7단원. 네트워크 보안

> 대상: 전공자 · 정보처리기사 취득자 | 목적: 인프라 직무 기술 면접 대비 중요도: ★★★ 최빈출 · ★★ 자주 출제 · ★ 기본 개념

Q1. 대칭키 암호화와 비대칭키 암호화의 차이를 설명하세요. ★★★

답안 **대칭키**는 암호화와 복호화에 같은 키를 씁니다(AES). 빠르고 대용량에 적합하지만, 통신 상대와 **키를 안전하게 공유하는 문제**가 있습니다. **비대칭키(공개키)**는 공개키와 개인키 쌍을 씁니다(RSA, ECC). 공개키로 암호화하면 개인키로만 풀 수 있어 키 배포 문제가 없고 전자서명·인증이 가능하지만 연산이 느립니다. 그래서 실무에서는 **하이브리드** 방식을 씁니다. 비대칭키로 세션 키(대칭키)를 안전하게 교환한 뒤, 실제 데이터는 빠른 대칭키로 암호화합니다. TLS가 정확히 이 구조입니다.

관련 개념 AES/RSA/ECC, 키 배포 문제, 하이브리드 암호화, 전자서명

Q2. TLS 핸드셰이크 과정을 설명하세요. ★★★

답안 TLS는 ① **암호 협상**(클라이언트가 지원 암호 스위트·TLS 버전 제시, 서버가 선택), ② **서버 인증**(서버가 인증서 전송, 클라이언트가 CA 체인으로 검증), ③ **키 교환**(ECDHE로 양쪽이 사전 마스터 시크릿을 안전하게 합의), ④ **세션 키 도출** 후 대칭키 암호화 통신으로 전환합니다. TLS 1.3은 이 과정을 **1 RTT**로 단축하고(1.2는 2 RTT), 재연결 시 0-RTT를 지원하며, 취약한 알고리즘(RSA 키 교환, RC4 등)을 제거했습니다. **ECDHE**를 쓰는 이유는 **전방향 안전성(PFS)** — 서버 개인키가 유출돼도 과거에 녹음된 트래픽을 복호화할 수 없음 — 을 위해서입니다.

관련 개념 암호 스위트, 인증서 체인 검증, ECDHE, PFS, TLS 1.3 개선점

Q3. HTTPS에서 인증서와 CA는 어떤 역할을 하나요? ★★★

답안 인증서는 "이 공개키가 이 도메인의 것이 맞다"를 **CA(인증 기관)**가 보증한 전자 문서입니다. 브라우저는 OS·브라우저에 내장된 신뢰 루트 CA 목록을 기준으로, 서버 인증서 → 중간 CA → 루트 CA로 이어지는 **체인의 서명을 검증**하고, 도메인 일치·유효기간·폐기 여부(CRL/OCSP)를 확인합니다. 하나라도 실패하면 경고를 띄웁니다. 이 구조 덕분에 사용자는 처음 보는 사이트도 신뢰할 수 있습니다(PKI). Let's Encrypt가 무료·자동화(ACME)로 HTTPS를 보편화했습니다. 실무 단골 장애는 인증서 만료, 중간 인증서 누락(브라우저는 되는데 일부 클라이언트에서 실패), 도메인 불일치(SAN 누락)입니다.

관련 개념 PKI, 인증서 체인, OCSP/CRL, SAN, Let's Encrypt/ACME, 인증서 만료 모니터링

Q4. 대칭키 교환에 쓰이는 Diffie-Hellman의 핵심 아이디어는? ★★

답안 DH는 **공개된 채널에서 도청자가 보고 있어도 양쪽만 같은 비밀 값을 만들어내는** 키 교환 방법입니다. 공개 값(g, p)과 각자의 비밀 값(a, b)으로 g^a, g^b를 교환하고, 각자 상대 값에 자기 비밀 지수를 올리면 둘 다 g^(ab)에 도달합니다.

도청자는 g^a, g^b를 봐도 이산 로그 문제 때문에 a, b를 역산할 수 없습니다. 매 연결마다 임시(ephemeral) 키 쌍을 쓰는 **ECDHE**가 전방향 안전성을 제공합니다. "키를 주고받지 않고 어떻게 같은 키를 갖느냐"는 질문의 답이 바로 DH입니다.

관련 개념 이산 로그 문제, ECDHE, 전방향 안전성, 중간자 공격(인증서로 방어)

Q5. 방화벽의 종류(패킷 필터링, 상태 기반, 차세대)를 설명하세요. ★★★

답안 패킷 필터링(stateless)은 각 패킷의 IP·포트·프로토콜을 규칙과 대조해 허용/차단합니다. 빠르지만 연결 문맥을 몰라 정교한 제어가 어렵습니다. **상태 기반(stateful)**은 연결 상태 테이블을 유지해 "내가 보낸 요청의 응답"만 자동 허용하므로, 아웃바운드는 열되 인바운드는 기존 연결의 응답만 통과시키는 정책이 가능합니다(현대 방화벽의 기본). **차세대 방화벽(NGFW)**은 여기에 애플리케이션 식별(포트가 아닌 실제 앱), IPS, TLS 검사, 사용자 인증 연동을 더합니다. 클라우드의 **보안 그룹**은 상태 기반 인스턴스 방화벽, **NACL**은 상태 비저장 서브넷 방화벽으로 대비되며 이 차이가 자주 출제됩니다.

관련 개념 stateful vs stateless, NGFW/IPS, 보안 그룹 vs NACL, 기본 거부(default deny)

Q6. DDoS 공격의 유형과 방어 방법을 설명하세요. ★★★

답안 볼류메트릭(대역폭 고갈: UDP 플러드, DNS·NTP 증폭)은 실제 대역폭을 채워 정상 트래픽을 밀어냅니다. **프로토콜 공격**(SYN 플러드처럼 연결 자원·상태 테이블 고갈). **애플리케이션 계층**(HTTP 플러드처럼 적은 트래픽으로 서버 로직·DB를 소진, 탐지가 어려움). 방어는 계층적입니다. ① 상류의 **스크러빙 센터/CDN**(Cloudflare, AWS Shield)이 대용량 트래픽을 흡수·정화, ② 애니캐스트로 공격을 지리적으로 분산, ③ 레이트 리미팅·캡차·WAF로 애플리케이션 공격 필터링, ④ SYN 쿠키·연결 제한, ⑤ 오토스케일링으로 용량 확보. 핵심은 "혼자 감당하지 말고 상류에서 흡수한다"는 것입니다.

관련 개념 증폭 공격(반사), SYN 플러드, 스크러빙, 애니캐스트 분산, 레이트 리미팅

Q7. 중간자 공격(MITM)은 무엇이고 어떻게 방어하나요? ★★

답안 공격자가 통신 양단 사이에 끼어들어 트래픽을 가로채거나 변조하는 공격입니다. 수법으로 ARP 스푸핑(LAN 내), 악성 Wi-Fi AP, DNS 스푸핑, BGP 하이재킹이 있습니다. 방어의 핵심은 **암호화와 인증**입니다. TLS는 인증서로 상대 신원을 검증해 공격자가 중간에서 위조하지 못하게 합니다(그래서 인증서 경고를 무시하면 안 됨). 추가로 **HSTS**(HTTP를 강제로 HTTPS로), **인증서 피닝**(앱이 특정 인증서만 신뢰), DNSSEC, 신뢰할 수 없는 네트워크에서 VPN 사용이 있습니다. "TLS를 쓰면 왜 MITM이 막히는가"는 인증서 검증 원리로 답하면 됩니다.

관련 개념 ARP/DNS 스푸핑, TLS 인증, HSTS, 인증서 피닝, DNSSEC

Q8. VPN의 동작 원리와 종류를 설명하세요. ★★

답안 VPN은 공용 네트워크 위에 **암호화된 터널**을 만들어 사설망처럼 안전하게 통신하는 기술입니다. **사이트 간 VPN**은 두 네트워크(본사-지사, 온프레미스-클라우드)를 연결하며 주로 **IPsec**을 씁니다(AH/ESP, IKE로 키 협상). **원격 접속 VPN**은 개인 단말을 회사망에 연결하며 SSL/TLS VPN, WireGuard 등을 씁니다. 캡슐화로 원래 패킷을 새 헤더로 감싸 전달하므로 **MTU가 줄어드는** 점(단편화·성능 이슈)을 유의해야 합니다. 최근에는 "네트워크 위치를 신뢰하지 않고 매 접근을 인증"하는 **제로 트러스트**(BeyondCorp 모델)가 전통 VPN을 보완·대체하는 흐름입니다.

관련 개념 IPsec(ESP/IKE), SSL VPN, WireGuard, 터널 MTU, 제로 트러스트

Q9. 인증(Authentication)과 인가(Authorization)의 차이는? ★★

답안 **인증**은 "당신이 누구인지 확인"하는 것(로그인, 신원 증명)이고, **인가**는 "무엇을 할 권한이 있는지 확인"하는 것(권한·접근 제어)입니다. 인증이 먼저, 인가가 나중입니다. HTTP에서 인증 실패는 401, 인가 실패는 403으로 구분됩니다. 다중 요소 인증(MFA)은 지식(비밀번호)·소유(OTP·기기)·생체 중 둘 이상을 결합해 인증을 강화합니다. 인가 모델로 RBAC(역할 기반), ABAC(속성 기반)가 있습니다. OAuth2(위임 인가)와 OpenID Connect(그 위의 인증)의 구분도 함께 알아두면 좋습니다.

관련 개념 401 vs 403, MFA, RBAC/ABAC, OAuth2 vs OIDC

Q10. 포트 스캔과 이를 탐지·차단하는 방법을 설명하세요. ★

답안 포트 스캔은 공격자가 열린 포트와 서비스를 파악하는 정찰 활동입니다(nmap). TCP SYN 스캔(SYN만 보내고 SYN+ACK 오면 열림으로 판단, 연결은 미완성해 로그 회피), 연결 스캔, UDP 스캔 등이 있습니다. 탐지·차단은 ① **IDS/IPS**가 짧은 시간에 다수 포트 접근 패턴을 탐지, ② 방화벽에서 불필요한 포트를 **기본 거부**로 닫고 필요한 것만 열기, ③ fail2ban으로 반복 시도 IP 차단, ④ 공격 표면 최소화(관리 포트는 VPN·배스천 뒤로). "닫힌 포트는 RST, 필터링된 포트는 무응답"이라는 응답 차이로 방화벽 유무까지 추정된다는 점도 이해 포인트입니다.

관련 개념 nmap, SYN 스캔, IDS/IPS, fail2ban, 공격 표면 최소화

Q11. 클라우드 보안 그룹과 NACL의 차이를 설명하세요. (인프라 연계) ★★★

답안 둘 다 AWS의 가상 방화벽이지만 계층과 성격이 다릅니다. **보안 그룹(SG)**은 **인스턴스(ENI) 단위**로 적용되는 **상태 기반** 방화벽으로, 허용 규칙만 있고(명시적 거부 불가), 인바운드를 허용하면 응답 아웃바운드는 자동 허용됩니다. **NACL**은 **서브넷 단위**의 **상태 비저장** 방화벽으로, 허용·거부 규칙을 번호 순서로 평가하며 상태를 기억하지 않아 인바운드와 아웃바운드(임시 포트 범위 포함)를 **양쪽 다 명시**해야 합니다. 일반적으로 세밀한 제어는 SG로 하고, NACL은 서브넷 광역 차단(특정 IP 대역 블랙리스트)에 보조적으로 씁니다. "SG만 열었는데 통신이 안 된다 → NACL의 아웃바운드/임시 포트 확인"이 전형적 트러블슈팅 흐름입니다.

관련 개념 상태 기반 vs 비저장, 인스턴스 vs 서브넷 범위, 임시 포트, 허용 전용 vs 허용/거부

Q12. 시큐어 코딩 관점의 네트워크 취약점(SQL 인젝션, XSS, CSRF)을 간단히 구분하세요. ★

답안 SQL 인젝션은 사용자 입력이 쿼리로 해석되어 DB를 조작당하는 것으로, **파라미터 바인딩(프리페어드 스테이트먼트)**으로 방어합니다. **XSS**는 악성 스크립트가 다른 사용자의 브라우저에서 실행되는 것으로, 출력 이스케이프·CSP·쿠키 HttpOnly로 방어합니다. **CSRF**는 로그인된 사용자의 브라우저가 자동 첨부하는 쿠키를 악용해 의도치 않은 요청을 보내게 하는 것으로, CSRF 토큰·SameSite 쿠키로 방어합니다. 공통 원칙은 "사용자 입력을 신뢰하지 말고, 입력 검증과 출력 인코딩을 분리하라"입니다. 네트워크 직무라도 애플리케이션 보안 기본은 자주 묻습니다.

관련 개념 프리페어드 스테이트먼트, CSP, SameSite, OWASP Top 10
