제 금액 등 필요한 공통 정보까지 모두 감춘다는 뜻은 아니다.

③ (O) 지불정보는 먼저 **대칭키로 암호화**하고, 그 대칭키를 지불 게이트웨이의 공개키로 암호화하여 전자봉투에 담는다. 게이트웨이는 자신의 개인키로 대칭키를 얻은 뒤 지불정보를 복호화한다. 여기서 선지의 “비밀키”는 지불정보 암호화용 대칭키를 뜻한다.

④ (X) 이중서명은 주문정보와 지불정보의 해시값을 각각 구하고, 두 해시값을 이어 붙인 결과를 다시 해시한 뒤 **카드사용자의 개인 서명키로 서명**하여 만든다. 카드사용자의 공개키는 서명 검증에 사용한다. 따라서 “공개키로 암호화”를 “개인 서명키로 서명”으로 고쳐야 하므로 정답은 ④이다. 이때 두 해시값을 합한다는 것은 산술 덧셈이 아니라 연결을 뜻한다.

NoteBase

문 8. 블록 암호 공격 기법에 관한 설명으로 가장 적절한 것은?

① 차분 해독법은 기지 평문 공격법이다.

② 선형 해독법은 평문 블록들의 비트의 차이를 이용하는 기법이다.

③ 통계적 분석 기법은 암호문 단독 공격법이다.

④ 전수 공격법은 DES보다 AES에 효과적인 기법이다.

정답 3번

간단한 해설

이 문항에서 통계적 분석은 **암호문에 나타나는 분포·빈도를 분석**하는 전통적 방법을 뜻한다. 대응 평문을 확보하지 않고 수행하는 이 분석은 암호문 단독 공격에 해당하며, 통계를 사용하는 모든 암호 해독을 뜻하는 것은 아니다.

선지별 해설

이 문항은 대표적인 기본 공격 분류 중 가장 적절한 설명을 고르는 문제이다. 아래 O/X는 이 기본 분류에 대한 판단이며, 차분 해독의 모든 변형이나 통계 기법 전체를 하나의 공격 환경으로 한정하는 뜻은 아니다.

① X: 차분 해독은 기본적으로 **일정한 차이를 갖도록 선택한 평문 쌍**과 대응 암호문 쌍의 차이가 나타나는 확률을 분석하므로 선택 평문 공격으로 분류한다. 이미 알려진 평문·암호문 쌍을 사용하는 선형 해독의 대표 분류와 구별해야 한다. 다만 충분한 자료에서 필요한 쌍을 확보하는 기지 평문 형태의 차분 공격도 가능하므로, 기지 평문 공격으로는 절대 수행할 수 없다는 뜻은 아니다.

② X: 평문 쌍의 비트 차이와 그 차이가 암호문에 전파되는 양상을 분석하는 것은 차분 해독이다. 선형 해독은 평문·암호문·키의 일부 비트 사이에 성립하는 **선형 근사식의 확률 편향**을 이용한다. 대표적으로 알려진 평문·암호문 쌍을 다수 수집하여 키 정보를 추정한다.

③ O: 여기서 통계적 분석은 **암호문의 분포·출현 빈도**를 관찰하고 예상되는 평문의 통계적 성질과 비교하는 전통적인 분석을 가리킨다. 대응 평문이나 선택한 입력 없이 암호문만 확보하여 수행하는 형태이므로 암호문 단독 공격으로 분류하는 것이 이 문항에서 가장 적절하다. 통계적 추정 자체는 선형·차분 해독에도 쓰이므로, 모든 통계 기법이 암호문 단독 공격이라는 일반 명제로 확대하지 않는다.

④ X: 전수 공격은 가능한 키를 차례로 시험하므로 **키 공간이 작은 DES에 더 유리**하다. DES의 유효 키 길이는 56비트이고 AES는 128·192·256비트이므로, AES의 키 공간이 훨씬 크다. 일반적인 전수 키 탐색에서 AES가 DES보다 더 쉽게 공격된다는 선지의 비교는 반대이다.

NoteBase

문 9. CCM 기술에 관한 설명으로 가장 적절한 것은?

① CTR 운용 모드와 nested MAC을 통합한 기법이다.

② 암호화와 MAC 알고리즘에 동일한 하나의 키가 사용된다.

③ 공개키를 사용하여 기밀성과 사용자 인증을 동시에 제공한다.

④ 주요 알고리즘은 3DES, CTR 운용 모드, CMAC이다.

정답 2번

간단한 해설

CCM은 **CTR과 CBC-MAC을 결합**한 대칭키 기반 인증 암호화 방식이다. 두 연산에 **동일한 하나의 키**를 사용하므로 정답은 ②이다.

선지별 해설

① X — CCM은 **CTR 운용 모드와 CBC-MAC**을 결합한다. CTR은 메시지 암호화로 기밀성을 제공하고, CBC-MAC은 인증 태그 계산에 사용된다. 따라서 nested MAC이라는 부분을 CBC-MAC으로 고쳐야 한다.

② O — CCM에서는 암호화용 CTR 연산과 CBC-MAC 계산에 **동일한 하나의 대칭키**를 사용한다. CCM의 정해진 입력 구성과 연산 절차에 따라 같은 블록 암호와 키를 공유하는 방식이다.

③ X — CCM은 공개키 방식이 아니라 **대칭키 기반 인증 암호화**이다. 기밀성과 함께 데이터의 무결성·메시지 인증을 제공한다. 여기서 인증은 공유키에 기초한 데이터 인증이며, 공개키로 사용자의 신원을 직접 인증한다는 설명은 부적절하다.

④ X — 대표적인 AES-CCM의 구성은 **AES, CTR, CBC-MAC**이다. 3DES는 64비트 블록 암호이므로 128비트 블록 암호를 전제로 하는 CCM의 기반으로 맞지 않고, 인증 방식도 CMAC이 아니라 CBC-MAC이다. CTR이라는 부분만 옳다.

따라서 가장 적절한 설명은 ②이다.

NoteBase

문 10. SQL 삽입(Injection) 보안취약점에 관한 설명으로 가장 적절하지 않은 것은?

① 공격자가 a' or 'a'='a 같은 조건값을 주입하면, 결과적으로 조건절이 'a' or 'a'='a' 으로 바뀌어 참이 되므로, 테이블의 모든 내용이 조회될 수도 있다.

② PreparedStatement 객체 등을 이용하여 DB에 컴파일된 쿼리문을 전달하는 방법을 사용하는 것은 SQL 삽입(Injection) 공격에 대한 보안대책에 해당한다.

③ 사용자가 입력한 <, >, &, ", ' 문자열 등을 replaceAll() 메소드를 사용하여 <, >, &, " 등으로 치환하는 것은 SQL 삽입(Injection) 공격에 대한 보안대책에 해당한다.

④ 외부입력값을 MyBatis 쿼리맵에 바인딩할 경우 $ 기호가 아닌 # 기호를 사용하는 것은 SQL 삽입(Injection) 공격에 대한 보안대책에 해당한다.

정답 3번

간단한 해설

HTML 엔터티 치환은 웹 출력에서 **XSS를 막기 위한 처리**와 관련되며, SQL 코드와 입력 데이터를 분리하는 대책은 아니다. SQL 삽입 방어의 핵심은 **매개변수 바인딩**이다.

선지별 해설

① O. 외부 입력을 SQL 문자열에 그대로 이어 붙이면 입력 내용이 데이터가 아니라 **쿼리의 조건식으로 해석**될 수 있다. 이때 항상 참인 비교식을 OR로 결합하면 원래의 검색 조건이 무력화되어 전체 행이 조회될 수 있다. 다만 선지의 예시 문자열이 모든 DB와 쿼리에서 그대로 실행되는 것은 아니며, 실제 SQL 문법과 따옴표가 맞고 다른 제한 조건이 없는 경우의 위험을 설명한 것이다.

② O. PreparedStatement에 쿼리 구조를 미리 정하고 외부 입력을 **매개변수로 바인딩**하면 SQL 코드와 입력 데이터를 분리할 수 있다. 입력 안의 SQL 기호도 값으로 처리되므로 쿼리 구조를 바꾸기 어렵다. 보안의 핵심은 매개변수 바인딩이며, 외부 입력을 문자열로 결합한 뒤 PreparedStatement 객체에 넣는 것만으로는 안전해지지 않는다.

③ X. 특수문자를 HTML 엔터티로 치환하는 것은 적절한 HTML 출력 문맥에서 브라우저가 입력을 태그나 코드로 해석하지 않도록 하는 **XSS 방어**와 관련된다. replaceAll()을 사용했다는 사실이나 이러한 HTML 치환만으로 SQL 삽입을 방지할 수는 없다. SQL 삽입 방어에는 **매개변수화된 쿼리로 값 바인딩**을 해야 하며, XSS 방어도 실제 출력 문맥에 맞는 인코딩이 필요하다.

④ O. MyBatis의 **#{...}는 PreparedStatement 매개변수 바인딩**으로 값을 전달하지만, ${...}는 문자열을 SQL에 직접 대입한다. 따라서 외부 입력값에는 #{...}를 사용하는 것이 SQL 삽입 방어에 적절하다. 테이블명·열명처럼 값 매개변수로 처리할 수 없는 식별자는 별도의 허용 목록 검증 등이 필요하다.

문 11. crontab 파일은 정기적으로 처리해야 할 작업 목록들을 설정하는 파일이다. 매주 일요일 13시 20분에 동작하는 crontab 파일의 시간 설정으로 가장 적절한 것은?

① 13 20 0 * *

② 13 20 * * 0

③ 20 13 0 * *

④ 20 13 * * 0

정답 4번

간단한 해설

crontab은 **분·시·일·월·요일** 순서로 적는다. 20분, 13시, 날짜와 월 제한 없음, 일요일 0을 적용하면 20 13 * * 0이다.

핵심 풀이

crontab의 다섯 시간 필드는 왼쪽부터 **분 → 시 → 일 → 월 → 요일** 순서이다. 13시 20분이므로 첫째 필드는 20, 둘째 필드는 13으로 설정한다.

매주 일요일에 실행하므로 특정 날짜와 월은 제한하지 않아 셋째·넷째 필드에 각각 *를 넣는다. 마지막 요일 필드의 0은 일요일을 뜻하므로 시간 설정은 20 13 * * 0이 된다.

13 20 * * 0은 일요일 20시 13분을 뜻하므로 시각이 다르다. 13 20 0 * *과 20 13 0 * *은 일요일을 나타내는 0을 셋째 일 필드에 잘못 넣었다. 일 필드의 유효 범위는 1~31이므로 0일은 허용되지 않는다.

요구한 시각과 요일을 모두 만족하는 것은 ④ 20 13 * * 0이다.

NoteBase

문 12. 행정안전부 「소프트웨어 개발보안 가이드 (2021년 11월)」 구현단계 보안약점 제거기준에서 “입력데이터 검증 및 표현” 유형으로 가장 적절하지 않은 것은?

① 메모리 버퍼 오버플로우(Memory Buffer Overflow)

② HTTP 응답분할

③ Private 배열에 Public 데이터 할당

④ 크로스사이트 요청 위조(Cross-Site Request Forgery)

정답 3번

간단한 해설

정답은 ③이다. “Private 배열에 Public 데이터 할당”은 객체의 내부 상태를 외부에서 변경할 수 있게 하는 **캡슐화** 유형의 보안약점이다.

개념 구분

Private 배열에 Public 데이터 할당은 해당 가이드의 **캡슐화** 유형에 속하므로 정답은 ③이다. 외부에서 받은 배열의 참조를 내부의 private 배열에 그대로 저장하면, 외부 코드가 같은 배열을 수정하여 객체의 내부 상태까지 바꿀 수 있다. 입력값의 내용 검증보다 **내부 상태에 대한 외부 접근 차단**이 핵심이며, 필요한 범위의 방어적 복사로 참조 공유를 끊어야 한다.

메모리 버퍼 오버플로우는 입력데이터 검증 및 표현 유형에 포함된다. 데이터의 길이·범위 등을 적절히 제한하지 않아 **할당된 버퍼의 경계 밖에 접근**할 수 있는 문제이다.

HTTP 응답분할도 입력데이터 검증 및 표현 유형이다. 응답 헤더에 반영되는 외부 입력에 CR·LF 같은 **개행문자를 허용**하면 공격자가 응답의 구조를 바꿀 수 있으므로 입력과 헤더 구성을 검증해야 한다.

크로스사이트 요청 위조 역시 이 가이드에서는 입력데이터 검증 및 표현 유형으로 분류한다. 인증된 사용자의 권한을 악용하여 **사용자가 의도하지 않은 요청**을 실행하게 하는 문제이므로, 요청이 정당한 사용자 의사에 따른 것인지 검증하는 것이 중요하다.

NoteBase

문 13. 소수 p가 23이며 원시근 g가 5인 Diffie-Hellman 키 합의 알고리즘에서 송신자의 x값이 6이고 수신자의 y값이 4일 때 분배되는 대칭키로 가장 적절한 것은?

① 2

② 5

③ 11

④ 12

정답 1번

간단한 해설

공유값은 K=g^xy p=5^2423=2이다. 송신자의 공개값 8을 수신자의 비밀값 4로 거듭제곱해 나머지를 구하면 ① **2**가 된다.

계산 과정

주어진 값은 소수 $p = 23$, 원시근 $g = 5$, 송신자의 비밀값 $x = 6$, 수신자의 비밀값 $y = 4$이다.
Diffie-Hellman에서 송신자는 공개값 A=g^x p를 보내고, 수신자는 K=A^y p를 계산한다. 상대편도 같은 방식으로 K=g^xy p를 얻으므로 **두 사람의 공유값은 같다**.

먼저 송신자의 공개값을 계산한다. 5^2=25≡2±od23이므로 5^4≡2^2=4±od23이다. 따라서 A=5^623=(5^4×5^2)23=4×2=8이다.

수신자는 자신의 비밀값 4를 지수로 사용한다. 8^2=64≡18±od23,
8^4≡18^2=324=23×14+2±od23이므로 K=8^423=2이다.

문제에서 대칭키로 지칭하는 공유 비밀값은 **2**이므로 정답은 ①이다. 실제 시스템에서는 보통 이 공유 비밀값에 키 유도 함수를 적용하여 사용할 대칭키를 만든다.

NoteBase

문 14. 생체인증의 오인식률(FAR)과 오거부율(FRR)에 관한 설명으로 가장 적절한 것은?

① 민감도가 높을수록 오인식률(FAR)이 높아진다.

② 오거부율(FRR)이 낮을수록 사용자 편의성이 증가한다.

③ 오인식률(FAR)을 제1종 오류(Type I error)라고 한다.

④ 보안 수준을 높일 경우에는 오거부율(FRR)은 낮아지고 오인식률(FAR)은 높아진다.

정답 2번

간단한 해설

FRR이 낮을수록 **정당한 사용자가 잘못 거부되는 빈도**가 줄어 인증을 다시 시도하는 불편이 감소한다. 같은 시스템의 판정 기준을 엄격하게 조정하면 보통 FAR은 낮아지고 FRR은 높아진다.

선지별 해설

FAR은 타인을 정당한 사용자로 잘못 받아들이는 비율이고, FRR은 정당한 사용자를 잘못 거부하는 비율이다. 이 문항의 민감도와 보안 수준은 같은 시스템에서 본인으로 인정하는 기준을 얼마나 엄격하게 설정하는지를 뜻한다.

① X: 인정 기준을 엄격하게 하면 타인이 통과하기 어려워져 **FAR은 낮아지는 방향**으로 변한다. 반면 정당한 사용자도 기준을 충족하지 못할 가능성이 커져 FRR은 높아지는 경향이 있다. 여기서의 '민감도'를 통계학에서 참양성률을 뜻하는 민감도와 혼동하지 않는다.

② O: FRR이 낮다는 것은 **정당한 사용자가 잘못 거부되는 일이 적다**는 뜻이다. 다시 인증하거나 다른 확인 절차를 거쳐야 하는 불편이 줄어 사용자 편의성이 높아진다. 다만 FRR만으로 타인의 잘못된 수락까지 적다고 판단할 수는 없다.

③ X: 이 문항이 사용하는 생체인증 오류 분류에서는 **FRR이 제1종 오류, FAR이 제2종 오류**이다. '접근자가 정당한 사용자이다'를 귀무가설로 놓으면, 정당한 사용자를 거부하는 것은 참인 귀무가설을 기각하는 제1종 오류이고, 타인을 받아들이는 것은 거짓인 귀무가설을 기각하지 못하는 제2종 오류이다. 가설 설정을 반대로 하면 명칭 대응도 달라질 수 있으므로 FAR·FRR의 실제 의미를 먼저 구별해야 한다.

④ X: 동일한 인증 시스템에서 보안을 위해 인정 기준을 높이면 일반적으로 **FAR은 낮아지고 FRR은 높아진다**. 선지는 두 변화 방향을 뒤집었다. 이는 판정 기준 조정의 상충 관계이며, 센서나 알고리즘 자체를 개선하여 두 오류를 함께 줄이는 경우와는 구별한다.

NoteBase

문 15. 파밍(Pharming) 보안공격과 관련한 설명으로 가장 적절하지 않은 것은?

① 악성코드를 이용해 희생자의 Hosts 파일을 수정하여 가짜 사이트로 유도하는 공격이다.

② 공격자는 Hosts 파일의 특정 도메인명에 대응하는 IP 주소를 위조한 사이트의 IP 주소로 수정한다.

③ 사용자들이 도메인 주소나 URL 주소를 주의해서 보아도 공격을 식별할 수 없다.

④ 사용자 PC는 자신의 Hosts 파일에 찾고자 하는 도메인명이 등록되어 있으면, 도메인명에 대응하는 IP 주소를 Hosts 파일과 DNS를 조회하여 가져온다.

정답 4번

간단한 해설

Hosts 파일에 일치하는 도메인 매핑이 있으면 일반적으로 **그 로컬 정보를 우선 사용**한다. 같은 주소를 얻기 위해 DNS 서버까지 함께 조회해야 하는 것은 아니므로, 변조된 Hosts 정보가 가짜 사이트로 접속을 유도할 수 있다.

선지별 해설

① (O) Hosts 파일을 악성코드로 변조하여 정상 도메인의 접속 목적지를 가짜 사이트로 바꾸는 것은 파밍의 한 방식이다. 사용자가 정상 주소를 입력해도 이름 해석 과정이 조작되어 공격자의 사이트로 연결될 수 있다.

② (O) Hosts 파일에는 도메인명과 IP 주소의 대응 정보가 들어 있다. 공격자가 정상 도메인에 연결된 IP 주소를 가짜 사이트의 주소로 바꾸면, 사용자의 요청이 잘못된 목적지로 향한다.

③ (O) 도메인 문자열은 정상인 채 연결되는 IP 주소가 바뀔 수 있으므로 **주소의 철자만 확인해서는** 이 공격을 식별할 수 없다. 이는 모든 탐지가 불가능하다는 뜻은 아니며, Hosts 파일의 변경이나 인증서 이상 등은 별도로 확인할 수 있다.

④ (X) Hosts 파일의 해당 항목으로 이름을 해석할 수 있다면, 일반적인 이름 해석 과정에서는 **그 로컬 매핑을 우선 사용**하므로 같은 주소를 얻기 위해 DNS 서버까지 함께 조회할 필요가 없다. Windows에서는 Hosts의 매핑을 로컬 해석 캐시에 반영하며, 로컬에서 해결되지 않을 때 DNS 서버에 질의한다. 따라서 Hosts에 일치하는 항목이 있어도 반드시 Hosts와 DNS를 모두 조회한다는 설명은 고쳐야 한다.

Hosts 항목이 일치할 때의 처리 방식을 잘못 설명한 ④가 가장 적절하지 않다.

NoteBase

문 16. User2 소유의 setuid 비트가 설정된 실행 파일을 User1이 실행했을 때 동작하는 프로세스에 관한 설명으로 가장 적절한 것은? (단, 소유자(ruid)와 권한(euid)은 해당 uid를 갖는 사용자 계정명으로 표현한다)

① 프로세스의 소유자(ruid)는 User2이다.

② 프로세스의 소유자(ruid)는 root이다.

③ 프로세스의 권한(euid)은 User2이다.

④ 프로세스의 권한(euid)은 root이다.

정답 3번

간단한 해설

setuid는 실행 시 euid를 **실행 파일 소유자**의 UID로 설정한다. 따라서 이 문항에서는 ruid가 실행자 User1로 유지되고, euid는 파일 소유자 User2가 된다.

선지별 해설

① X. ruid는 실제 사용자 ID로, 이 문항에서는 프로그램을 실행한 **User1**에 해당한다. 실행 파일의 소유자가 User2라는 사실은 setuid 적용 시 euid를 정하는 기준이며, ruid를 User2로 바꾸는 근거가 아니다.

② X. setuid 비트는 실행자의 ruid를 root로 바꾸는 기능이 아니다. User1이 실행한 프로세스의 **ruid는 User1로 유지**된다.

③ O. setuid가 정상적으로 적용되는 실행 파일을 실행하면 프로세스의 euid는 **실행 파일 소유자의 UID**가 된다. 파일 소유자가 User2이므로 euid는 User2이며, 해당 사용자의 권한으로 접근 여부를 판단하게 된다.

④ X. setuid가 설정되었다고 항상 root 권한을 얻는 것은 아니다. euid는 파일 소유자를 따르므로 이 경우에는 **root가 아니라 User2**이다. root 소유 파일에 setuid가 정상적으로 적용되는 경우라야 실행 시 euid가 root가 된다.

NoteBase

문 17. IPSec(IP Security)에 관한 설명으로 가장 적절하지 않은 것은?

① AH 프로토콜은 발신지 호스트를 인증하고 IP 패킷 페이로드의 기밀성을 보장한다.

② 송신자와 수신자 양쪽 모두가 호스트가 아닌 경우에는 터널 모드를 적용한다.

③ AH 프로토콜은 해시함수를 사용하여 메시지 다이제스트를 만들고 이를 AH(인증헤더)에 삽입한다.

④ ESP 프로토콜은 페이로드와 ESP 트레일러를 암호화한 후 ESP 헤더를 추가한다.

정답 1번

간단한 해설

①이 적절하지 않다. AH는 인증과 무결성을 제공하지만 **페이로드를 암호화하지 않아 기밀성을 보장하지 않는다**. 기밀성 보호는 암호화를 사용하는 ESP의 기능과 구분해야 한다.

선지별 해설

① X — AH는 데이터 출처 인증과 무결성 검사를 제공하지만 **암호화를 통한 기밀성은 제공하지 않는다**. 따라서 발신지 인증이라는 부분과 달리, 페이로드의 기밀성을 보장한다는 부분은 틀렸다. 기밀성이 필요하면 암호화 기능을 사용하는 ESP 등을 적용해야 한다.

② O — **보안 게이트웨이 간의 통상적인 배치에서는 터널 모드**를 사용한다. 원래 IP 패킷을 캡슐화하고 게이트웨이 사이의 새 IP 헤더를 붙이는 방식이다. 다만 모든 비호스트 구간에 대한 절대 규칙은 아니다. 표준은 경로상의 중간 시스템 사이를 보호하는 제한적인 전송 모드 구성도 허용한다.

③ O — HMAC 같은 키 기반 해시를 사용하는 경우, 패킷에 대한 **키 기반 무결성 검사값(ICV)**을 계산해 AH의 인증 데이터 필드에 넣는다. 수신자는 이를 다시 계산하여 검증한다. 선지의 메시지 다이제스트는 이 인증값을 가리키는 설명으로 이해해야 하며, 비밀키 없는 단순 해시만으로 발신자를 인증한다는 뜻은 아니다.

④ O — **암호화를 사용하는 ESP**에서는 보호할 페이로드와 ESP 트레일러의 패딩·패딩 길이·다음 헤더 필드를 암호화한다. SPI와 순서번호가 들어 있는 ESP 헤더는 암호화 영역 밖에 붙는다. 이 설명은 일반적인 암호화 적용 구조이며, ESP의 제공 기능은 선택한 알고리즘과 설정에 따라 달라진다.

따라서 AH가 기밀성을 보장한다고 한 ①이 가장 적절하지 않다.

NoteBase

문 18. 데이터베이스 보안기술에 관한 설명으로 가장 적절하지 않은 것은?

① 데이터베이스 사용자 계정의 패스워드는 일방향으로 암호화해야 한다.

② 추론방지는 비인가된 쿼리수행과 응답을 통해 기밀데이터에서 대량의 비기밀데이터를 생성하는 것을 방지하는 데이터베이스 보안 요구사항이다.

③ 뷰(View)는 데이터의 논리적 독립성을 제공하며 자료에 대한 접근제어로 보안성을 제공한다.

④ 집성(Aggregation)은 낮은 보안등급의 정보조각을 조합하여 높은 등급의 정보를 알아내는 데이터베이스 보안위협을 의미한다.

정답 2번

간단한 해설

추론방지는 허용된 질의 결과나 비기밀정보를 이용해 **접근 권한이 없는 기밀정보를 유추**하는 것을 막는 것이다. ②는 정보가 유추되는 방향을 거꾸로 설명하고, 추론의 경로도 비인가 질의로 한정했으므로 적절하지 않다.

선지별 해설

① O: 패스워드는 원문을 복원해 비교하는 대신 **일방향 해시**로 검증할 수 있도록 저장해야 한다. 선지의 일방향 암호화는 이런 비가역적 처리를 가리킨다. 실제 저장에는 사용자별 솔트와 충분한 연산 비용을 갖춘 패스워드 해시 방식을 사용하여 유출된 검증값에 대한 추측 공격을 어렵게 한다.

② X: 추론방지는 **비기밀정보에서 기밀정보를 유추**하는 것을 막는 요구사항이다. 사용자가 조회할 권한이 있는 자료나 통계적 질의 결과라도 서로 비교하면 권한 밖의 정보를 알아낼 수 있으므로, 비인가 질의만이 문제가 되는 것은 아니다. 따라서 선지는 “허용된 질의와 그 응답 등을 통해 비기밀정보에서 기밀정보를 추론하는 것을 방지한다”로 고쳐야 한다. 기밀데이터에서 대량의 비기밀데이터를 생성하는 것을 막는다는 설명은 방향이 반대이므로 정답은 ②이다.

③ O: 뷰는 기본 테이블의 데이터를 질의로 표현한 **논리적 테이블**이다. 기본 테이블의 구조가 바뀌더라도 뷰 정의를 조정하여 사용자에게 보이는 구조를 유지할 수 있어 논리적 데이터 독립성을 지원한다. 또한 필요한 행이나 열만 포함한 뷰에 접근 권한을 부여하면 사용자에게 노출되는 자료를 제한할 수 있어 보안에 활용된다.

④ O: 집성 위협은 개별적으로는 낮은 보안등급인 정보들을 모았을 때 **더 높은 등급의 민감한 정보**가 드러나는 것이다. 각 정보조각의 등급만으로 판단하면 결합된 전체가 갖는 민감도를 놓칠 수 있으므로, 조합 결과에 필요한 보호 수준도 고려해야 한다.

NoteBase

문 19. 법 17(mod 17) 상에서 숫자 4의 곱셈에 대한 역원을 구했을 때, 그 역원에 관한 설명으로 가장 적절한 것은? (단, 역원은 0보다 크고 17보다 작은 숫자이다)

① 2의 배수이다.

② 3의 배수이다.

③ 5의 배수이다.

④ 소수이다.

정답 4번

간단한 해설

곱셈 역원은 4와 곱했을 때 **17로 나눈 나머지가 1**이 되는 수이다. $4 \times 13 = 52 = 17 \times 3 + 1$이므로 조건을 만족하는 역원은 13이며, 이는 소수이다.

계산 과정

1. 역원의 조건: 구할 역원을 u라고 하면 4u≡1±od17이고 0 u17이어야 한다. 즉, **4와 곱한 값을 17로 나눈 나머지가 1**인 정수를 찾는다. 4와 17은 서로소이므로 곱셈 역원이 존재한다.

2. 역원 계산: $17 = 4 \times 4 + 1$이므로 $1 = 17 - 4 \times 4$이다. 따라서 4×(-4)≡1±od17이어서 역원은 -4와 합동이다. 문제에서 **0보다 크고 17보다 작은 대표값**을 요구하므로 $-4 + 17 = 13$을 택한다.

3. 검산과 성질: $4 \times 13 = 52 = 17 \times 3 + 1$이므로 나머지는 1이고, 구한 값 13은 역원의 조건을 만족한다. 13은 2, 3, 5의 배수가 아니다. 또한 √‾(13)4이므로 2와 3으로 나누어떨어지지 않음을 확인하면 13이 소수임을 알 수 있다.

따라서 구한 역원은 13이고 소수이므로 정답은 ④이다.

NoteBase

문 20. 네트워크 기반 공격에 관한 설명으로 가장 적절하지 않은 것은?

① TCP SYN Flooding 공격은 RST 패킷 수신에 의한 연결설정 작업중단을 막기 위해, 사용하지 않는 IP 주소를 SYN 패킷의 출발지 주소로 위조하는 것이 일반적이다.

② ARP Spoofing 공격은 사용자 시스템이 ARP Reply 메시지의 송신자에 대한 인증과정 없이 자신의 ARP 테이블을 갱신하는ARP 프로토콜의 취약점을 이용한다.

③ ARP Spoofing은 호스트의 주소매핑 테이블에 위조된 IP 주소가 설정되도록 하는 공격이다.

④ DNS Spoofing은 먼저 도착한 패킷을 신뢰하고 다음에 도착한 정보는 참조하지 않는 DNS서비스의 특성을 악용한 일종의 시간차 공격이다.

정답 3번

간단한 해설

ARP Spoofing은 대상 IP에 대응하는 **MAC 주소를 잘못 연결**하도록 ARP 캐시를 오염시키는 공격이다. 이를 위조된 IP 주소 설정 자체로 설명한 ③이 가장 적절하지 않다.

선지별 해설

① O: SYN Flooding은 연결이 완성되지 않은 반개방 연결을 많이 남겨 정상 연결에 필요한 자원을 고갈시키는 공격이다. 실제 호스트의 IP 주소를 도용하면 그 호스트가 예상하지 않은 SYN/ACK에 RST로 응답하여 서버의 대기 상태가 해제될 수 있다. 따라서 **RST를 보내지 않는 비응답 IP 주소**를 출발지로 위조하는 것이 전형적인 방식이다. 다만 봇넷 등을 이용하면 출발지 위조 없이도 가능하므로 모든 SYN Flooding의 필수조건은 아니다.

② O: 기본 ARP에는 **송신자의 IP·MAC 대응을 인증하는 절차가 없다**. 호스트가 위조된 ARP Reply의 대응 정보를 받아 ARP 캐시를 갱신하면 공격자가 의도한 MAC 주소로 프레임을 보내게 될 수 있다. 이는 프로토콜의 인증 부재를 이용하는 것이며, 모든 운영체제가 모든 ARP Reply를 조건 없이 받아들인다는 뜻은 아니다.

③ X: ARP는 같은 링크에서 IP 주소에 대응하는 MAC 주소를 알아내는 프로토콜이다. ARP Spoofing의 핵심은 주소 매핑 테이블에 단순히 위조된 IP 주소를 설정하는 것이 아니라, **대상 IP에 공격자의 MAC 주소가 대응하도록** 잘못된 연결 정보를 주입하는 데 있다. 예컨대 게이트웨이 IP에 공격자 MAC이 연결되면 게이트웨이로 보낼 프레임이 공격자에게 향할 수 있다. 따라서 가장 적절하지 않은 것은 ③이다.

④ O: 전통적인 DNS 응답 위조 공격은 **대기 중인 질의와 일치하는 위조 응답**을 진짜 응답보다 먼저 받아들이게 하는 시간 경쟁을 이용한다. 먼저 도착했다는 사실만으로 수용되는 것은 아니며, 질의와 대응하는 출발지·목적지 주소, 포트, 질의 ID, 질의 이름·유형·클래스 등도 맞아야 한다. 이러한 조건을 만족한 응답을 먼저 채택한 뒤 같은 질의의 늦은 응답을 쓰지 않는 처리를 악용한다는 취지이다.

NoteBase