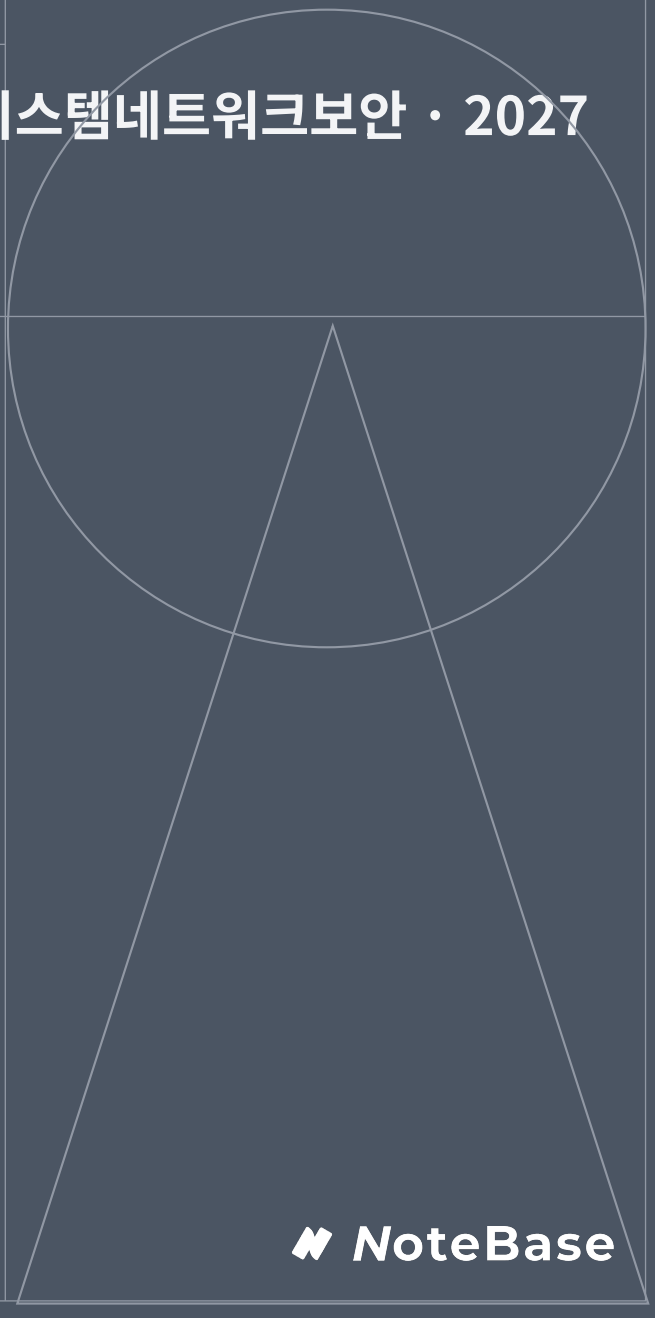
기출OX 해설집

공무원/경찰/공채(경위) · 시스템네트워크보안 · 2027



NoteBase

기출OX 해설집

공무원/경찰/공채(경위) · 시스템네트워크보안 · 2027

공무원/경찰/공채(경위) · 시스템네트워크보안 · 2027

문 1. 공격자가 일반 사용자 계정을 탈취하여 리눅스 시스템으로 로그인 한 후, 사용자의 이전 실행 명령어 내역을 보려고 한다. 해당 명령어로 가장 적절한 것은?

① touch

② dmesg

③ history

④ messages

정답 3번

간단한 해설

사용자의 이전 셸 명령어 내역은 history로 확인한다. **셸 히스토리와 커널·시스템 로그**는 서로 다른 기록이다.

핵심 풀이

사용자 명령 이력의 확인: Bash와 같은 셸에서 history는 **셸의 명령어 히스토리 목록을 표시**하는 기능을 제공한다. 셸에 저장되거나 현재 세션에 기록된 이전 명령어를 확인하려는 목적에 맞으므로 ③이 해당한다. 모든 사용자와 모든 프로그램의 실행을 빠짐없이 기록하는 시스템 감사 로그라는 뜻은 아니다.

다른 항목과의 구별: touch는 파일의 접근·수정 시각을 갱신하며, 기본 동작에서는 없는 파일을 생성할 수 있다. dmesg는 **커널 링 버퍼의 메시지**를 확인하는 명령어로 부팅·장치·드라이버 관련 메시지와 관계가 있다. 두 명령어 모두 사용자의 이전 셸 명령 목록을 조회하는 기능과는 다르다.

messages의 의미: messages는 일부 리눅스 시스템에서 일반 시스템 로그 파일의 이름으로 사용된다. 사용자 명령어 이력을 보여 주는 표준 셸 명령어의 이름은 아니다. 커널 메시지, 일반 시스템 로그, 셸 히스토리를 서로 구분해야 한다.

확인 가능한 범위: 표시되는 이력은 사용하는 셸과 저장 설정, 파일에 실제 남아 있는 기록에 따라 달라진다. 따라서 history가 해당 목적에 가장 적절하다는 것과, 과거의 모든 실행 내역을 반드시 복원할 수 있다는 것은 별개이다.

NoteBase

문 2. FIN Scan에 대한 설명으로 가장 적절한 것은?

① UDP에서 사용한다.

② 닫힌 포트는 RST를 반환한다.

③ 열린 포트는 reachable을 반환한다.

④ Nmap에서는 옵션 -sX를 사용한다.

정답 2번

간단한 해설

FIN Scan은 TCP의 FIN 플래그를 이용한다. 표준 동작을 따르는 **닫힌 포트는 RST로 응답**하고, 열린 포트는 응답하지 않는다.

선지별 해설

① X: FIN Scan은 TCP 헤더의 FIN 플래그만 설정하여 응답을 비교하는 방식이다. UDP에는 TCP의 FIN 플래그가 없으므로 **UDP가 아니라 TCP**에서 사용한다.

② O: 표준 동작을 따르는 TCP 구현에서 닫힌 포트로 FIN 탐색 패킷이 도착하면 RST로 응답한다. FIN Scan은 **RST 응답을 닫힌 포트의 근거**로 사용한다.

③ X: 표준 동작을 따르는 열린 TCP 포트는 이 탐색 패킷을 버려 응답하지 않는다. reachable이라는 응답을 반환하는 방식이 아니다. 다만 필터가 패킷을 버려도 무응답일 수 있으므로 Nmap은 무응답을 open|filtered로 표시한다.

④ X: Nmap의 FIN Scan 옵션은 **-sF**이다. -sX는 FIN·PSH·URG 플래그를 함께 설정하는 Xmas Scan 옵션이다.

따라서 닫힌 포트의 RST 응답을 설명한 ②가 가장 적절하다. 일부 TCP 구현은 열린 포트에도 RST를 반환하므로 이 방법의 판별 가능성은 구현에 따라 달라진다.

NoteBase

문 3. 윈도우 레지스트리에 대한 설명으로 가장 적절하지 않은 것은?

① HKEY_USERS는 컴퓨터에 있는 사용자들의 프로파일 정보가 있다.

② HKEY_LOCAL_MACHINE은 하드웨어 구동 시 필요한 드라이버 정보가 있다.

③ HKEY_CURRENT_CONFIG는 시스템 시작 시 사용하는 하드웨어 프로파일 정보가 있다.

④ HKEY_CLASS_ROOT는 애플리케이션 관리자의 권한 정보가 있다.

정답 4번

간단한 해설

HKEY_CLASSES_ROOT는 **파일 형식 연결과 COM 클래스 등록** 등에 쓰인다. 관리자 권한 정보를 저장하는 키로 설명한 부분이 부적절하다.

선지별 해설

① O: HKEY_USERS에는 로드된 사용자 프로파일의 사용자별 설정이 저장된다. 화면·응용프로그램 환경 등 사용자별 구성을 다루며, 현재 사용자의 설정을 가리키는 HKEY_CURRENT_USER와 연결된다. 모든 사용자의 프로파일이 항상 동시에 로드되어 있다는 의미는 아니다.

② O: HKEY_LOCAL_MACHINE은 특정 사용자에 한정되지 않는 컴퓨터 전체의 하드웨어·소프트웨어 구성을 저장한다. 시스템 하위 영역에는 장치와 드라이버, 서비스의 시작 및 구성 정보가 포함된다.

③ O: HKEY_CURRENT_CONFIG는 현재 사용 중인 하드웨어 프로파일의 구성 정보를 나타낸다. 시스템 시작 시 선택되어 적용된 하드웨어 프로파일을 확인하는 항목이며, 컴퓨터의 모든 과거 하드웨어 구성을 별도로 저장하는 영역은 아니다.

④ X: 표준 키 이름은 HKEY_CLASSES_ROOT이며, **파일 형식 연결과 COM 클래스 등록 정보** 등을 다룬다. 애플리케이션 관리자의 권한 정보를 위한 루트 키가 아니다. 원문의 HKEY_CLASS_ROOT라는 표기는 표준 이름과도 다르다.

따라서 클래스·파일 형식 정보를 관리자 권한 정보로 설명한 ④가 가장 적절하지 않다.

NoteBase

문 4. 공격당한 서버의 메모리 덤프 이미지로부터 은닉된 악성코드를 분석하기 위한 도구로 가장 적절한 것은?

① tcpdump

② Burp suite

③ tripwire

④ Volatility

정답 4번

간단한 해설

메모리 덤프에서 프로세스와 코드 영역의 흔적을 분석하려면 Volatility가 적합하다. **메모리 포렌식**을 패킷 분석·웹 진단·파일 무결성 점검과 구분한다.

핵심 풀이

분석 대상에 따른 선택: 문제는 네트워크 패킷이나 디스크 파일이 아니라 **서버의 메모리 덤프 이미지**를 제시한다. Volatility는 메모리 포렌식 프레임워크로, 메모리에 남아 있는 프로세스·모듈·코드 영역 등의 정보를 분석하여 은닉된 프로세스나 악성코드의 흔적을 조사하는 데 적합하다. 따라서 ④가 해당한다.

네트워크·웹 도구와의 차이: tcpdump는 네트워크 패킷의 캡처와 내용 확인에 사용하는 도구이다. Burp Suite는 웹 애플리케이션의 요청·응답과 취약점을 점검하는 도구 모음이다. 둘 다 보안 분석에 활용되지만, **메모리 덤프 내부 구조를 해석하는 목적**과는 주된 분석 대상이 다르다.

파일 무결성 도구와의 차이: Tripwire는 파일 등의 상태를 기준값과 비교하여 변경을 탐지하는 무결성 점검에 활용된다. 파일 변조 흔적을 찾는 일과 메모리에만 남아 있는 실행 상태를 복원·분석하는 일은 구분해야 한다.

판단의 한계: 메모리 분석 결과는 덤프의 완전성, 운영체제와 도구의 지원 범위 등에 영향을 받는다. Volatility가 적절한 도구라는 것은 모든 은닉 악성코드를 자동으로 확정하거나 제거한다는 뜻이 아니라, 관련 메모리 증거를 조사할 수 있다는 뜻이다.

NoteBase

문 5. DNSSEC이 제공하는 기능으로 가장 적절하지 않은 것은?

① DNS 정보 무결성

② DNS 응답 정보 인증

③ DNS 응답 정보 기밀성

④ DNS 정보 위변조 방지

정답 3번

간단한 해설

DNSSEC은 전자서명으로 DNS 데이터의 출처와 무결성을 검증한다. 내용을 암호화해 숨기는 **기밀성은 제공하지 않는다**.

선지별 해설

① O: DNSSEC은 DNS 레코드 집합에 대한 전자서명을 검증하여 내용이 변조되지 않았는지 확인하는 데이터 무결성 기능을 제공한다. 단순히 응답이 도착했다는 사실이 아니라 서명된 데이터가 보존되었는지를 검증한다.

② O: 신뢰 기준점에서 이어지는 서명 검증을 통해 응답 데이터가 정당한 출처의 것인지 확인하는 **데이터 출처 인증**을 제공한다. 응답을 전달한 모든 중간 서버나 전송 연결 자체를 인증하는 의미와는 구별된다.

③ X: DNSSEC은 전자서명으로 진위와 무결성을 검증할 뿐 **질의·응답 내용을 암호화하여 숨기지 않는다**. 따라서 DNS 응답 정보의 기밀성을 제공하는 기능으로 볼 수 없다.

④ O: 위조되거나 변조된 DNS 데이터는 올바른 서명 검증을 통과하지 못하도록 하여 신뢰하지 않는 데 활용된다. 이 의미에서 DNS 정보의 위·변조 공격을 방어한다. 공격자가 패킷을 변경하는 물리적 행위 자체를 불가능하게 만든다는 뜻은 아니다.

따라서 DNSSEC의 제공 기능이 아닌 것은 ③ 기밀성이다.

NoteBase

문 6. WPA3 무선 LAN 보안 표준에서 기존 WPA2의 PSK (Pre-Shared Key) 방식의 무차별 대입 공격 취약점을 방어하기 위해 도입한 인증 방식으로 가장 적절한 것은?

① WEP(Wired Equivalent Privacy)

② TKIP(Temporal Key Integrity Protocol)

③ SAE(Simultaneous Authentication of Equals)

④ AES-CCMP(CounterMode/CBC MAC Protocol)

정답 3번

간단한 해설

WPA3-Personal의 SAE는 비밀번호 기반 인증·키 교환 방식이다. **수집한 통신을 이용한 오프라인 비밀번호 추측**에 대한 저항성을 제공하며, 데이터 암호화 방식인 AES-CCMP와 구별된다.

핵심 풀이

인증 방식의 변화: WPA3-Personal에서는 기존 WPA2-Personal의 PSK 방식 대신 SAE를 사용한다. SAE는 **비밀번호를 바탕으로 상호 인증과 키 교환을 수행**하는 방식이며, 제시된 인증 방식에 해당하므로 ③이 적절하다. WPA3의 모든 운용 모드가 동일한 개인용 인증을 사용한다는 뜻은 아니다.

방어 대상의 구체화: 핵심은 수집한 인증 통신만으로 비밀번호 후보를 반복 검증하는 **오프라인 사전 대입 공격에 대한 저항성**이다. 모든 형태의 추측 시도나 구현상의 취약점까지 절대적으로 불가능하게 만든다는 의미로 확대하면 안 된다. 비밀번호 품질과 안전한 구현·설정도 여전히 중요하다.

WEP와 TKIP: WEP는 초기 무선 LAN 보안 방식이고, TKIP는 기존 WEP의 문제를 보완하기 위해 WPA에서 사용한 과도기적 보안 방식이다. 둘 다 WPA3-Personal에서 새로 도입한 SAE 인증을 가리키지 않는다.

AES-CCMP와의 차이: AES-CCMP는 무선 데이터의 기밀성과 무결성을 보호하는 방식으로 WPA2에서도 사용한다. **데이터를 보호하는 암호 방식과 비밀번호 기반 인증·키 교환 방식**은 역할이 다르므로, PSK 대체 인증을 묻는 이 문제에서는 CCMP가 아니라 SAE를 선택한다.

NoteBase

문 7. 다음의 YARA 룰에 대한 설명으로 가장 적절하지 않은 것은?

```
rule mal_a{
 meta:
  author = “police”
  date = “2026-08-01”
  description = “a 악성코드 탐지”

 strings:
  $mal_str1 = “VirtualAllocEx”
  $mal_str2 = “WriteProcessMemory”
  $mal_str3 = “cmd.exe /c” nocase
  $file_header = {4D 5A}

 condition:
  $file_header at 0 and 2 of ($mal_str*) and ($mal_str3 >
2)
}
```

① 파일의 시작이 MZ이다.

② strings는 텍스트 문자, 16진수, 정규표현식 모두 가능하다.

③ 이 YARA 룰은 파일에 포함된 문자 패턴을 탐지한다.

④ 문자열 중 2번째 문자인 “WriteProcessMemory”와 “cmd.exe /c”가 2번 이상 나타날 때 탐지한다.

정답 4번

간단한 해설

'2 of ($mal_str*)'는 세 문자열 패턴 중 **적어도 두 종류가 존재한다**는 뜻이지, 특정 두 문자열이 각각 두 번 이상 나타난다는 뜻이 아니다. 출현 횟수는 '$' 대신 **'#'으로 표시**하며, 2회 이상을 검사하려면 '>= 2'로 비교한다.

NoteBase

선지별 해설

① O: '$file_header = {4D 5A}'에서 16진수 바이트 4D와 5A는 각각 ASCII 문자 M과 Z에 해당한다. '$file_header at 0'은 이 바이트열이 **파일의 시작 위치인 오프셋 0**에 있는지를 검사하므로, 파일의 시작이 MZ라는 설명에 부합한다.

② O: YARA의 strings 영역에는 **텍스트 문자열, 16진수 바이트 패턴, 정규표현식**을 정의할 수 있다. 제시된 룰에서는 'VirtualAllocEx', 'WriteProcessMemory', 'cmd.exe /c'를 텍스트 문자열로, '{4D 5A}'를 16진수 패턴으로 정의했다. 정규표현식을 사용한다면 패턴을 슬래시로 둘러싸서 정의한다.

③ O: 이 룰은 strings 영역에 정의된 문자열과 바이트 패턴을 파일에서 찾고, condition 영역에서 그 결과를 조합하는 방식이다. 따라서 **파일에 포함된 패턴을 검사**하는 룰이다. 'WriteProcessMemory'라는 문자열이 있는지 확인하는 것과 해당 함수를 실제로 호출하는지는 구별해야 한다.

④ X: '2 of ($mal_str*)'에서 '$mal_str*'는 '$mal_str1', '$mal_str2', '$mal_str3'를 묶어 가리킨다. 이 조건은 세 패턴 중 **적어도 두 종류가 발견**되면 충족하므로, 반드시 'WriteProcessMemory'와 'cmd.exe /c'가 함께 있어야 하는 것도 아니고 각 문자열의 반복 횟수를 검사하는 것도 아니다. 예를 들어 이 부분만 보면 'VirtualAllocEx'와 'WriteProcessMemory'가 한 번씩 있어도 충족한다. 또한 제시문의 마지막 조건은 '$mal_str3 > 2'인데, '$mal_str3'는 문자열의 존재 여부를 나타내며 출현 횟수가 아니다. 횟수를 비교하려면 '#mal_str3'를 사용해야 하므로, 이 표현을 'cmd.exe /c가 2번 이상 나타남'으로 해석할 수 없다. 두 문자열이 각각 2회 이상 나타나는지를 검사하는 표현은 '#mal_str2 >= 2 and #mal_str3 >= 2'이다. 횟수에 '> 2'를 적용하면 2회 이상이 아니라 **3회 이상**이라는 점도 구별해야 한다.

NoteBase

문 8. 2023년 대규모 웹 서비스들을 마비시켰던 고난도 DDoS 공격 기법으로, 단일 TCP 연결 내에서 수많은 스트림을 동시에 열고 복수의 RST_STREAM 프레임을 전송함으로써 서버의 프레임 처리 루프와 자원을 순식간에 고갈시키는 HTTP/2 프로토콜 고유의 취약점 공격으로 가장 적절한 것은?

① Rapid Reset Attack

② Smurfing Attack

③ HTTP GET Flooding Attack

④ CC Attack

정답 1번

간단한 해설

HTTP/2에서 스트림을 열고 **RST_STREAM으로 즉시 취소하는 동작을 반복**하는 것이 Rapid Reset Attack의 특징이다. 요청을 취소해도 서버에는 이미 시작한 요청 처리와 취소 처리의 부담이 남아 자원이 고갈될 수 있다.

공통 해설

HTTP/2는 하나의 TCP 연결 안에서 여러 요청을 스트림으로 나누어 처리한다. RST_STREAM은 특정 스트림을 취소하는 프레임으로, 이를 전송해도 **TCP 연결 자체는 유지**되므로 같은 연결에서 새 스트림을 계속 열 수 있다.

Rapid Reset Attack은 다수의 스트림을 열어 요청을 보낸 뒤, 응답을 기다리지 않고 RST_STREAM으로 취소하는 동작을 빠르게 반복하는 공격이다. 서버는 취소된 요청에 대해서도 스트림 자료구조 할당, 헤더 압축 해제, 요청 분석 등의 처리 비용을 부담한다. 공격자는 응답 완료를 기다리지 않고 계속 새 요청을 보낼 수 있어 서버의 처리 자원을 소모시킨다. 따라서 문항의 핵심은 단순히 요청을 많이 보내는 것이 아니라 **HTTP/2 스트림의 생성과 즉시 취소를 반복**한다는 점이며, 이 동작을 가리키는 명칭이 Rapid Reset Attack이다.

NoteBase

문 9. BGP 라우팅 프로토콜의 취약점을 악용한 IP 하이재킹(BGP Hijacking)을 원천적으로 차단하기 위해, 인터넷 주소 자원(IP 및 AS 번호)의 소유권을 암호학적으로 인증된 자격 증명(ROA) 형태로 발행하고 라우터가 이를 검증하도록 하는 프레임워크로 가장 적절한 것은?

① RPKI(Resource Public Key Infrastructure)

② BGPsec(Border Gateway Protocol Security)

③ IPSec(Internet Protocol Security)

④ DANE(DNS-based Authentication of Named Entities)

정답 1번

간단한 해설

인터넷 주소 자원의 정당한 보유자를 인증하고, 특정 IP 대역을 어느 AS가 광고할 수 있는지를 **ROA로 증명**하는 공개키 기반 체계가 RPKI이다.

공통 해설

RPKI는 IP 주소와 AS 번호의 할당 관계를 자원 인증서로 검증할 수 있게 하는 공개키 기반 구조이다. 이 기반에서 IP 주소 자원 보유자는 특정 AS가 해당 IP 대역의 경로를 광고하도록 허용하는 ROA(Route Origin Authorization)를 전자서명하여 발행한다. 따라서 문제의 '인터넷 주소 자원 인증과 ROA를 이용하는 프레임워크'는 RPKI에 해당한다.

ROA의 핵심은 **IP 대역과 경로 광고를 허용받은 기원 AS의 연결**이다. 검증기가 인증서와 ROA의 서명을 확인하면, 라우터는 검증된 정보를 이용해 수신한 BGP 경로의 기원 AS와 광고 대역이 허용 범위에 맞는지 판단한다. 운영자가 유효하지 않은 경로를 거부하도록 정책을 적용하면, 허가받지 않은 AS의 경로 광고를 차단할 수 있다.

이때 검증하는 것은 경로가 시작되는 AS의 광고 권한이다. **경로에 포함된 모든 AS의 전달 과정을 검증하는 BGPsec**과는 구별된다. RPKI 기반 기원 검증만으로 모든 형태의 BGP 하이재킹을 차단하는 것은 아니며, 실제 차단에는 검증 결과를 반영하는 라우팅 정책도 필요하다.

NoteBase

문 10. 데이터베이스 암호화의 유형 중 컬럼 암호화 방식으로 가장 적절하지 않은 것은?

① API 방식

② Plug-in 방식

③ Hybrid 방식

④ TDE 방식

정답 4번

간단한 해설

컬럼 암호화와 블록 암호화를 나누는 이 문항의 분류에서 TDE는 **블록 암호화**에 해당한다. 다만 Oracle처럼 TDE 컬럼 암호화도 지원하는 구현이 있으므로, TDE가 컬럼 암호화를 전혀 지원하지 않는다고 일반화해서는 안 된다.

선지별 해설

컬럼 암호화는 테이블의 특정 컬럼을 선택하여 그 컬럼의 데이터를 암호화하는 방식이다.

① O: API 방식은 **응용프로그램에서 암·복호화 함수를 호출**하여 선택한 컬럼의 데이터를 암호화하는 방식이다. 응용프로그램 소스에 함수 호출을 반영해야 하며, 암·복호화 처리를 응용프로그램 쪽에서 수행한다.

② O: Plug-in 방식은 **DBMS에 설치한 암·복호화 모듈**을 이용하여 선택한 컬럼의 데이터를 암호화하는 방식이다. 암·복호화 처리가 DB 서버에서 이루어지므로 DB 서버의 자원을 사용한다.

③ O: Hybrid 방식은 **API 방식과 Plug-in 방식을 함께 사용**하는 컬럼 암호화 방식이다. 응용프로그램과 DB 서버에 암·복호화 처리를 나누어 적용하여 응용프로그램 수정 범위와 DB 서버의 부하를 조정한다.

④ X: 이 문항의 분류에서 TDE는 컬럼 암호화가 아니라 **DBMS 커널에서 수행하는 블록 암호화**로 구분한다. 여기서의 TDE는 데이터 파일에 저장할 때 암호화하고 읽을 때 DBMS가 자동으로 복호화하여, 응용프로그램이 별도의 암·복호화 함수를 호출하지 않아도 저장 데이터를 보호하는 방식이다. 다만 실제 Oracle TDE는 개별 컬럼 암호화와 테이블스페이스 암호화를 모두 지원한다. 따라서 이 문항의 분류와 달리, 모든 TDE 제품에서 컬럼 암호화가 불가능한 것은 아니다.

NoteBase

문 11. TCP SYN Flooding 공격 방어기법인 SYN Cookie에 대한 설명으로 가장 적절한 것은?

① 서버가 SYN 수신 시 즉시 메모리를 할당한다.

② 클라이언트의 ACK가 도착한 후 연결 정보를 메모리에 할당한다.

③ TCP Window Size를 증가시켜 공격을 완화한다.

④ TCP Sequence Number를 생성하지 않는다.

정답 2번

간단한 해설

SYN Cookie는 미완료 연결 정보를 시퀀스 번호 등에 담아 보내고, **유효한 ACK가 돌아온 뒤 연결 상태를 복원·할당**한다. SYN만 보내는 요청마다 상태 메모리를 유지하지 않는 것이 핵심이다.

선지별 해설

① X: SYN Cookie의 핵심은 SYN을 받았다는 이유만으로 **반쯤 열린 연결의 상태를 따로 유지하지 않는 것**이다. 필요한 정보를 SYN-ACK의 시퀀스 번호 등에 담아 돌려주고, 연결 완료가 확인될 때까지 해당 연결 상태의 메모리 할당을 미룬다. 패킷을 처리하는 데 어떤 메모리도 전혀 쓰지 않는다는 의미는 아니다.

② O: 클라이언트의 ACK가 돌아오면 그 확인 번호 등에 포함된 정보를 바탕으로 쿠키의 유효성을 검사한다. **유효한 ACK로 연결 완료를 확인한 뒤** 필요한 연결 정보를 복원하고 상태를 할당한다. 아무 ACK나 도착하면 무조건 연결을 수락한다는 뜻은 아니다.

③ X: TCP 윈도 크기는 수신 측이 받아들일 수 있다고 알리는 데이터량과 관련된 흐름 제어 요소이다. SYN Cookie는 윈도를 키우는 방식이 아니라, 미완료 연결의 상태 저장을 피하여 **SYN 대기로 인한 상태 메모리 소모를 줄이는 방식**이다.

④ X: SYN Cookie는 시퀀스 번호를 없애지 않는다. 오히려 **검증에 필요한 정보를 인코딩한 시퀀스 번호**를 생성하여 SYN-ACK에 넣고, 이후 ACK를 이용해 검증한다. TCP의 번호 체계를 활용하는 방어기법이다.

따라서 유효한 ACK가 돌아온 뒤 연결 상태를 할당한다는 ②가 적절하다. 이 기법은 미완료 연결 상태 고갈을 완화하지만 대역폭이나 처리 자원을 노리는 모든 서비스 거부 공격을 없애는 것은 아니다.

NoteBase

문 12. TLS 1.3 프로토콜이 TLS 1.2에 비해 보안성과 성능을 크게 향상시킨 특징으로 가장 적절하지 않은 것은?

① 서버 인증서를 포함한 모든 메시지를 SHA-224 이상으로 난독화했다.

② Static RSA와 Static DH 키 교환 방식을 퇴출하고 Forward Secrecy를 강제화했다.

③ 차단 및 취약성이 발견된 CBC 모드 암호화 및 SHA-1 해시 함수 사용을 금지했다.

④ 1-RTT 핸드셰이크를 기본으로 하며, 0-RTT 모드를 지원하여 성능을 높였다.

정답 1번

간단한 해설

SHA-224는 **해시 함수**이므로 메시지를 암호화하는 수단이 아니다. TLS 1.3은 서버 인증서를 포함한 **ServerHello 이후의 핸드셰이크 메시지**를 AEAD 방식으로 암호화한다.

선지별 해설

① X: SHA-224는 입력의 해시값을 만드는 함수이며, **메시지를 암호화하는 알고리즘이 아니다**. TLS 1.3은 AES-GCM 등 인증된 암호화 방식인 AEAD로 ServerHello 이후의 핸드셰이크 메시지를 보호한다. 서버 인증서는 암호화되지만, ClientHello와 ServerHello는 평문이므로 '모든 메시지'라는 설명도 잘못되었다.

② O: **공개키 기반 키 교환의 변경**에 관한 설명이다. Static RSA와 Static DH를 제거하고 임시 키를 사용하는 (EC)DHE로 순방향 비밀성을 제공한다. 이는 장기 비밀키가 나중에 유출되어도 과거 통신의 기밀성을 지키는 성질이다. 다만 '강제화'를 모든 통신에 적용하면 부정확하다. PSK 단독 방식은 순방향 비밀성을 보장하지 않으며, 0-RTT 조기 데이터에도 이 보장이 없다.

③ O: 레코드 보호에는 **AEAD만 허용**하므로 CBC 모드를 사용하지 않는다. RFC 8446의 TLS 1.3 암호군은 SHA-256 또는 SHA-384를 사용하며, 핸드셰이크의 CertificateVerify 서명에도 SHA-1을 금지한다. 다만 구형 인증서의 SHA-1 서명까지 예외 없이 금지한다는 뜻은 아니다.

④ O: RTT는 메시지를 보내고 응답을 받는 왕복 시간이다. 기본 핸드셰이크에서는 한 번의 왕복 후 클라이언트가 애플리케이션 데이터를 보낼 수 있다. 0-RTT는 PSK와 서버의 허용 조건이 갖춰졌을 때 **ClientHello와 함께 조기 데이터를 전송**하여 대기 시간을 줄인다. 다만 조기 데이터에는 재전송 공격 위험이 있다.

NoteBase

문 13. 웹 서버를 대상으로 하는 DoS 공격 중, HTTP 요청 헤더의 끝을 알리는 개행 문자(\r\n\r\n)를 완전하게 전송하지 않고 \r\n만 보낸 뒤, 지속적으로 가짜 헤더 필드를 천천히 전송하여 서버의 연결 자원을 고갈시키는 공격으로 가장 적절한 것은?

① SYN Flooding Attack

② Slowloris Attack

③ Hulk DoS Attack

④ Hash DoS Attack

정답 2번

간단한 해설

HTTP 요청 헤더를 **완성하지 않은 채 천천히 추가 전송**하면 서버가 헤더 수신을 기다리며 연결을 유지하게 된다. 이처럼 연결을 오래 점유해 자원을 고갈시키는 방식이 Slowloris 공격이다.

공통 해설

HTTP/1.1에서 각 헤더 줄은 \r\n으로 끝나지만, 헤더 영역 전체가 끝났음을 나타내려면 빈 줄을 포함한 \r\n\r\n이 필요하다. Slowloris Attack은 **헤더 종료를 알리는 빈 줄을 보내지 않고** 가짜 헤더 필드를 조금씩 추가하여 요청을 미완성 상태로 유지하는 DoS 공격이다. 취약한 웹 서버는 남은 헤더를 기다리는 동안 해당 연결에 자원을 할당한 채 유지한다. 이런 연결이 다수 쌓이면 정상 사용자의 요청을 처리할 연결 자원이 부족해진다. 문항의 핵심 조건인 불완전한 헤더, 느린 추가 전송, 연결 자원 고갈은 이 공격의 작동 방식에 해당한다.

NoteBase

문 14. 중요 서버에 대한 인바운드 공격을 방어할 때, 악성 트래픽이 방화벽을 통과하여 내부망 깊숙이 들어오기 전에 ISP 또는 최상위 네트워크 경계에서 공격 트래픽을 폐기(drop)하도록 유도하는 라우팅 기반 방어 기법으로 가장 적절한 것은?

① BGP Flowspec

② RTBH Filtering

③ VPC Peering

④ Split Tunneling

정답 2번

간단한 해설

RTBH는 라우팅 정보를 이용하여 공격 트래픽을 **상위 경계의 폐기 경로**로 보낸다. 공식 채점 답은 RTBH이며, 상위망 폐기 자체는 BGP FlowSpec으로도 구현할 수 있다.

NoteBase

개념 구분

RTBH Filtering은 원격에서 라우팅 정보를 배포하여 특정 트래픽이 상위 경계의 폐기 경로로 향하도록 만드는 블랙홀 필터링 기법이다. 대상 주소에 대한 경로의 다음 홉을 폐기 인터페이스로 해석하도록 구성하는 등의 방법으로, 공격 트래픽이 내부 장비나 회선을 더 소모하기 전에 버리도록 한다.

문항은 중요한 서버로 들어오는 공격을 내부 방화벽이 아니라 ISP나 최상위 경계에서 폐기하도록 유도한다고 한다. RTBH는 이 목적을 **상위 라우터의 경로 선택을 이용해 블랙홀로 보내는 방식**으로 달성한다. 목적지 기반 RTBH는 해당 서버로 향하는 정상 트래픽도 함께 버릴 수 있으므로 정밀한 공격 패킷만을 자동 구분하는 기술로 이해하면 안 된다.

BGP FlowSpec은 주소·프로토콜·포트 등 흐름 조건과 필터링 동작을 배포하여 더 세밀한 통제를 수행한다. 이 기법도 상위망의 인입 라우터에서 조건에 맞는 트래픽을 폐기할 수 있다. RTBH가 폐기 경로를 설정하는 데 비해, FlowSpec은 **흐름 조건에 따른 필터링 규칙과 동작**을 배포한다는 차이가 있다.

VPC Peering은 가상 사설망 사이의 연결이고,

Split Tunneling은 VPN을 통과할 트래픽과 직접 나갈 트래픽을 구분하는 방식이므로 상위 경계에 폐기 경로를 유도하는 기법과 다르다.

공식 채점 답은 ② RTBH Filtering이다. 다만 원문은 블랙홀 경로나 폐기 인터페이스의 설정을 명시하지 않았으므로, 상위 경계에서 라우팅 정보를 이용해 트래픽을 폐기한다는 조건만으로는 ① BGP FlowSpec을 배제하고 ②만을 유일하게 구별하기 어렵다.

NoteBase

문 15. Zero Trust 아키텍처의 논리적 구성 요소 중, 주체와 리소스 간의 연결을 궁극적으로 정의하고 엔터프라이즈 리소스에 대한 접근 권한을 부여하거나 취소하는 핵심 '두뇌' 역할을 수행하는 구성 요소로 가장 적절한 것은?

① 정책 결정 지점(PDP)

② 정책 집행 지점(PEP)

③ 신뢰 엔진(TE)

④ 지속적 진단 및 완화 시스템(CDM)

정답 1번

간단한 해설

정책 결정 지점(PDP)은 접근 요청과 보안 정책·상태 정보를 바탕으로 리소스 접근의 허용·거부·취소를 결정한다. 이처럼 **접근 여부를 최종 판단**하므로 주체와 리소스의 연결을 통제하는 '두뇌'에 해당한다.

공통 해설

정책 결정 지점(PDP)은 주체가 특정 리소스에 접근해도 되는지를 보안 정책과 사용자·기기 상태 등의 정보에 따라 판단하는 논리적 구성 요소이다. 핵심은 단순히 정보를 수집하거나 연결을 차단하는 것이 아니라, **접근의 허용·거부·취소를 결정**한다는 데 있다. PDP 내부에서는 정책 엔진(PE)이 접근 여부를 판단하고, 정책 관리자(PA)가 그 판단에 따라 연결을 설정하거나 종료하도록 정책 집행 지점에 명령한다. 따라서 문항에서 주체와 리소스 간 연결을 궁극적으로 정의하는 '두뇌'는 정책 결정 지점에 대응한다.

이 기준에서 정책 집행 지점(PEP)은 **결정된 정책을 실제 연결에 적용**하는 역할이다. 신뢰 엔진(TE)의 신뢰도 계산과 지속적 진단 및 완화 시스템(CDM)이 제공하는 기기 상태 정보는 접근 판단의 근거가 된다. 이들은 각각 판단의 실행이나 지원을 담당하므로, 접근 여부를 최종 결정하는 PDP와 구별된다.

NoteBase

문 16. SSLv3 프로토콜은 단일 계층이기보다는 2개의 계층으로 된 프로토콜이다. 상위 계층과 하위 계층으로 나눌 때 하위 계층에 해당하는 프로토콜로 가장 적절한 것은?

① Alert 프로토콜

② Record 프로토콜

③ Handshake 프로토콜

④ Change Cipher Spec 프로토콜

정답 2번

간단한 해설

SSLv3의 **Record 프로토콜이 하위 계층**을 이루며, 상위의 핸드셰이크·경고 등 메시지를 레코드로 처리하여 전달한다.

개념 구분

SSLv3의 하위 계층인 Record 프로토콜은 상위 프로토콜의 메시지와 응용 데이터를 레코드 단위로 처리한다. 데이터를 분할하고 필요한 보호 처리를 적용한 뒤 헤더를 붙여 TCP를 통해 전달하며, 수신 측에서는 이를 역으로 처리하여 상위 계층에 제공한다.

따라서 Alert·Handshake·Change Cipher Spec은 **Record 계층 위에서 전달되는 제어 프로토콜**이다. Handshake는 인증 및 보안 매개변수·키 설정을 협의하고, Alert는 오류나 연결 종료를 알리며, Change Cipher Spec은 협의한 암호 상태의 적용 전환을 알린다.

이 구분은 문항이 지정한 SSLv3의 구조를 설명한 것이다. 상위 제어 메시지의 의미와 이를 실제 전송·보호하는 하위 Record의 역할을 구별하면 된다.

따라서 하위 계층에 해당하는 것은 ② Record 프로토콜이다.

NoteBase

문 17. IPSec의 보안 연관(Security Association)의 식별 정보로 사용되는 매개변수로 가장 적절하지 않은 것은?

① 보안 매개변수 인덱스

② IP 목적지 주소

③ IPSec 프로토콜 모드

④ 보안 프로토콜 식별자(AH 또는 ESP)

정답 3번

간단한 해설

보안 연관의 전통적인 식별 조합은 SPI, IP 목적지 주소, AH·ESP 프로토콜 식별자이다. **전송·터널 모드는 식별자가 아니라 동작 속성**이므로 구분해야 한다.

선지별 해설

이 문항은 보안 연관을 구별하는 전통적인 세 요소, 즉 SPI·IP 목적지 주소·보안 프로토콜 식별자의 조합을 묻는다. 보안 연관에 저장하는 모든 설정값이 그 식별 요소인 것은 아니다.

① O: 보안 매개변수 인덱스인 SPI는 수신 측이 해당 패킷을 어떤 보안 연관으로 처리할지 선택하는 식별값이다. **보안 연관을 찾아가는 인덱스**이므로 식별 정보에 해당한다.

② O: 전통적인 세 요소 식별 방식에서 IP 목적지 주소는 해당 보안 연관의 수신 대상을 구별한다. SPI 및 보안 프로토콜 식별자와 함께 사용되는 정보이다. 터널 모드의 수신 패킷을 조회할 때의 목적지 주소와, 복호화 후 내부 패킷의 트래픽 선택 조건을 혼동하지 않는다.

③ X: 전송 모드와 터널 모드 같은 IPsec 프로토콜 모드는 **선택된 보안 연관이 패킷을 보호하는 동작 방식**이다. 보안 연관의 속성으로 관리되지만, 이 문항의 세 요소 식별 조합에는 포함되지 않는다. 식별에 쓰는 값과 식별 후 적용하는 처리 설정을 구별해야 한다.

④ O: 보안 프로토콜 식별자는 AH용 보안 연관인지 ESP용 보안 연관인지를 구별한다. **AH·ESP라는 프로토콜 종류**는 전송·터널이라는 동작 모드와 서로 다른 구분이다.

따라서 식별 요소로 가장 적절하지 않은 것은 ③이다. 참고로 이후 IPsec 규격에서는 유니캐스트 보안 연관을 SPI만으로 구별하고 프로토콜·목적지 주소를 선택적으로 함께 사용하는 방식도 허용하므로, 모든 구현이 반드시 세 항목을 모두 조회한다는 뜻으로 일반화하지 않는다.

NoteBase

문 18. ICMP 오류 보고 메시지 중 Type 3인 Destination unreachable 메시지의 유형을 생성하는 경우로 가장 적절하지 않은 것은?

① 라우터에서 목적지로 패킷을 전송할 경로를 찾지 못한 경우

② 목적지 호스트의 특정 프로토콜(protocol)을 사용할 수 없는 경우

③ IP 패킷의 단편화가 필요하지만 단편화할 수 없는 경우

④ 특정 호스트를 목적지로 하는 경로를 재지정해야 하는 경우

정답 4번

간단한 해설

목적지 도달 불가는 ICMP Type 3이다. 더 적절한 경로를 사용하도록 알리는 경로 재지정은 **Type 5인 Redirect**에 해당한다.

선지별 해설

① O: 라우터가 목적지 네트워크나 호스트에 도달할 경로를 찾을 수 없으면 목적지 도달 불가를 알릴 수 있다. IPv4 ICMP Type 3에서 Code 0은 네트워크 도달 불가, Code 1은 호스트 도달 불가를 나타낸다.

② O: 목적지 호스트가 IP 헤더에 지정된 상위 프로토콜을 지원하지 못하면 **Type 3, Code 2인 Protocol Unreachable**로 보고할 수 있다. 특정 응용 포트에 도달할 수 없음을 나타내는 Code 3과 구별된다.

③ O: 다음 링크로 전달하려면 단편화가 필요하지만 DF 비트 때문에 단편화가 금지되어 있으면 **Type 3, Code 4**를 사용한다. 이는 단편화가 필요하지만 DF가 설정되어 있음을 알리는 경우이다.

④ X: 더 적절한 다음 홉이나 경로를 사용하도록 송신 호스트에 알리는 메시지는 **Type 5인 Redirect**이다. 목적지에 도달할 수 없다는 Type 3과 목적지로 가는 경로를 재지정하라는 Type 5는 목적이 다르다.

따라서 Destination Unreachable의 생성 상황이 아닌 것은 ④이다.

NoteBase

문 19. TCP 연결 생성, TCP 연결 중단 또는 TCP 연결 종료 과정에서 사용되는 TCP 제어 플래그로 가장 적절하지 않은 것은?

① ACK

② FIN

③ RST

④ URG

정답 4번

간단한 해설

ACK는 수신 확인, FIN은 송신 종료, RST는 연결 리셋에 쓰인다. URG는 **긴급 포인터의 유효성**을 나타내므로 연결 생성·중단·종료 목적과 구별된다.

선지별 해설

① O: ACK는 확인 번호 필드가 유효함을 나타낸다. 연결 생성에서는 SYN에 대한 수신 확인에, 정상 종료에서는 FIN에 대한 수신 확인에 사용되므로 **연결 설정과 종료 절차에 참여**한다. 데이터 전달 중에도 쓰이므로 연결 제어에만 한정된 플래그는 아니다.

② O: FIN은 송신 측이 더 이상 보낼 데이터가 없음을 알린다. TCP는 양방향 통신이므로 **한쪽의 FIN은 그쪽 송신 방향의 종료**를 뜻하며, 반대 방향도 별도로 종료될 수 있다. 정상적인 연결 종료에서 사용하는 플래그이다.

③ O: RST는 연결 리셋을 나타내며, 비정상적인 연결 중단이나 연결 요청의 거부 등에 사용된다. 수신 측이 유효한 리셋으로 받아들이면 정상적인 FIN 교환 절차와 달리 연결을 중단한다. 수신한 모든 RST를 검증 없이 무조건 수락한다는 뜻은 아니다.

④ X: URG는 **긴급 포인터 필드가 유효함**을 표시한다. 긴급 데이터 처리와 관련된 제어 정보이며, 연결을 생성하거나 리셋하거나 정상 종료하도록 지시하는 기능은 아니다. URG 자체가 TCP 플래그가 아니라는 것이 아니라, 문제에서 묻는 연결 생명주기 제어 목적에 맞지 않는 것이다.

따라서 연결 생성·중단·종료를 위한 플래그로 가장 적절하지 않은 것은 ④ URG이다.

NoteBase

문 20. SSH 프로토콜에 대한 설명으로 가장 적절하지 않은 것은?

① SSH 사용자 인증 프로토콜은 SSH 전송 계층 프로토콜의 하위 계층에서 동작한다.

② SSH 전송 계층 프로토콜은 데이터 기밀성을 제공한다.

③ SSH 연결 프로토콜은 하나의 채널 상에서 여러 개의 논리적 통신 채널을 다중화한다.

④ SSH가 지원하는 원격 포워딩에서는 사용자의 SSH 클라이언트가 서버의 역할을 할 수 있다.

정답 1번

간단한 해설

SSH 사용자 인증은 **전송 계층 위에서** 이루어진다. 전송 계층이 제공하는 암호화된 통신 기반을 이용하여 사용자를 인증한다.

선지별 해설

① X: SSH 사용자 인증 프로토콜은 **SSH 전송 계층 프로토콜 위에서** 동작한다. 먼저 전송 계층이 암호화와 서버 인증 등 안전한 통신 기반을 마련하고, 사용자 인증 계층이 그 위에서 클라이언트 사용자를 서버에 인증한다.

② O: SSH 전송 계층은 협의한 암호 알고리즘으로 전달 데이터를 보호하여 기밀성을 제공한다. 서버 인증과 데이터 무결성 보호도 담당하며, 이는 사용자 인증 및 연결 프로토콜이 이용하는 보안 기반이다.

③ O: SSH 연결 프로토콜은 하나의 암호화된 연결을 여러 논리적 채널로 나누어 사용한다. 원격 셸이나 명령 실행, 포워딩 등이 각각의 채널을 통해 전달될 수 있으므로 물리적 연결을 용도마다 새로 만들 필요가 없다.

④ O: 원격 포워딩에서는 SSH 서버 쪽에서 수신한 연결을 SSH 터널을 통해 클라이언트 쪽으로 전달하고, 클라이언트 쪽에서 지정한 서비스에 연결할 수 있다. 따라서 클라이언트 측 호스트가 서비스 제공 서버 역할을 할 수 있다. 이는 **터널 끝의 서비스 역할**을 뜻하며, SSH 연결 자체의 클라이언트·서버 역할이 뒤바뀐다는 뜻은 아니다.

따라서 사용자 인증 계층의 위치를 거꾸로 설명한 ①이 가장 적절하지 않다.

NoteBase

문 21. 리눅스 라우팅 테이블 설정 정보를 출력했을 때 항목 정보에 대한 설명으로 가장 적절한 것은?

① Destination에는 네트워크 IP 주소는 들어갈 수 없고 목적지 호스트 주소만 들어간다.

② Genmask가 255.255.255.255인 경우에는 목적지 호스트 식별 마스크이다.

③ Gateway는 목적지로 전송하기 위한 네트워크 인터페이스 이름이다.

④ UG 플래그는 해당 경로가 활성화되어 있지 않으며 게이트웨이를 사용한다는 의미이다.

정답 2번

간단한 해설

255.255.255.255 마스크는 32비트가 모두 일치하는 특정 호스트를 식별한다.
Destination·Gateway·Iface의 역할과 U·G 플래그를 각각 구분해야 한다.

선지별 해설

① X: Destination은 목적지 호스트뿐 아니라 **목적지 네트워크**도 나타낸다. 특정 호스트 하나로 향하는 경로와 어떤 네트워크 대역으로 향하는 경로를 모두 라우팅 테이블에 둘 수 있으므로, 호스트 주소만 가능하다는 제한이 잘못이다.

② O: Genmask가 255.255.255.255이면 IPv4 주소의 32비트 전체를 비교하는 마스크이다. 따라서 **특정 호스트 주소 하나에 대응하는 /32 경로**를 나타낸다. 여기서 마스크의 값과 패킷의 목적지 주소를 혼동하여 브로드캐스트라고 판단하지 않는다.

③ X: Gateway는 다음으로 패킷을 전달할 게이트웨이 주소를 나타내며, 인터페이스 이름이 아니다. **출력 인터페이스는 Iface 항목**으로 구별한다. 직접 연결된 경로처럼 게이트웨이가 별도로 없는 경우도 있다.

④ X: U는 해당 경로가 활성화되어 있다는 뜻이고, G는 게이트웨이를 사용한다는 뜻이다. 따라서 UG는 **활성화된 게이트웨이 경로**이지 비활성 경로가 아니다. 경로가 활성이라는 표시만으로 목적지 서비스가 실제로 항상 응답함을 보장하는 것은 아니다.

따라서 Genmask가 255.255.255.255이면 호스트 경로라는 ②가 적절하다.

NoteBase

문 22. NAT 기술에 대한 설명으로 가장 적절하지 않은 것은?

① NAT 라우터의 내부 네트워크에 있는 호스트들은 사설 IP 주소들이 할당되어 있다.

② Static NAT 기술은 하나의 사설 IP 주소와 하나의 공인 IP 주소를 1:1 고정적으로 변환하는 방식이다.

③ Dynamic NAT 기술은 다수의 사설 IP 주소와 소수의 공인 IP 주소를 m:n 동적으로 변환하는 방식이다.

④ NAPT 기술은 포트 번호를 이용하여 하나의 사설 IP 주소를 다수의 공인 IP 주소로 변환하는 방식이다.

정답 4번

간단한 해설

NAPT는 포트 번호까지 변환하여 **여러 내부 호스트가 공인 주소를 공유**하게 한다. 하나의 사설 주소를 여러 공인 주소로 늘리는 방식이 아니다.

선지별 해설

① O: 사설망을 공인망에 연결하는 일반적인 NAT 구성에서는 내부 호스트에 사설 IP 주소를 부여하고, 외부 통신 시 NAT 장비가 공인 IP 주소로 변환한다. 내부 주소를 외부에서 직접 라우팅할 수 없다는 문제를 주소 변환으로 해결한다.

② O: Static NAT는 특정 내부 주소와 특정 외부 주소의 **고정된 1:1 대응**을 설정한다. 해당 매핑은 통신할 때마다 공인 주소 풀에서 새로 선택하는 방식이 아니다.

③ O: Dynamic NAT는 내부 주소 집합에 대해 공인 주소 풀에서 사용 가능한 주소를 필요할 때 동적으로 할당한다. 전체 주소 집합의 관계를 m:n이라고 표현하지만, 기본 동적 NAT에서 **각 활성 매핑은 1:1**이며 주소 풀이 소진되면 추가 매핑이 제한된다. 포트 번호로 여러 내부 호스트를 같은 공인 주소에 동시에 구별하는 NAPT와 다르다.

④ X: NAPT는 IP 주소뿐 아니라 전송 계층 포트 번호도 변환하여 **다수의 내부 사설 주소가 하나 또는 소수의 공인 주소를 공유**하도록 한다. 하나의 사설 주소를 다수의 공인 주소로 늘린다는 설명은 변환 방향과 핵심 효과를 거꾸로 서술한 것이다.

따라서 NAPT의 다대일 주소·포트 변환을 반대로 설명한 ④가 가장 적절하지 않다.

NoteBase

문 23. netstat 명령어에 대한 설명으로 가장 적절한 것은?

① -l 옵션은 프로토콜별로 모든 네트워크 통계 정보를 출력한다.

② -n 옵션은 네트워크 인터페이스별 통계 정보를 출력한다.

③ -s 옵션은 연결 중이거나 연결 대기 중인 모든 소켓 정보를 출력한다.

④ -r 옵션은 라우팅 테이블 정보를 출력한다.

정답 4번

간단한 해설

netstat의 -r은 라우팅 테이블을 출력한다. **-l은 리스닝, -n은 숫자 표시, -s는 프로토콜별 통계**이므로 제시된 다른 설명과 구분된다.

선지별 해설

① X: 리눅스 net-tools의 netstat에서 -l은 **리스닝 소켓만 표시**하는 옵션이다. 프로토콜별 통계 정보를 보여주는 것은 -s이므로 서로의 기능을 바꾸어 설명했다.

② X: -n은 호스트명이나 서비스명으로 변환하지 않고 **주소와 포트 등을 숫자로 표시**하는 옵션이다. 네트워크 인터페이스별 정보를 출력하는 옵션은 -i이다. 출력 대상을 바꾸는 옵션과 표시 형식을 바꾸는 옵션을 구별해야 한다.

③ X: -s는 **프로토콜별 통계**를 표시한다. 리스닝 소켓과 비리스닝 소켓을 함께 표시하는 것은 -a이다. 통계 카운터를 보는 것과 개별 소켓 목록을 보는 것은 다른 기능이다.

④ O: -r은 **커널 라우팅 테이블**을 출력한다. 목적지와 게이트웨이, 인터페이스 등 패킷 전달 경로에 관한 정보를 확인하므로 제시된 설명이 맞다.

따라서 올바른 설명은 ④이다. 이 문항의 소문자 -l을 숫자 1이나 대문자 I로 혼동하지 않고, -l·-n·-s·-r의 역할을 각각 리스닝·숫자 표시·통계·경로로 연결한다.

NoteBase

문 24. nmap 명령어의 옵션에 대한 설명으로 가장 적절하지 않은 것은?

① -sS 옵션은 TCP SYN 스캔을 위한 옵션이다.

② -sU 옵션은 TCP NULL 스캔을 위한 옵션이다.

③ -p 옵션은 스캔할 포트 번호를 지정하는 옵션이다.

④ -O 옵션은 대상 호스트의 운영체제 정보를 스캔하기 위한 옵션이다.

정답 2번

간단한 해설

Nmap의 **-sU는 UDP Scan**을 지정한다. TCP NULL Scan의 옵션은 -sN이다.

선지별 해설

① O: -sS는 TCP SYN Scan을 지정한다. SYN을 보내 응답을 관찰하며, 일반적인 연결 수립 과정을 끝까지 완료하지 않고 포트 상태를 판별하는 방식이다.

② X: **-sU는 UDP Scan** 옵션이다. TCP NULL Scan은 -sN이며 TCP 플래그를 모두 0으로 둔 패킷을 사용한다. UDP를 뜻하는 U와 NULL을 뜻하는 N을 구별해야 한다.

③ O: -p는 조사할 포트 번호나 범위를 지정한다. 스캔 기법을 바꾸는 -s 계열 옵션과 달리 어떤 포트들을 대상으로 삼을지 정하는 옵션이다.

④ O: **-O는 운영체제 탐지**를 활성화한다. 대상의 네트워크 응답 특성을 운영체제 지문과 비교하여 OS를 추정하며, 항상 정확한 제품·버전을 확정한다는 의미는 아니다.

따라서 UDP Scan 옵션을 TCP NULL Scan으로 설명한 ②가 가장 적절하지 않다.

NoteBase

문 25. 방화벽과 패킷 필터링에 대한 설명으로 가장 적절하지 않은 것은?

① 방화벽은 패킷의 출발지 또는 목적지의 IP 주소나 포트 번호를 보고 패킷을 필터링한다.

② 방화벽에는 일반적으로 페일 세이프(fail safe) 원칙이 적용된다.

③ Egress 필터링은 라우터 내부로 유입되는 패킷을 필터링한다.

④ Blackhole 필터링은 패킷을 Null 인터페이스로 보내 통신이 되지 않도록 필터링한다.

정답 3번

간단한 해설

필터링 방향을 구별하면 된다. **Ingress는 유입, Egress는 유출** 트래픽을 대상으로 하므로 Egress를 내부로 들어오는 패킷의 필터링이라고 한 설명은 잘못되었다.

선지별 해설

① O: 패킷 필터링은 출발지·목적지 IP 주소와 TCP·UDP 포트 번호 등 헤더 정보를 정책과 비교하여 허용 여부를 결정한다. 이는 방화벽의 기본 기능이며, 모든 방화벽이 헤더만 검사한다는 뜻은 아니다. 상태 추적이나 응용 계층 검사도 함께 수행할 수 있다.

② O: 보안 장치에는 장애가 나더라도 무방비로 통신을 허용하기보다 안전한 상태를 유지하는 **페일 세이프 원칙**이 일반적으로 적용된다. 방화벽에서는 통제를 정상적으로 수행할 수 없을 때 차단하는 fail-closed 방식과 연결된다. 다만 가용성 요구에 따른 별도 구성이 있으므로 모든 제품이 항상 같은 장애 동작을 한다는 뜻은 아니다.

③ X: **Egress는 기준 네트워크에서 외부로 나가는 방향**의 트래픽을 뜻한다. 내부로 들어오는 트래픽을 검사하는 것은 Ingress 필터링이다. 따라서 선지는 유출과 유입 방향을 바꾸어 설명했다.

④ O: 블랙홀 필터링은 차단 대상 트래픽을 Null 인터페이스 등 폐기 경로로 보내 버리는 방법이다. 해당 패킷은 목적지까지 전달되지 않으므로 공격 트래픽을 차단할 수 있다. 차단 범위에 포함된 정상 트래픽도 폐기될 수 있다는 점에 유의한다.

따라서 Egress를 내부로 유입되는 방향으로 설명한 ③이 가장 적절하지 않다.

NoteBase

문 26. inetd 데몬이 사용하는 inetd.conf 파일의 구조 항목으로 가장 적절하지 않은 것은?

① 서비스명

② 소켓타입

③ 호스트명

④ 실행 경로명

정답 3번

간단한 해설

inetd.conf는 서비스명·소켓타입·프로토콜·실행 사용자·서버 경로 등으로 서비스를 정의한다. **호스트명은 기본 독립 항목이 아니다**.

선지별 해설

① O: 서비스명은 어떤 네트워크 서비스를 제공할지 지정하는 기본 항목이다. 일반적으로 서비스 이름을 서비스 데이터베이스의 포트·프로토콜 정보와 연결하여 사용한다.

② O: 소켓타입은 서비스가 사용하는 소켓의 형태를 지정한다. 예를 들어 stream은 스트림 방식, dgram은 데이터그램 방식의 소켓을 가리킨다.

③ X: 호스트명은 inetd.conf의 **기본 서비스 정의에서 독립된 필수 열**이 아니다. 전형적인 항목은 서비스명, 소켓타입, 프로토콜, 대기 여부, 실행 사용자, 서버 프로그램 경로, 인수이다. 구현별로 수신 주소를 지정하는 확장 문법이 있을 수 있다는 점과 기본 열 구성을 구별한다.

④ O: 실행 경로명은 요청이 도착했을 때 inetd가 실행할 서버 프로그램의 위치를 지정한다. 내부 제공 서비스를 사용하는 경우에는 해당 구현이 정한 별도 표기를 사용할 수 있다.

따라서 전형적인 inetd.conf의 기본 구조 항목으로 적절하지 않은 것은 ③ 호스트명이다.

NoteBase

문 27. i-node에 있는 파일의 속성 정보로 가장 적절하지 않은 것은?

① 파일의 심볼릭 링크 개수

② 파일의 소유자

③ 파일에 대한 접근 권한

④ 파일의 속성을 마지막으로 변경한 시간

정답 1번

간단한 해설

i-node에 저장되는 링크 수는 **하드 링크 수**이다. 심볼릭 링크는 별도 파일이므로 대상 i-node의 링크 수에 포함되지 않는다.

선지별 해설

① X: i-node의 링크 수는 해당 i-node를 참조하는 **하드 링크의 수**이다. 심볼릭 링크는 별도 i-node를 가지고 대상 경로를 저장하므로, 심볼릭 링크를 만든다고 대상 파일의 하드 링크 수가 증가하지 않는다.

② O: i-node에는 소유자 사용자 ID와 소유 그룹 ID가 저장된다. 사용자 이름 자체가 아니라 식별자를 기준으로 소유 관계와 권한 판단에 사용한다.

③ O: 파일 유형과 함께 소유자·그룹·기타 사용자에 대한 읽기·쓰기·실행 권한 등의 모드 정보가 저장된다. 운영체제는 이를 접근 통제에 활용한다.

④ O: i-node 상태가 마지막으로 변경된 시간인 **ctime**이 저장된다. 소유자나 권한 변경 등이 여기에 반영되며, 파일 내용의 마지막 수정 시간인 mtime 또는 파일 생성 시간과 동일한 개념이 아니다.

따라서 하드 링크 수를 심볼릭 링크 수로 바꾸어 설명한 ①이 가장 적절하지 않다.

NoteBase

문 28. 리눅스 시스템에서 프로세스가 시그널(signal)을 받으면 디폴트 동작으로 프로세스가 종료하는 시그널 이름으로 가장 적절하지 않은 것은?

① SIGINT

② SIGSTOP

③ SIGTERM

④ SIGQUIT

정답 2번

간단한 해설

SIGINT·SIGTERM은 기본적으로 종료하고, SIGQUIT은 종료와 코어 덤프 동작을 한다. **SIGSTOP은 실행만 정지**시키므로 종료와 구별해야 한다.

선지별 해설

① O: SIGINT는 인터럽트 시그널이며 **기본 동작은 종료**이다. 보통 터미널에서 Ctrl+C를 누를 때 전경 프로세스 그룹에 전달된다. 처리기를 설치하거나 무시하도록 설정한 경우에는 실제 동작이 달라질 수 있지만, 문제는 기본 동작을 묻는다.

② X: SIGSTOP은 프로세스를 **종료하지 않고 실행을 정지**시킨다. 정지 상태의 프로세스는 SIGCONT로 실행을 재개할 수 있다. SIGSTOP은 프로세스가 처리기를 등록하여 가로채거나 무시하거나 차단할 수 없다는 점도 SIGINT·SIGTERM과 다르다.

③ O: SIGTERM은 종료 요청 시그널로 기본 동작은 프로세스 종료이다. 처리기를 등록하면 자원 정리 등의 처리를 구현할 수 있다. 이는 기본 종료 동작 자체가 자동으로 모든 응용 프로그램의 정리 작업을 수행한다는 뜻은 아니다.

④ O: SIGQUIT의 기본 동작은 **종료와 코어 덤프**이다. 따라서 단순한 실행 정지가 아니라 종료 시그널에 해당한다. 실제 코어 파일 생성 여부는 자원 제한과 시스템 설정 등의 영향을 받는다.

따라서 기본 동작이 종료가 아닌 실행 정지인 ② SIGSTOP이 정답이다.

NoteBase

문 29. 다음에서 설명하는 FTP 공격으로 가장 적절한 것은?

◯ 제어 채널과 데이터 채널을 다르게 사용하고 데이터 채널을 생성할 때 목적지를 확인하지 않는 FTP 설계의 구조적 취약점을 이용하는 공격이다.
◯ FTP 서버의 전송 목적지 주소를 임의로 지정하여 FTP 서버를 경유해 임의의 목적지로 메시지나 자료를 전송하도록 할 수 있다.

① Anonymous FTP Attack
② Brute Force Attack
③ FTP Bounce Attack
④ TFTP Attack

정답 3번

간단한 해설

FTP Bounce Attack은 FTP 데이터 연결의 목적지 지정 기능을 악용하여 **서버를 제3의 목적지에 대한 중계자**로 사용한다.

대응 관계 해설

첫째 단서: FTP는 명령을 전달하는 제어 연결과 파일 등을 전달하는 데이터 연결을 구분한다. 능동 모드에서 클라이언트가 데이터 연결의 목적지 주소·포트를 지정하는 기능이 부적절하게 허용되면, 서버가 제어 연결 상대방이 아닌 제3의 대상에 연결하도록 유도할 수 있다.

둘째 단서: 공격자는 FTP 서버가 제3의 주소로 데이터 연결을 만들게 하여 서버를 중계자로 악용한다. 이처럼 **FTP 서버를 경유해 다른 목적지에 접근**하게 하는 것이 FTP Bounce Attack의 특징이다. 모든 FTP 서버가 무조건 취약한 것은 아니며 목적지 제한 등 구현·설정에 따라 방어된다.

Anonymous FTP Attack은 익명 접속의 권한·파일 공개 설정 등을 악용하는 경우, Brute Force Attack은 자격 증명을 반복 추측하는 경우와 관련된다. TFTP는 FTP와 다른 단순 파일 전송 프로토콜이며, 제어·데이터 연결을 분리하는 FTP의 목적지 지정 악용이라는 단서와 맞지 않는다.

따라서 두 단서가 공통으로 가리키는 것은 ③ FTP Bounce Attack이다.

NoteBase

문 30. 리눅스 FTP 서버(vsftpd) 접근 제어 설정 방법 중 TCP Wrapper를 이용하는 설정 방식의 설명으로 가장 적절하지 않은 것은?

① 파일: /etc/hosts.allow, 규칙: vsftpd:10.20.30. EXCEPT 10.20.30.40, 설명: vsftpd 서비스에 대해서 10.20.30.40 호스트 허용

② 파일: /etc/hosts.allow, 규칙: vsftpd: 10.20.30.11, 설명: vsftpd 서비스에 대해서 10.20.30.11 호스트 허용

③ 파일: /etc/hosts.deny, 규칙: ALL:ALL, 설명: 모든 서비스에 대해서 모든 호스트 거부

④ 파일: /etc/hosts.deny, 규칙: vsftpd:1.2.3.4, 설명: vsftpd 서비스에 대해서 1.2.3.4 호스트 거부

정답 1번

간단한 해설

TCP Wrapper가 적용되는 환경에서 **EXCEPT는 대상 제외**를 뜻한다. 10.20.30. EXCEPT 10.20.30.40은 해당 주소 대역 중 10.20.30.40을 제외하므로, 그 호스트를 이 규칙으로 허용한다는 설명은 틀리다.

선지별 해설

① X: EXCEPT는 앞의 목록에서 뒤의 대상을 제외한다. 따라서 이 규칙은 10.20.30.으로 시작하는 주소 중 **10.20.30.40을 제외한 호스트**를 허용 대상으로 삼는다. 10.20.30.40이 이 규칙으로 허용된다는 설명은 반대이다. 다만 허용 목록에서 제외되었다고 곧바로 최종 거부가 확정되는 것은 아니며, 다른 규칙과 기본 처리까지 확인해야 한다.

② O: hosts.allow에서 vsftpd와 10.20.30.11이 일치하면 해당 호스트의 서비스 접속을 허용한다. **hosts.allow를 먼저 검사**하므로 여기에서 허용되면 hosts.deny의 거부 규칙을 적용하지 않는다.

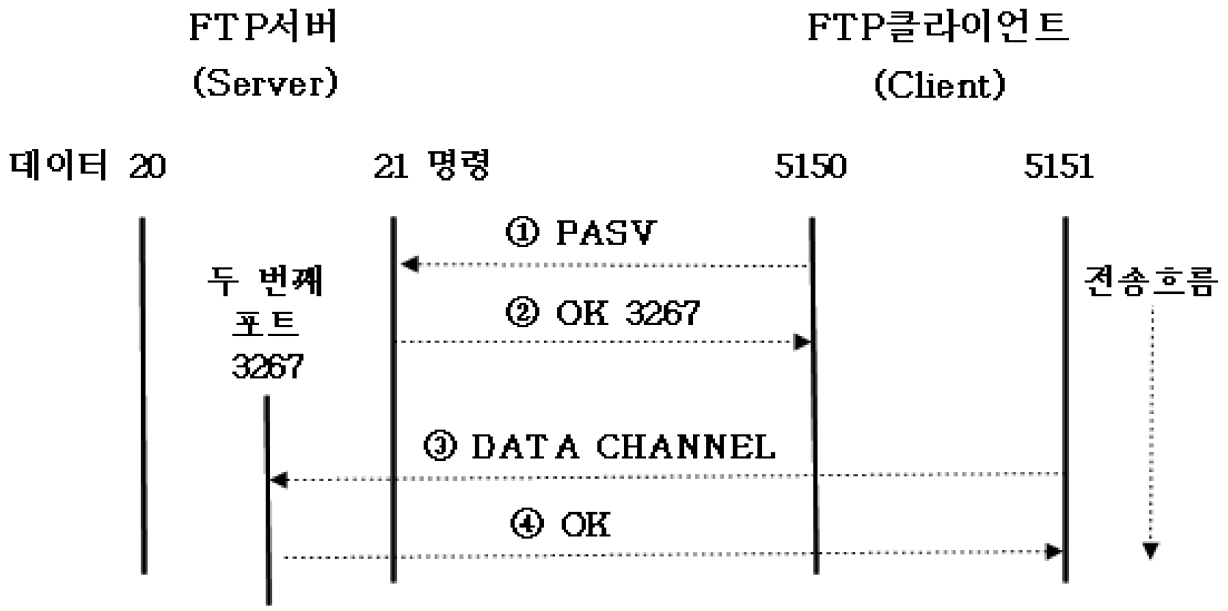
③ O: hosts.deny의 ALL:ALL은 모든 서비스·모든 호스트에 일치하는 거부 규칙이다. 여기서 모든 서비스는 **TCP Wrapper의 통제를 받는 서비스**라는 범위이며, hosts.allow에서 이미 허용한 접속은 예외이다. 운영체제의 모든 통신을 무조건 막는 방화벽 규칙이라는 뜻은 아니다.

④ O: hosts.deny의 vsftpd:1.2.3.4는 해당 주소의 vsftpd 접속을 거부하는 규칙이다. 단, hosts.allow에서 먼저 허용되지 않았을 때 적용된다. 두 파일의 어느 규칙에도 일치하지 않으면 기본적으로 접속을 허용한다.

따라서 EXCEPT로 제외한 호스트를 이 허용 규칙의 대상으로 설명한 ①이 가장 적절하지 않다.

NoteBase

문 31. 다음 FTP 운영 모드에 대한 설명으로 가장 적절하지 않은 것은?



① 클라이언트는 21번 포트로 접속을 시도하며, Passive 모드의 사용을 전달

② 서버는 사용할 포트(3267)를 결정하여 클라이언트에게 전달

③ 서버의 20번 포트는 클라이언트가 알려준 두 번째 포트로 데이터 채널 생성

④ 서버가 ACK로 응답

정답 3번

간단한 해설

FTP 수동 모드에서는 **클라이언트가 서버의 안내받은 데이터 포트로 접속**한다. 그림의 연결도 5151번에서 3267번으로 향하며 서버 20번에서 시작하지 않는다.

NoteBase

선지별 해설

① O: 그림의 클라이언트 5150번 포트와 서버 21번 포트 사이가 제어 연결이다. 클라이언트가 이 연결에서 PASV를 보내 수동 모드를 요청하는 방향이 표시되어 있다.

② O: 수동 모드에서는 서버가 데이터 연결을 기다릴 포트를 정하여 클라이언트에게 알린다. 그림에서는 서버의 두 번째 포트가 **3267번**이며, 제어 연결의 응답에 그 번호가 전달된다.

③ X: 그림의 데이터 연결 시작 화살표는 **클라이언트 5151번에서 서버 3267번으로** 향한다. 서버 20번 포트가 클라이언트의 지정 포트에 접속하는 것은 능동 모드의 전형적인 동작이다. 수동 모드에서 서버 20번 포트를 데이터 연결 시작점으로 설명하면 안 된다.

④ O: 그림의 마지막 OK 화살표는 서버 3267번에서 클라이언트 5151번으로 돌아오는 응답을 나타낸다. 클라이언트가 시작한 데이터 연결에 서버가 응답하는 방향은 수동 모드와 맞는다. 실제 TCP 연결 수립의 최초 응답은 SYN과 ACK가 함께 설정된 세그먼트이므로 그림의 응답 표기는 이를 간략화한 것으로 이해한다.

따라서 수동 모드 그림을 능동 모드의 데이터 연결 방향으로 설명한 ③이 가장 적절하지 않다.

NoteBase

문 32. 다음 ㉠~㉢의 PGP 알고리즘 각각의 기능으로 가장 적절한 것은?

㉠ Radix-64 Conversion
㉡ IDEA, CAST, Triple-DES
㉢ RSA, DSS, RIPEMD-160

① ㉠: 기밀성, ㉡: 인증, ㉢: 호환성
② ㉠: 기밀성, ㉡: 호환성, ㉢: 인증
③ ㉠: 호환성, ㉡: 인증, ㉢: 기밀성
④ ㉠: 호환성, ㉡: 기밀성, ㉢: 인증

정답 4번

간단한 해설

전통적인 PGP 기능 분류에서 Radix-64는 전송 형식의 호환성, 대칭키 암호는 기밀성, 해시와 전자서명은 인증에 대응한다. 이 문제는 **알고리즘 묶음의 역할**을 구분하는 것으로, 나열된 모든 알고리즘을 현재 사용하도록 권장하는 것은 아니다.

대응 관계 해설

㉠ Radix-64 Conversion → 호환성: 이진 데이터를 전송 가능한 ASCII 문자 형태로 바꾸어, 문자 중심의 전자우편 환경에서도 전달할 수 있게 한다. **표현 형식을 바꾸는 인코딩**이며, 그 변환 자체가 비밀키 없이 내용을 읽지 못하게 하는 암호화는 아니다.

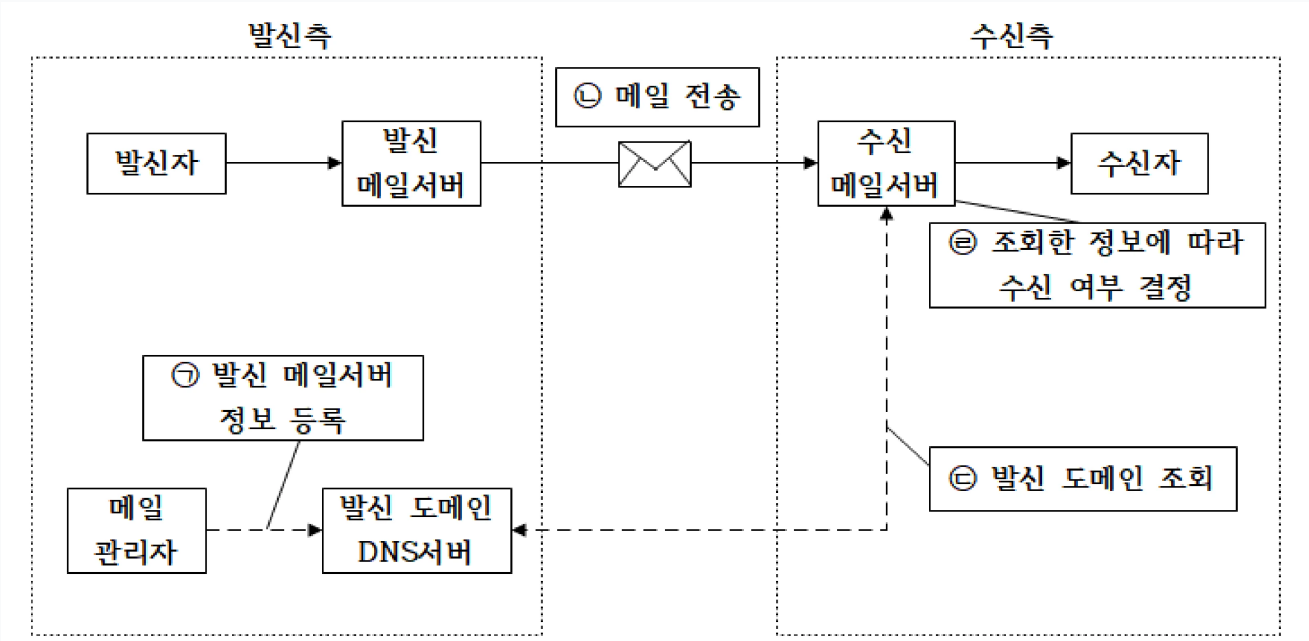
㉡ IDEA · CAST · Triple-DES → 기밀성: 제시된 알고리즘은 대칭키 암호에 해당한다. 메시지 내용을 암호화하여 적절한 키를 가진 수신자가 복호화하도록 하므로, 이 분류에서는 **내용을 숨기는 기밀성**에 대응한다.

㉢ RSA · DSS · RIPEMD-160 → 인증: 이 묶음은 해시와 전자서명을 이용한 인증 기능을 나타낸다. RIPEMD-160은 메시지 요약을 만드는 해시 함수이고, RSA 또는 DSS 계열 서명은 그 요약을 바탕으로 서명을 생성·검증하는 데 쓰인다. **해시 단독이 아니라 서명과의 결합**으로 발신자 인증과 변조 검증을 수행하며, 공개키가 실제 발신자의 것인지에 대한 신뢰도 필요하다. RSA가 다른 맥락에서는 암호화에도 쓰일 수 있다는 사실과 구별한다.

따라서 ㉠ 호환성, ㉡ 기밀성, ㉢ 인증의 조합인 ④가 적절하다.

NoteBase

문 33. 다음의 스팸메일 대응 기술로 가장 적절한 것은?



① Sender Policy Framework

② Domain Keys Identified Mail

③ Spam Assassin

④ Procmail

정답 1번

간단한 해설

그림은 발신 도메인의 DNS에 등록된 정보를 조회하여 **실제 전송 서버가 메일 발송을 허용받았는지** 확인하는 방식이다. 이러한 발송 권한 검증 기술이 **SPF(Sender Policy Framework)**이다.

NoteBase

공통 해설

SPF(Sender Policy Framework)는 도메인 관리자가 DNS에 공개한 발송 허용 정책을 이용해, 메일을 보내 온 서버가 해당 도메인을 사용하여 발송할 권한이 있는지 확인하는 기술이다. 핵심 기준은 **실제 전송 서버의 IP 주소가 도메인의 발송 허용 정책에 부합하는지**이다.

그림의 ㉠에서는 관리자가 발신 도메인의 DNS 서버에 허용된 발신 메일서버 정보를 등록한다. ㉡에서 메일이 전송되면, 수신 메일서버는 ㉢에서 발신 도메인의 SPF 정보를 DNS로 조회하고 실제 전송 서버의 IP 주소와 대조한다. ㉣에서는 그 검증 결과를 수신 정책에 반영한다. 이 등록·조회·대조 관계가 SPF의 동작을 나타낸다. 여기서 확인하는 도메인은 SMTP의 MAIL FROM이나 HELO/EHLO에 사용된 도메인이며, 검증 통과가 메일 내용의 안전성까지 보장하지는 않는다.

DNS를 이용한다는 사실만으로 DKIM과 구별할 수는 없다. SPF가 발송 서버의 권한을 확인하는 데 비해, DKIM은 DNS에서 얻은 공개키로 **메일에 붙은 전자서명을 검증**한다. 그림의 판별 근거는 공개키나 서명이 아니라 발신 메일서버 정보를 등록하고 조회하는 관계이다.

NoteBase

문 34. /etc/mail/access 파일의 설정 제어 옵션에 대한 설명으로 가장 적절하지 않은 것은?

① RELAY - 지정한 대상의 메일 릴레이를 허용한다.

② REJECT - 거부 메시지를 보내서 지정한 대상의 메일을 모두 거부한다.

③ DISCARD - 폐기 메시지를 보내서 지정한 대상의 메일을 모두 폐기한다.

④ OK - 지정한 대상의 메일을 조건 없이 모두 허용한다.

정답 3번

간단한 해설

DISCARD는 **폐기 사실을 통지하지 않고 메일을 폐기**한다. 거부 메시지를 보내는 REJECT와 달리, 발신자에게 '폐기 메시지'를 보내는 방식이 아니다.

선지별 해설

① O: RELAY는 지정한 대상에 대해 **메일 릴레이를 허용**한다. 릴레이란 메일 서버가 메일을 받아 다른 메일 서버로 중계하는 기능이며, RELAY에는 수신 허용의 의미도 포함된다.

② O: REJECT는 지정한 대상의 메일을 **거부 메시지와 함께 거부**한다. 송신 측에 오류 응답으로 거부 사실을 알린다는 점에서, 통지 없이 폐기하는 DISCARD와 구분된다.

③ X: DISCARD는 지정한 대상의 메일을 **폐기 통지 없이 조용히 폐기**한다. 따라서 '폐기 메시지를 보내서'는 '폐기 사실을 알리지 않고'로 고쳐야 한다. 메일을 수신자에게 전달하지 않으면서도 송신 측에는 폐기 사실을 통지하지 않는 방식이다.

④ O: OK는 지정한 대상의 메일을 현재 규칙 집합의 다른 거부 규칙에도 불구하고 수신하도록 허용한다. 선지의 '조건 없이'는 **해당 규칙 집합 안에서의 수신 허용**으로 이해해야 한다. 모든 보안 검사를 해제하거나 외부로의 릴레이까지 허용한다는 뜻은 아니며, 릴레이 허용은 RELAY와 구분한다.

NoteBase

문 35. 다음에서 설명하는 보안 전자 메일 프로토콜로 가장 적절한 것은?

◯ 메일을 안전하게 전송하기 위한 프로토콜
◯ 메시지에 대한 기밀성, 무결성, 사용자 인증, 송신 사실 부인 방지
◯ X.509 인증서를 통해 송신자 인증 보장

① SMTP
② S/MIME
③ IMAP
④ RBL

정답 2번

간단한 해설

S/MIME는 **X.509 인증서 기반의 전자서명**과 메시지 암호화를 지원한다. 전자서명은 무결성·송신자 인증·부인 방지를, 암호화는 기밀성을 제공한다.

공통 해설

S/MIME(Secure/Multipurpose Internet Mail Extensions)는 MIME 형식의 전자 메일 메시지에 암호화와 전자서명을 적용하는 보안 표준이다. 메시지 암호화는 허가받지 않은 사람이 내용을 읽지 못하게 하여 기밀성을 제공한다. 전자서명은 메시지의 변조 여부와 서명자를 확인하게 하여 **무결성·송신자 인증·부인 방지**를 지원한다. 따라서 지문의 보안 기능을 갖춘 전자 메일 프로토콜은 S/MIME에 해당한다.

X.509 인증서는 사용자의 신원과 공개키를 연결한다. 수신자는 **인증서의 유효성을 확인하고 그 공개키로 전자서명을 검증**하여 송신자를 인증한다. 인증서를 첨부하는 것만으로 인증이 끝나는 것은 아니며, 유효한 인증서와 서명 검증이 함께 필요하다.

NoteBase

문 36. 다음의 SQL Injection 공격 방식의 유형으로 가장 적절한 것은?

```
id=1 UNION SELECT 1, username, password FROM users --
```

① Form SQL Injection

② Union SQL Injection

③ Stored Procedure SQL Injection

④ Error-based SQL Injection

정답 2번

간단한 해설

UNION SELECT로 기존 조회 결과에 users 테이블의 username과 password 조회 결과를 결합하려는 공격이다. 다른 테이블의 데이터를 기존 응답에 섞어 가져오는 Union SQL Injection의 특징이 드러난다.

공통 해설

Union SQL Injection은 입력값에 SQL의 UNION 연산자를 삽입하여 **기존 SELECT의 결과와 공격자가 추가한 SELECT의 결과를 결합**하는 방식이다. 제시문에서는 'UNION SELECT 1, username, password FROM users'가 users 테이블에서 사용자명과 비밀번호를 조회해 기존 결과에 합치려는 부분이다. 따라서 공격 유형은 Union SQL Injection에 해당한다.

추가된 SELECT의 첫 번째 열은 상수 1이고, 두 번째와 세 번째 열은 username과 password이다. UNION으로 결합하려면 두 SELECT의 **반환 열 수가 같고 대응하는 열의 자료형이 호환**되어야 하므로, 이 구문이 실행되려면 기존 조회 결과도 세 열이어야 한다. 끝의 '--'는 뒤따르는 SQL 구문을 주석 처리하려는 부분이다. 공격 방식을 가르는 기준은 입력 경로나 오류 발생 여부가 아니라, UNION으로 조회 결과를 결합한다는 점이다.

NoteBase

문 37. 다음 빈칸 (㉠)~(㉢)에 들어갈 용어로 가장 적절한 것은?

○ (㉠) 공격은 가장 일반적인 XSS 공격 유형으로, 단순히 게시판 또는 자료실과 같이 사용자가 글을 저장할 수 있는 부분에 정상적인 평문이 아닌 스크립트 코드를 입력하는 공격 기법이다.
○ (㉡) 공격은 사용자가 악성 스크립트 코드를 포함하고 있는 악성 URL을 클릭하면 서버의 응답으로 악성 스크립트가 실행되어 사용자의 정보를 탈취하거나 리다이렉트하는 공격 기법이다.
○ (㉢) 공격은 보안에 취약한 자바스크립트 코드를 이용해 사용자가 예측하지 못한 방식으로 동작하는 HTML 문서 객체 모델을 제어하는 과정에서 발생하는 공격 기법이다.

① ㉠: Reflected XSS, ㉡: Stored XSS, ㉢: DOM based XSS
② ㉠: Reflected XSS, ㉡: DOM based XSS, ㉢: Stored XSS
③ ㉠: Stored XSS, ㉡: DOM based XSS, ㉢: Reflected XSS
④ ㉠: Stored XSS, ㉡: Reflected XSS, ㉢: DOM based XSS

정답 4번

간단한 해설

저장되는 코드 → Stored, 요청이 응답에 반영 → Reflected, 클라이언트 DOM 처리 → DOM based로 대응시킨다.

NoteBase

빈칸별 해설

㉠ Stored XSS: 게시판 등에 입력한 악성 코드가 저장되어, 이후 다른 사용자가 해당 내용을 조회할 때 전달·실행되는 저장형 공격이다. 구별의 핵심은 발생 빈도에 대한 표현이 아니라 **입력 내용의 저장과 재사용**이다.

㉡ Reflected XSS: 악성 URL 등의 입력이 요청에 실려 서버로 전달되고 응답에 반영되어 실행되는 반사형 공격이다. 제시문은 저장된 게시물의 재조회가 아니라 특정 요청과 그에 대한 서버 응답을 단서로 준다.

㉢ DOM based XSS: 신뢰할 수 없는 데이터를 클라이언트 스크립트가 위험한 방식으로 처리하면서 DOM이나 실행 환경을 바꾸어 생기는 공격이다. 제시문의 취약한 자바스크립트와 문서 객체 모델 조작이 이 유형을 가리킨다.

저장형·반사형은 입력의 저장·전달 양상, DOM 기반은 클라이언트 측 처리 원인을 구분하는 개념이므로 실제로 서로 결합될 수 있다. 다만 이 문항의 각 설명은 저장, 서버 응답 반영, DOM 처리라는 서로 다른 핵심 단서를 제시한다.

따라서 ㉠ Stored XSS, ㉡ Reflected XSS, ㉢ DOM based XSS의 대응이며, 이 배열과 일치하는 선택지는 ④이다.

NoteBase

문 38. 다음에서 설명하는 웹 애플리케이션 공격 기법으로 가장 적절한 것은?

◯ 웹 해킹을 통해 시스템의 권한을 획득한 후 해당 시스템에 텔넷과 같이 직접 명령을 입력하고 확인할 수 있는 쉘을 획득하기 위한 방법
◯ 해커가 자신의 시스템에서 특정 포트를 개방하고, 웹 서버에서 해당 포트로 접속함으로써 해커가 웹 서버에 대한 원격 접속을 얻는 공격 기법

① Cross Site Request Forgery
② HTTP Header Injection
③ Reverse Shell
④ Insecure Direct Object Reference

정답 3번

간단한 해설

웹 서버가 공격자의 시스템으로 접속하여 명령을 입력하고 실행 결과를 확인할 수 있게 하는 방식이 리버스 쉘(Reverse Shell)이다. 연결을 먼저 요청하는 쪽이 공격자가 아니라 공격 대상 서버라는 점이 핵심이다.

공통 해설

리버스 쉘(Reverse Shell)은 공격 대상 시스템이 공격자 측으로 연결을 시작하고, 그 연결을 통해 공격자가 대상 시스템의 쉘을 사용하는 방식이다. 쉘은 운영체제에 명령을 전달하고 실행 결과를 확인하는 인터페이스이다. 여기서 '리버스'는 **공격 대상이 공격자에게 연결을 요청한다**는 연결 방향을 뜻한다.

지문에서는 공격자가 자신의 시스템에서 특정 포트를 열어 연결을 기다리고, 권한을 이미 획득한 웹 서버가 그 포트로 접속한다. 연결이 맺어지면 공격자는 이를 통해 웹 서버에 명령을 전달하고 실행 결과를 받는다. 따라서 두 설명은 모두 **웹 서버에서 공격자 측으로 연결하여 원격 쉘을 제공**하는 리버스 쉘의 작동 방식을 나타낸다.

NoteBase

문 39. 다음의 DNS Spoofing 공격 순서를 올바르게 나열한 것으로 가장 적절한 것은?

㉠ 해커는 사용자의 DNS 질의를 감지하여 가짜 응답을 준비한다.
㉡ 사용자가 특정 도메인에 대한 DNS 질의를 보낸다.
㉢ 해커는 악성 DNS 서버 또는 조작된 IP 주소를 포함한 응답을 사용자에게 전송한다.
㉣ 사용자의 웹 브라우저는 위조된 IP 주소 또는 악성 웹 사이트로 접속된다.

① ㉠→㉡→㉢→㉣
② ㉡→㉠→㉢→㉣
③ ㉠→㉢→㉣→㉡
④ ㉡→㉢→㉠→㉣

정답 2번

간단한 해설

이 시나리오는 **질의가 먼저 발생하고, 그 질의를 감지하여 만든 위조 응답이 수용된 다음 잘못된 주소로 접속**하는 인과관계로 배열한다.

올바른 순서

㉡ → ㉠ → ㉢ → ㉣

㉡ 질의 전송: 사용자가 도메인 이름에 대응하는 주소를 알아내기 위해 DNS 질의를 보내는 것이 이 시나리오의 출발점이다.

㉠ 질의 감지와 준비: 공격자는 이미 전송된 질의를 감지한 뒤 그 질의에 대한 위조 응답을 준비한다. 따라서 질의 발생보다 감지가 앞설 수 없다.

㉢ 위조 응답 전송: 준비한 거짓 주소 정보 등이 포함된 응답을 사용자 측에 전달한다. 정상 응답 대신 받아들여지려면 질의 식별 정보 등이 맞아야 하며, 일반적인 응답 경쟁에서는 정상 응답보다 먼저 도착하는 것도 중요하다.

㉣ 잘못된 주소로 접속: 위조 정보가 DNS 응답으로 수용되고 이름 해석에 사용된 뒤 브라우저가 공격자가 의도한 주소로 향한다. 실제 공격의 성공은 응답 검증과 추가 보안 통제 등에 좌우되므로 위조 응답을 보내기만 하면 항상 성공한다는 뜻은 아니다.

NoteBase

문 40. 다음에서 설명하는 애플리케이션 보안 취약점으로 가장 적절한 것은?

◯ 리눅스 계열 OS에서 주로 사용하는 GNU Bash 환경변수를 통해 코드 인젝션 기법으로 특정 코드를 삽입하여 실행할 수 있는 취약점이다.
◯ 공격자가 환경변수를 조작하여 원격에서 임의의 명령을 실행할 수 있다.

① Heartbleed
② Log4j
③ ShellShock
④ Stuxnet

정답 3번

간단한 해설

GNU Bash의 **환경변수 처리 결함**을 이용해 임의의 명령을 실행하는 취약점은 ShellShock이다.

공통 해설

ShellShock은 GNU Bash가 환경변수에 담긴 함수 정의를 읽어 들일 때 **함수 정의 뒤에 붙은 명령까지 실행**할 수 있는 취약점이다. 환경변수로 전달된 값이 단순한 데이터에 머물지 않고 명령으로 실행된다는 점에서 코드 인젝션에 해당한다. 원격 서비스가 공격자가 조작한 값을 환경변수로 전달하고 취약한 Bash를 실행하는 경우, 공격자는 해당 프로세스의 권한으로 임의의 명령을 실행할 수 있다. 지문의 GNU Bash, 환경변수 조작, 원격 명령 실행이라는 특징은 모두 ShellShock을 가리킨다.

NoteBase