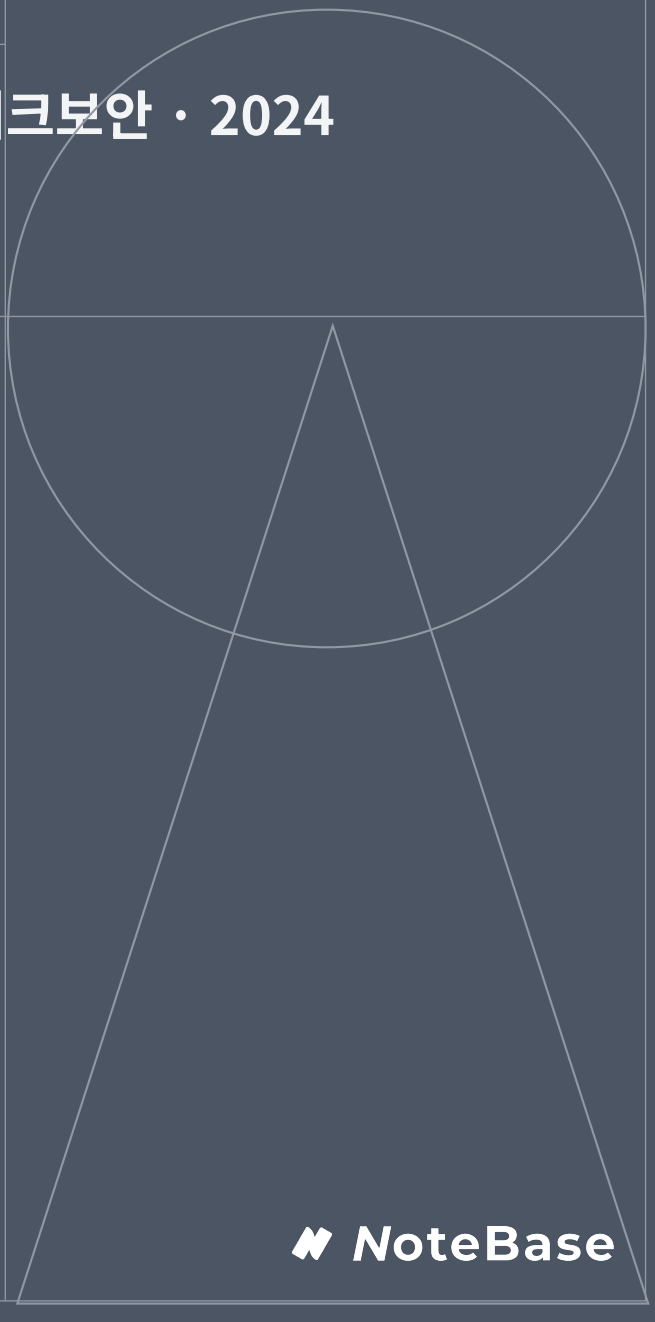
기출OX 해설집

공무원/국가직/9급 · 네트워크보안 · 2024



NoteBase

기출OX 해설집

공무원/국가직/9급 · 네트워크보안 · 2024

공무원/국가직/9급 · 네트워크보안 · 2024

문 1. 공격자가 ARP 스푸핑 공격으로 위조하려는 것은?

① Port 번호

② ICMP 에코 요청 패킷

③ MAC 주소

④ DNS 응답 메시지

정답 3번

간단한 해설

ARP 스푸핑은 다른 장치의 IP 주소에 대응하는 **MAC 주소를 공격자의 MAC 주소로 속이는** 공격이다. 피해 장치가 잘못된 주소 대응을 믿으면 원래 장치로 보낼 프레임이 공격자에게 전달될 수 있다.

핵심 풀이

ARP는 같은 로컬 네트워크에서 IPv4 주소에 대응하는 MAC 주소를 알아내는 프로토콜이다. 장치는 이 대응 관계를 ARP 캐시에 저장하고, 프레임을 보낼 때 목적지 MAC 주소를 정하는 데 사용한다. ARP 스푸핑은 위조한 ARP 메시지로 이 캐시에 **거짓 IP–MAC 대응 관계**가 저장되도록 하는 공격이다.

예를 들어 공격자가 '게이트웨이의 IP 주소에 대응하는 MAC 주소는 공격자의 MAC 주소'라고 속이면, 피해 장치는 게이트웨이로 보낼 프레임을 공격자의 MAC 주소로 전송할 수 있다. 즉, 게이트웨이의 IP 주소 자체를 바꾸는 것이 아니라 **그 IP 주소에 연결된 MAC 주소**를 속인다.

Port 번호는 TCP·UDP에서 통신할 응용 서비스를 구분하는 데 사용하며, ICMP 에코 요청 패킷은 통신 상대의 응답 여부를 확인하는 데 사용한다. DNS 응답 메시지를 위조하여 도메인 이름에 대한 주소 정보를 속이는 것은 DNS 스푸핑이다. 이들은 ARP의 IP–MAC 주소 대응을 속이는 것과 구별된다.

NoteBase

문 2. route 명령으로 얻은 라우팅 테이블 정보와 같은 결과물을 출력할 수 있는 것은?

① traceroute
② tcpdump
③ netstat
④ ping

정답 3번

간단한 해설

netstat에 **-r 옵션**을 사용하면 route 명령으로 조회하는 **라우팅 테이블**을 출력할 수 있다.

핵심 풀이

라우팅 테이블은 목적지 네트워크에 따라 패킷을 어느 게이트웨이나 인터페이스로 보낼지 판단하는 데 쓰는 정보이다. netstat은 연결 상태뿐 아니라 라우팅 정보도 조회하며, **netstat -r**로 라우팅 테이블을 출력한다. 따라서 route 명령으로 확인하는 라우팅 정보를 netstat으로도 확인할 수 있다.

traceroute는 특정 목적지까지 패킷이 거치는 중간 경로를 추적한다. **호스트에 저장된 라우팅 테이블 조회**와 **목적지까지의 경로 추적**은 구분해야 한다.

NoteBase

문 3. 다음에서 설명하는 것은?

◯ 외부 방화벽과 내부 방화벽 사이에 존재한다.
◯ 외부에서 접속할 수 있어야 하며 보호되어야 하는 시스템(예: 회사의 웹사이트, 이메일 서버 등)을 여기에 배치한다.

① NAT(Network Address Translation)
② DMZ(Demilitarized Zone)
③ VPN(Virtual Private Network)
④ PAN(Personal Area Network)

정답 2번

간단한 해설

외부 공개 서버를 **내부망과 분리**하여 배치하면서 외부의 서비스 접속을 허용하는 중간 네트워크 영역이 DMZ이다. 제시된 구성에서는 외부 방화벽과 내부 방화벽 사이에 위치한다.

공통 해설

DMZ(Demilitarized Zone)는 외부에 서비스를 제공하는 서버를 내부망과 분리하여 두는 완충 네트워크 영역이다. 제시된 이중 방화벽 구성은 '외부망 → 외부 방화벽 → DMZ → 내부 방화벽 → 내부망'으로 연결된다. 외부 방화벽은 웹·메일 등 공개 서비스에 필요한 외부 접속을 허용하고, 내부 방화벽은 **DMZ에서 내부망으로의 접근을 제한**한다. 따라서 공개 서버가 공격받더라도 내부망으로 피해가 확산될 위험을 줄일 수 있다. 지문의 웹사이트·이메일 서버처럼 외부에서 접속할 수 있어야 하면서 보호도 필요한 시스템을 배치하는 영역이 DMZ이다.

NoteBase

문 4. 접근 제어 방식에 대한 다음 설명의 (가)와 (나)에 들어갈 내용을 바르게 연결한 것은?

(가)는 사용자 ID가 아닌 시스템에서 정의한 사용자의 역할을 바탕으로 접근 권한을 정의한다. 각 사용자는 (나) 정적 또는 동적으로 할당될 수 있다.

① (가): DAC, (나): 하나의 역할에만
② (가): DAC, (나): 여러 개의 역할에
③ (가): RBAC, (나): 하나의 역할에만
④ (가): RBAC, (나): 여러 개의 역할에

정답 4번

간단한 해설

RBAC는 **역할에 권한을 부여**하고 사용자를 그 역할에 연결한다. 한 사용자는 **여러 개의 역할**을 맡을 수 있으므로 사용자와 역할의 관계가 일대일로 제한되지 않는다.

빈칸별 해설

(가) RBAC: 역할 기반 접근 제어(Role-Based Access Control)는 업무나 책임에 따라 역할을 정의하고, 각 역할에 필요한 접근 권한을 연결하는 방식이다. 사용자는 자신에게 할당된 역할을 통해 권한을 얻으므로, 지문의 **'시스템에서 정의한 사용자의 역할'**이 결정적인 단서이다. DAC는 객체 소유자가 사용자나 그룹에 대한 접근 권한을 재량으로 정하는 방식이므로 이 빈칸에 해당하지 않는다.

(나) 여러 개의 역할에: RBAC에서는 **한 사용자에게 여러 역할을 할당**할 수 있다. 예를 들어 한 사용자가 '일반 직원' 역할과 '프로젝트 관리자' 역할을 함께 부여받을 수 있다. 정책에 따라 역할을 미리 정적으로 할당하거나 상황에 맞게 동적으로 할당할 수 있으며, 사용자마다 하나의 역할만 가져야 하는 것은 아니다.

따라서 (가)는 RBAC, (나)는 '여러 개의 역할에'로 연결되며, ④의 조합과 일치한다.

NoteBase

문 5. HTTP 버전 1.1에서 정의된 메소드의 설명으로 옳지 않은 것은?

① POST는 서버가 클라이언트에게 문서를 전달하는 데 사용한다.

② GET은 클라이언트가 서버에게 문서를 요청할 때 사용한다.

③ HEAD는 클라이언트가 문서 자체가 아니라 문서에 대한 정보를 원할 때 사용한다.

④ TRACE는 클라이언트가 검사 목적으로 서버에게 정보 반환을 요청하는 데 사용한다.

정답 1번

간단한 해설

POST는 **클라이언트가 서버에 데이터를 보내 처리하도록 요청**하는 메소드이다. 서버가 클라이언트에게 문서를 보내는 것은 요청에 대한 응답이며, POST의 용도를 설명한 것이 아니다.

선지별 해설

① X: POST는 **클라이언트가 서버에 데이터를 제출하여 처리를 요청**할 때 사용한다. 예를 들어 입력한 게시글 내용을 서버에 보내 등록을 요청할 수 있다. 서버가 처리 결과로 문서를 돌려줄 수는 있지만, 이를 서버가 클라이언트에게 문서를 전달하는 메소드라고 정의할 수는 없다.

② O: GET은 클라이언트가 요청 URI로 지정한 자원의 내용을 가져오도록 서버에 요청하는 메소드이다. 웹 문서를 요청하면 서버는 해당 문서의 내용을 응답으로 전달하므로, **문서 조회 요청**에 사용한다는 설명이 성립한다.

③ O: HEAD는 GET과 같은 자원을 요청하되, 서버가 **응답 본문을 보내지 않고 헤더 정보만 반환**하는 메소드이다. 문서 자체를 내려받지 않고도 문서의 크기나 최종 수정 시각 같은 정보를 확인할 때 사용한다.

④ O: TRACE는 서버 등 최종 수신자가 **수신한 요청 메시지를 클라이언트에게 되돌려 보내도록**하는 메소드이다. 클라이언트는 자신이 보낸 요청과 반환된 내용을 비교하여 중간 프록시 등을 거치면서 요청이 어떻게 전달되었는지 검사할 수 있다.

NoteBase

문 6. DNS 스푸핑 공격의 순서를 바르게 나열한 것은?

(가) 클라이언트는 공격자가 전송한 DNS Response를 올바른 메시지로 인식한다.
(나) 로컬에 존재하며 DNS 서버보다 클라이언트와 가까이 위치한 공격자는 위조된 DNS Response를 클라이언트에게 전송한다.
(다) 공격자는 스니핑 공격을 통해 클라이언트가 DNS 서버로 DNS Query를 전송하는 것을 확인한다.

① (가) → (나) → (다)
② (나) → (가) → (다)
③ (나) → (다) → (가)
④ (다) → (나) → (가)

정답 4번

간단한 해설

공격자는 **클라이언트의 DNS 질의를 먼저 확인**해야 그 질의에 대응하는 위조 응답을 보낼 수 있다. 클라이언트가 위조 응답을 정상 응답으로 받아들이는 것은 그 응답을 받은 뒤의 일이다.

올바른 순서

(다) → (나) → (가)

(다) 질의 확인: 공격자는 스니핑으로 클라이언트가 DNS 서버에 DNS Query를 보내는 것을 확인한다. 스니핑은 통신 내용을 엿보는 행위이며, 여기서는 **위조 응답을 보낼 대상 질의를 파악**하는 단계이다.

(나) 위조 응답 전송: 로컬에 있으며 DNS 서버보다 클라이언트에 가까이 위치한 공격자가 위조된 DNS Response를 보낸다. 앞서 확인한 질의에 대해 **DNS 서버가 보낸 것처럼 꾸민 응답**을 전달하는 단계이므로 질의 확인 다음에 온다.

(가) 위조 응답 수용: 클라이언트가 공격자의 DNS Response를 올바른 메시지로 인식한다. 응답을 수신해야 그 내용을 받아들일 수 있으므로 위조 응답 전송 뒤에 온다. 이때 클라이언트가 **위조된 정보를 정상적인 DNS 응답으로 신뢰**하게 된다.

NoteBase

문 7. 서버가 클라이언트로부터 받은 비밀키에 대하여 여러 번 해시 함수를 적용하여 얻은 해시 체인으로 일회용 비밀번호(OTP)를 생성하는 것은?

① S/KEY 방식
② 시간 동기화 방식
③ 이벤트 동기화 방식
④ 챌린지-응답 방식

정답 1번

간단한 해설

비밀정보에 해시 함수를 반복 적용하여 만든 **해시 체인**의 값을 일회용 비밀번호로 사용하는 방식은 **S/KEY**이다.

핵심 풀이

S/KEY: 비밀정보를 바탕으로 해시 함수를 반복 적용해 값들을 연결한 해시 체인을 만든다. 이때 앞의 해시값을 다시 해시하면 다음 값이 만들어진다. OTP는 **해시 적용 횟수가 줄어드는 방향**으로 사용한다. 예를 들어 서버가 세 번 해시한 값을 저장했다면, 클라이언트는 두 번 해시한 값을 제출한다. 서버는 제출된 값을 한 번 해시하여 저장값과 비교하고, 일치하면 인증을 허용한 뒤 제출된 값으로 저장값을 갱신한다. 해시값에서 그 이전 입력값을 구하기 어렵다는 성질 때문에, 이미 사용된 OTP를 알아도 다음 OTP를 쉽게 계산할 수 없다. 표준적인 S/KEY에서는 비밀정보로 OTP를 생성하는 주체가 클라이언트이고, 서버는 전달받은 OTP를 검증한다.

시간 동기화 방식과 이벤트 동기화 방식: 시간 동기화 방식은 클라이언트와 서버가 맞춘 **시간값**을, 이벤트 동기화 방식은 OTP 생성 등의 이벤트에 따라 증가하는 **카운터값**을 이용한다. 문항의 반복 해시 체인은 시간값이나 카운터값을 OTP 생성에 사용하는 이 구분과 다르다.

챌린지-응답 방식: 서버가 보낸 **질의값에 대응하는 응답값**을 클라이언트가 비밀정보로 계산하고, 서버가 이를 검증한다. S/KEY에서도 서버가 해시 횟수와 시드 같은 생성 매개변수를 질의로 보낼 수 있으므로, 질의가 있다는 사실만으로 구별하지 않는다. 이 문항을 결정하는 특징은 반복 해시로 만든 체인을 이용한다는 점이다.

NoteBase

문 8. WPA(Wi-Fi Protected Access)에 대한 설명으로 옳지 않은 것은?

① WPA는 WEP의 보안 취약점을 보완하기 위해 제안되었다.

② WPA의 TKIP는 암호 알고리즘으로 AES를 사용한다.

③ WPA-EAP는 인증을 위해서 RADIUS 서버를 이용할 수 있다.

④ WPA-PSK는 사전에 공유된 비밀키를 사용한다.

정답 2번

간단한 해설

WPA의 TKIP는 **RC4**를 암호 알고리즘으로 사용한다. AES를 기반으로 하는 것은 WPA2의 대표적인 암호화 방식인 **CCMP**이다.

선지별 해설

① O: WPA는 WEP의 취약점을 보완하기 위해 도입되었다. TKIP를 통해 패킷별 암호화 키를 생성하고 메시지 무결성 검사를 강화하여 **키 관리와 무결성 보호**를 개선하였다.

② X: TKIP는 **RC4 기반**으로 데이터를 암호화하며, AES를 사용하지 않는다. AES 기반 암호화 방식은 WPA2에서 사용하는 CCMP이다. 따라서 'AES를 사용한다'는 부분은 'RC4를 사용한다'로 고쳐야 한다.

③ O: WPA-EAP는 802.1X/EAP를 이용하여 사용자를 인증하는 방식이다. 무선 단말의 인증 메시지를 액세스 포인트가 **RADIUS 인증 서버로 전달**하고, 서버가 인증을 수행하도록 구성할 수 있다.

④ O: WPA-PSK의 PSK는 사전 공유 키를 뜻한다. 무선 단말과 액세스 포인트가 **미리 공유한 비밀값**을 인증과 통신용 키 생성의 기초로 사용하므로, 별도의 RADIUS 인증 서버 없이 구성할 수 있다.

NoteBase

문 9. DoS 공격 방식에 대한 설명으로 옳지 않은 것은?

① 스머프(Smurf) 공격은 발신지 주소가 거짓으로 조작된 ICMP 패킷을 받은 네트워크의 많은 컴퓨터들이 발신지 주소로 응답을 보내게 하여 트래픽 범람을 일으키는 것이다.

② 티어드롭(Teardrop) 공격은 IP 패킷 헤더 정보를 조작해서 패킷 조각(fragment)이 중첩되도록 보내어 이를 전달받은 시스템의 IP 패킷 재조합 과정에서 오류를 발생시킨다.

③ 슬로로리스(Slowloris) 공격은 HTTP 메시지 헤더의 Content-Length 필드에 임의의 큰 값을 설정하여 웹 서버가 클라이언트에서 해당 크기의 메시지를 전송할 때까지 연결을 유지하게 함으로써 정상적인 사용자의 접속을 받아들일 수 없게 하는 것이다.

④ HTTP GET 플러딩 공격은 정상적인 TCP 세션과 함께 정상적으로 보이는 과도한 HTTP GET을 요청받은 웹 서버에 과부하가 걸리게 한다.

정답 3번

간단한 해설

③의 큰 Content-Length 값을 이용해 본문 수신을 기다리게 하는 방식은 **Slow HTTP POST 공격**에 해당한다. **Slowloris는 HTTP 헤더를 끝까지 보내지 않고** 조금씩 전송하여 연결을 오래 점유한다.

선지별 해설

① O: 스머프 공격은 ICMP Echo Request의 발신지 IP 주소를 **피해자의 IP 주소로 위조**한 뒤 브로드캐스트로 전송한다. 이에 응답하는 여러 컴퓨터의 ICMP Echo Reply가 피해자에게 집중되므로, 응답 트래픽이 증폭되어 피해자의 네트워크 자원을 소모한다.

② O: 티어드롭 공격은 IP 헤더의 조각 오프셋 등 조각 위치 정보를 조작하여 **재조합할 데이터 영역이 서로 겹치도록** 한다. 중첩된 조각을 제대로 처리하지 못하는 취약한 시스템에서는 재조합 오류나 시스템 장애가 발생하여 서비스가 중단될 수 있다.

③ X: Content-Length는 HTTP 메시지 본문의 바이트 수를 나타낸다. 여기에 큰 값을 지정하고 본문을 매우 느리게 전송하여 서버가 나머지 데이터를 기다리게 하는 방식은 **Slow HTTP POST 공격**이다. Slowloris는 이와 달리 **HTTP 헤더를 완성하지 않은 채** 헤더 일부를 조금씩 계속 보내어 연결이 닫히지 않게 한다. 서버의 동시 연결 자원이 이러한 미완성 요청에 점유되면 정상 사용자의 새 접속을 처리하기 어려워진다.

④ O: HTTP GET 플러딩 공격은 TCP 연결을 정상적으로 수립한 뒤, 형식상 정상적인 HTTP GET 요청을 대량으로 보낸다. 서버는 각 요청에 대해 자원 조회와 응답 생성 등을 수행해야 하므로 **과도한 요청 처리량** 때문에 자원이 소모되고 정상 사용자에게 제공할 서비스가 지연되거나 중단된다.

NoteBase

문 10. (가)와 (나)에 들어갈 내용을 바르게 연결한 것은?

○ (가)은/는 네트워크 레벨에서 패킷에 대한 보안을 제공하기 위해 설계된 프로토콜의 모음이다.
○ (가)이/가 지원하는 (나)에서는 새로운 IP 헤더가 추가된다.

① (가): IPSec, (나): 전송모드
② (가): IPSec, (나): 터널모드
③ (가): TLS, (나): 전송모드
④ (가): TLS, (나): 터널모드

정답 2번

간단한 해설

IPSec은 IP 계층에서 패킷을 보호하는 프로토콜 모음이다. 이 중 **터널모드**는 원래 IP 패킷 전체를 감싸고 바깥에 새로운 IP 헤더를 붙인다.

빈칸별 해설

(가) IPSec: 네트워크 계층인 IP 계층에서 패킷에 보안 서비스를 제공하는 프로토콜 모음이다. **IP 패킷 자체를 보호**한다는 설명이 IPSec을 가리킨다. TLS는 일반적으로 TCP 위에서 응용 데이터를 보호하므로, 이 문항의 네트워크 계층 보안 프로토콜 모음에 해당하지 않는다.

(나) 터널모드: 원래의 IP 헤더와 데이터를 포함한 IP 패킷 전체를 하나의 내부 패킷으로 캡슐화하고, 전송에 사용할 **새로운 외부 IP 헤더를 추가**한다. 반면 전송모드는 원래 IP 헤더를 유지하면서 그 뒤의 상위 계층 데이터를 보호하며, 터널모드처럼 별도의 외부 IP 헤더를 덧붙이지 않는다. 따라서 새로운 IP 헤더가 추가된다는 조건은 터널모드를 뜻한다.

(가)는 IPSec, (나)는 터널모드로 연결되며, 이는 ②의 조합과 일치한다.

NoteBase

문 11. 무선 랜의 MAC 프레임에 대한 다음 설명의 (가)와 (나)에 들어갈 내용을 바르게 연결한 것은?

802.11 무선 랜 표준의 매체 접속 프로토콜은 (가)이고, 무선 MAC 계층 프레임 포맷에는 (나)개의 주소 필드가 있다.

① (가): CSMA/CD, (나): 2
② (가): CSMA/CD, (나): 4
③ (가): CSMA/CA, (나): 2
④ (가): CSMA/CA, (나): 4

정답 4번

간단한 해설

802.11 무선 랜은 전송 중 충돌을 검출하기보다 전송 전에 충돌 가능성을 줄이는 **CSMA/CA**를 사용한다. MAC 프레임의 일반 포맷은 중계 구간의 송수신자와 원래 출발지·최종 목적지를 구분하기 위해 **최대 4개의 주소 필드**를 둔다.

빈칸별 해설

(가) CSMA/CA: 무선 랜의 기본 매체 접속 방식은 반송파 감지 다중 접속 및 충돌 회피 방식이다. 먼저 채널의 사용 여부를 확인하고, 정해진 대기 및 임의 백오프 절차를 거쳐 전송하여 여러 단말의 동시 전송 가능성을 줄인다. 무선에서는 송신 중 다른 단말의 신호를 감지하여 충돌을 알아내기 어려우므로 **충돌 검출(CD) 대신 충돌 회피(CA)**를 사용한다. CSMA/CD는 전통적인 공유 매체의 반이중 이더넷에서 사용하는 방식이다.

(나) 4: 802.11 MAC 프레임의 일반 포맷에는 Address 1~Address 4의 네 주소 필드가 있다. 무선 구간에서 직접 프레임을 주고받는 송신자(TA)·수신자(RA)가 데이터의 원래 출발지(SA)·최종 목적지(DA)와 다를 수 있기 때문이다. 예를 들어 AP 사이에서 프레임을 중계하면, 두 AP의 주소 외에 원래 출발지와 최종 목적지 주소도 구분해야 한다. **To DS와 From DS가 모두 1인 데이터 프레임**에서는 Address 1에 RA, Address 2에 TA, Address 3에 DA, Address 4에 SA가 들어간다. 다만 모든 프레임이 네 필드를 모두 사용하는 것은 아니며, 실제 주소 필드 수와 의미는 프레임 종류와 전달 방향에 따라 달라진다.

(가)는 CSMA/CA, (나)는 4이므로 ④의 조합과 일치한다.

NoteBase

문 12. SSL/TLS 핸드셰이크 프로토콜에서 클라이언트가 보내는 client_hello 메시지에 포함되지 않는 것은?

① 난수
② 클라이언트가 지원하는 암호 그룹(Cipher Suite) 목록
③ 클라이언트가 지원하는 압축 방식 목록
④ 인증서

정답 4번

간단한 해설

인증서는 client_hello에 넣지 않고 **별도의 Certificate 메시지**로 전달한다. client_hello는 통신에 사용할 보안 방식을 협상하기 위해 클라이언트의 지원 정보를 알리는 메시지이다.

개념 구분

client_hello의 역할: 문항에서 다루는 TLS 1.2와 같은 기존 핸드셰이크 구조에서 client_hello에는 클라이언트의 난수, 지원하는 암호군(Cipher Suite) 목록, 지원하는 압축 방식 목록이 포함된다. 난수는 이후 키 생성에 쓰이며, 두 목록은 서버가 양측에서 사용할 수 있는 방식을 선택하도록 **지원 가능한 후보를 제시**하는 정보이다.

인증서의 전달: 인증서는 공개키와 그 소유자의 신원을 연결하는 인증 정보로, **client_hello의 구성 항목이 아니다**. 인증서를 사용하는 핸드셰이크에서는 별도의 Certificate 메시지로 전달한다. 클라이언트 인증서도 서버가 인증서를 요청한 경우 이후 단계에서 Certificate 메시지로 보내므로, 클라이언트가 보내는 정보라는 이유만으로 client_hello에 포함되는 것은 아니다.

NoteBase

문 13. (가)와 (나)에 들어갈 내용을 바르게 연결한 것은?

스위치 환경에서 공격자의 스위치 재밍을 통해 스위치의 매핑 테이블의 최대 저장 개수보다 더 많은 정보가 추가되어 스위치가 (가) 모드로 동작하여, 같은 스위치 내에 위치한 공격자는 네트워크 인터페이스 카드를 (나) 모드로 설정하여 스니핑이 가능하다.

① (가): Broadcast, (나): Full-duplex

② (가): Broadcast, (나): Promiscuous

③ (가): Unicast, (나): Full-duplex

④ (가): Unicast, (나): Promiscuous

정답 2번

간단한 해설

스위치 재밍으로 목적지 포트를 찾지 못한 프레임이 **여러 포트로 플러딩**되면 공격자에게도 다른 호스트의 트래픽이 도달할 수 있다. 공격자는 **Promiscuous 모드**로 자신의 MAC 주소를 목적지로 하지 않는 프레임까지 수집한다.

빈칸별 해설

(가) Broadcast: 스위치는 MAC 주소와 포트의 대응 관계를 매핑 테이블에 저장하고, 이를 이용해 프레임을 목적지 포트로 전달한다. 스위치 재밍은 다수의 MAC 주소 정보를 학습시켜 이 테이블을 가득 채우는 공격이다. 테이블 포화로 목적지 MAC 주소에 대응하는 포트를 알 수 없으면, 스위치는 해당 프레임을 같은 VLAN의 수신 포트를 제외한 다른 포트들로 플러딩한다. 문항의 Broadcast 모드는 이러한 확산 전달을 가리킨다. 엄밀히는 프레임의 목적지 주소가 브로드캐스트 주소로 바뀌는 것이 아니라, **목적지를 모르는 유니캐스트 프레임이 플러딩**되는 것이다. Unicast를 목적지 포트로만 전달하는 동작으로 이해하면 공격자가 다른 호스트의 트래픽을 받는 상황을 설명할 수 없다.

(나) Promiscuous: 무차별 모드로, 네트워크 인터페이스 카드에 도달한 프레임을 **자신의 MAC 주소가 목적지가 아니어도** 캡처할 수 있도록 한다. 따라서 스위치의 플러딩으로 공격자 포트에 도달한 다른 호스트의 프레임도 수집할 수 있다. 이 모드만으로 스위치가 다른 포트의 트래픽을 보내 주는 것은 아니므로, 문항에서는 스위치 재밍과 함께 사용한다. Full-duplex는 송신과 수신을 동시에 수행하는 통신 방식이며, 목적지 MAC 주소에 따른 수신 필터링을 해제하는 설정이 아니다.

(가)는 Broadcast, (나)는 Promiscuous이므로 ②의 조합에 해당한다.

NoteBase

문 14. 다음은 TCP 3-Way Handshaking 과정의 예를 나타낸 것이다. (가) ~ (라)에 들어갈 숫자를 바르게 연결한 것은? (단, C: Client, S: Server, [ ]: 세그먼트 플래그, seq: 순서 번호, ack: 확인응답 번호)

C → S, [SYN] seq = 102
S → C, [SYN-ACK] seq = (가), ack = (나)
C → S, [ACK] seq = (다), ack = (라)

① (가): 213, (나): 102, (다): 103, (라): 214
② (가): 102, (나): 103, (다): 103, (라): 214
③ (가): 103, (나): 102, (다): 213, (라): 214
④ (가): 330, (나): 103, (다): 103, (라): 331

정답 4번

간단한 해설

SYN은 **순서 번호 하나를 소비**하므로 클라이언트의 다음 순서 번호와 서버의 확인응답 번호는 모두 103이다.
서버의 초기 순서 번호가 330이면 이를 확인하는 ack는 331이다.

NoteBase

계산 과정

데이터가 따로 표시되지 않은 이 연결 설정 과정에서는 SYN이 소비하는 순서 번호만 반영한다. SYN은 데이터가 없어도 순서 번호 하나를 소비하며, ack는 **상대에게서 다음에 받기를 기대하는 순서 번호**를 뜻한다.

(가): 서버가 보내는 SYN-ACK의 seq는 서버 자신의 초기 순서 번호이다. 클라이언트의 초기 순서 번호 102와는 **별도로 정하는 값**이므로, 102에서 계산하지 않고 선택지의 (가)와 (라)가 서로 맞는지 확인한다.

(나): 서버는 클라이언트가 보낸 seq = 102의 SYN을 받았다. SYN이 순서 번호 하나를 소비하므로, 서버가 다음에 기대하는 클라이언트의 순서 번호는 102 + 1 = 103이다. 따라서 SYN-ACK의 ack인 (나)는 103이다.

(다): 클라이언트도 첫 SYN을 보내면서 자신의 순서 번호 하나를 사용했다. 그 뒤 별도의 데이터를 보내지 않았으므로, 세 번째 ACK에 넣는 자신의 seq인 (다)는 102 + 1 = 103이다. **서버의 SYN-ACK을 받았다는 이유로 자신의 seq를 다시 증가시키지는 않는다.**

(라): 클라이언트는 서버가 보낸 SYN을 확인해야 하므로, ack에는 서버의 초기 순서 번호 (가)에 1을 더한 값을 넣는다. 이는 서버가 다음에 사용할 순서 번호를 기대한다는 뜻이다.

(나)와 (다)가 모두 103이고, (라)가 (가)보다 1 커야 한다. 이 조건을 함께 만족하는 것은 ④의 (가) = 330, (나) = 103, (다) = 103, (라) = 331이며, 서버 쪽 관계도 330 + 1 = 331로 성립한다.

NoteBase

문 15. 암호 알고리즘에 대한 설명으로 옳지 않은 것은?

① SEED는 128비트의 블록 암호 알고리즘이며, SEED 128과 SEED 256의 키 길이는 각각 128비트와 256비트이다.

② HIGHT는 경량 환경 및 하드웨어의 효율성 향상을 위해 개발된 128비트의 블록 암호 알고리즘이다.

③ DES는 64비트의 블록 암호 알고리즘으로 16라운드로 구성되어 있다.

④ LEA는 국내에서 개발된 128비트의 블록 암호 알고리즘으로 128비트, 192비트, 256비트의 비밀키를 사용할 수 있다.

정답 2번

간단한 해설

HIGHT의 블록 크기는 **128비트가 아니라 64비트**이다. 128비트는 암호화에 사용하는 **키의 길이**이므로 블록 크기와 구별해야 한다.

선지별 해설

블록 크기는 한 번에 변환하는 데이터 묶음의 길이이고, 키 길이는 암호화·복호화에 사용하는 비밀키의 길이이다. 두 길이가 반드시 같아야 하는 것은 아니다.

① O: SEED 128과 SEED 256은 모두 128비트 블록을 처리하며, 키 길이는 각각 128비트와 256비트이다. 두 명칭의 숫자는 **블록 크기가 아닌 키 길이**를 구분한다.

② X: HIGHT는 저전력·경량 환경에서 효율적으로 구현하도록 개발된 블록 암호이다. 개발 목적에 관한 설명은 맞지만, 블록 크기는 **64비트**이다. HIGHT는 128비트 키로 64비트 평문 블록을 64비트 암호문 블록으로 변환하므로, '128비트의 블록 암호'를 '64비트의 블록 암호'로 고쳐야 한다.

③ O: DES는 64비트 평문 블록을 64비트 암호문 블록으로 변환하는 블록 암호이다. 라운드는 데이터를 변환하는 연산의 반복 단위이며, DES는 **16라운드**로 구성된다.

④ O: LEA는 국내에서 개발된 128비트 블록 암호로, 128비트·192비트·256비트의 비밀키를 지원한다. **키 길이가 달라도 블록 크기는 128비트**로 동일하다.

NoteBase

문 16. HMAC에 대한 설명으로 옳지 않은 것은?

① IP 보안에서 메시지 인증 목적으로 사용된다.

② MD5, SHA-1, RIPEMD-160과 같은 해시 함수를 내장하고 있다.

③ 비밀키를 적용한 블록 암호화와 해시 함수를 사용해서 결과를 산출한다.

④ IEEE 802.11i의 의사 난수 생성 과정에 사용된다.

정답 3번

간단한 해설

HMAC은 **비밀키와 해시 함수**로 메시지 인증값을 만들며, 그 과정에 **블록 암호화가 필요하지 않다**. 해시 함수가 데이터를 블록 단위로 처리하는 것과 블록 암호로 암호화하는 것은 서로 다르다.

선지별 해설

① O: HMAC은 IPsec에서 메시지의 **무결성과 데이터 출처 인증**을 제공하는 데 사용된다. 송신자와 수신자가 공유한 비밀키로 인증값을 계산하고 대조하여, 전송 중 메시지가 바뀌었는지와 그 키를 가진 상대가 만든 메시지인지를 확인한다.

② O: HMAC은 내부 해시 함수로 MD5, SHA-1, RIPEMD-160 등을 사용할 수 있다. **선택한 해시 함수를 비밀키와 결합**하여 HMAC-MD5, HMAC-SHA-1 등으로 구성하며, 열거된 해시 함수를 모두 동시에 사용하는 것은 아니다.

③ X: HMAC은 **블록 암호화가 아니라 비밀키를 결합한 해시 연산**으로 인증값을 산출한다. 길이를 조정한 비밀키에 내부 패드(ipad)를 XOR한 값과 메시지를 이어 붙여 해시한 뒤, 비밀키에 외부 패드(opad)를 XOR한 값에 그 해시 결과를 이어 붙여 다시 해시한다. 해시 함수 내부에서 데이터를 블록 단위로 처리하더라도, 이는 비밀키로 블록 암호화를 수행한다는 뜻이 아니다.

④ O: IEEE 802.11i는 키를 유도하는 **의사 난수 함수(PRF)에 HMAC-SHA-1**을 사용한다. 예를 들어 쌍별 마스터 키(PMK), 양측의 일회성 값인 논스, 단말과 액세스 포인트의 주소를 입력으로 사용하여 쌍별 임시 키(PTK)를 생성한다.

NoteBase

문 17. DNSSEC(DNS Security Extensions)에 대한 설명으로 옳지 않은 것은?

① DNS 응답 메시지에 대한 기밀성을 제공한다.

② DNS 데이터에 공개키 기반 전자서명을 추가하여 데이터의 무결성을 제공한다.

③ DNS 응답 수신자(resolver)는 수신한 데이터에 대한 서명을 검증한다.

④ DNSSEC는 DNS 캐시 포이즈닝 공격을 방어하도록 설계되었다.

정답 1번

간단한 해설

DNSSEC는 전자서명으로 DNS 데이터의 **출처 인증과 무결성**을 제공한다. 응답 내용을 암호화하여 숨기는 기능은 없으므로 **기밀성은 제공하지 않는다**.

선지별 해설

① X: 기밀성은 권한 없는 사람이 데이터 내용을 읽지 못하도록 보호하는 성질이다. DNSSEC는 DNS 데이터에 전자서명을 붙여 출처와 변조 여부를 확인하게 하지만, **응답 내용을 암호화하지는 않는다**. 따라서 '기밀성을 제공한다'는 '데이터 출처 인증과 무결성을 제공한다'로 고쳐야 한다.

② O: DNS 영역의 서명자는 개인키로 DNS 자원 레코드 집합에 전자서명을 생성하고, 검증자는 대응하는 공개키로 이를 검증한다. 서명 후 데이터가 바뀌면 검증에 실패하므로 **데이터의 무결성**을 확인할 수 있다. 이 서명은 RRSIG 레코드에 담긴다.

③ O: DNSSEC 검증을 수행하는 리졸버는 수신한 DNS 데이터와 서명을 대조하여 **전자서명의 유효성을 검증**한다. 이때 서명 검증에 쓰는 공개키도 신뢰 앵커에서 이어지는 신뢰 관계를 통해 확인해야 한다. 검증 결과로 데이터가 신뢰할 수 있는 출처에서 왔으며 변조되지 않았는지 판단한다.

④ O: DNS 캐시 포이즈닝은 위조한 DNS 정보를 리졸버의 캐시에 넣어 사용자를 잘못된 주소로 유도하는 공격이다. DNSSEC는 서명된 영역의 응답을 검증할 때 **유효한 서명을 제시하지 못하는 위조 데이터**를 정상적인 데이터로 받아들이지 않게 하므로 이 공격을 방어하도록 설계되었다. 보호 효과를 얻으려면 해당 데이터가 서명되어 있고 리졸버가 이를 검증해야 한다.

NoteBase

문 18. 하트블리드(Heartbleed)에 대한 다음 설명의 (가)와 (나)에 들어갈 내용을 바르게 연결한 것은?

TLS를 구현하기 위한 오픈 소스인 OpenSSL에 내재된 버그로, (가) 프로토콜의 요청을 받은 컴퓨터에서 적절한 (나)이 이루어지지 않아서 피해자가 의도하지 않은 내용이 응답 패킷의 페이로드에 복사되어 공격자에게 전송될 수 있다.

① (가): Handshake, (나): 페이로드 길이 확인

② (가): Handshake, (나): 사용자 인증

③ (가): Heartbeat, (나): 페이로드 길이 확인

④ (가): Heartbeat, (나): 사용자 인증

정답 3번

간단한 해설

하트블리드는 **Heartbeat** 요청에 적힌 페이로드 길이를 실제 수신 데이터와 대조하지 않아 발생한다. 이 때문에 요청 데이터의 범위를 넘는 메모리 내용까지 응답에 포함될 수 있으므로, 핵심은 **페이로드 길이 확인**이다.

빈칸별 해설

(가) Heartbeat: TLS 연결의 상대방이 응답하는지 확인하는 기능이다. 정상적인 응답은 요청에 담긴 페이로드를 그대로 돌려보낸다. 하트블리드는 이 **Heartbeat 요청의 처리 과정**에 존재하는 OpenSSL 구현 버그이다. Handshake는 암호 통신에 필요한 매개변수와 키 등을 설정하는 절차이므로 이 취약점이 발생한 프로토콜과 구별된다.

(나) 페이로드 길이 확인: 요청에 표시된 길이가 실제 수신 메시지의 범위 안에 있는지 검사해야 한다. 취약한 구현은 이 검사를 빠뜨려, 공격자가 실제 보낸 데이터보다 큰 길이를 지정하면 **수신 데이터의 범위를 넘어 메모리를 읽고** 그 내용까지 응답에 복사할 수 있다. 따라서 문제의 원인은 사용자 인증의 누락이 아니라 페이로드 길이에 대한 경계 검사의 누락이다.

(가)에 Heartbeat, (나)에 페이로드 길이 확인을 넣은 조합은 ③과 일치한다.

NoteBase

문 19. SSH(Secure Shell)에 대한 설명으로 옳지 않은 것은?

① SSH는 안전한 원격 로그온 기능을 제공하기 위해 설계되었다.

② 사용자 인증 프로토콜을 통해 사용자가 서버에게 인증받는다.

③ 포트 전달을 통해 임의의 안전하지 않은 TCP 연결을 안전한 SSH 연결로 변환시킬 수 있다.

④ 전송 계층 프로토콜은 하나의 SSH 연결을 여러 개의 논리적 통신 채널로 다중화한다.

정답 4번

간단한 해설

④의 논리적 채널 다중화는 **연결 프로토콜**의 역할이다. SSH 전송 계층 프로토콜은 **서버 인증·기밀성·무결성**을 제공하여 통신을 보호한다.

선지별 해설

① O: SSH는 안전하지 않은 네트워크에서도 **안전한 원격 로그인**과 명령 실행 등을 제공하기 위한 프로토콜이다. 통신 내용을 암호화하고 변조 여부를 검출하여 원격 접속 과정에서 주고받는 정보를 보호한다.

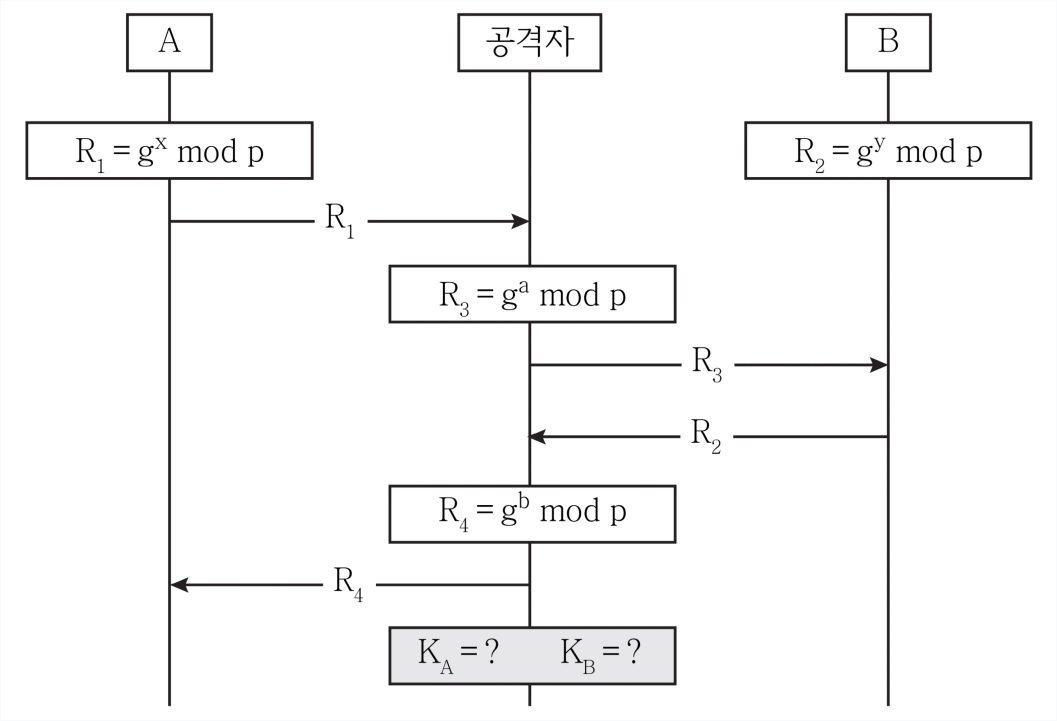
② O: 사용자 인증 프로토콜은 **클라이언트 측 사용자를 서버에 인증**한다. 서버가 접속하려는 사용자의 신원을 확인하는 절차이며, 클라이언트가 접속 대상 서버를 확인하는 서버 인증과는 방향이 다르다. SSH의 서버 인증은 전송 계층 프로토콜에서 수행한다.

③ O: 포트 전달은 일반 TCP 연결의 데이터를 **암호화된 SSH 터널 안으로 전달**하여 보호하는 기능이다. 기존 응용 프로토콜 자체에 암호화 기능이 없어도 SSH 연결을 통과하는 구간에서는 데이터를 안전하게 전송할 수 있다. 다만 SSH 터널 밖의 구간까지 자동으로 암호화되는 것은 아니다.

④ X: 하나의 SSH 연결에서 여러 논리적 통신 채널을 다중화하는 주체는 **연결 프로토콜(Connection Protocol)**이다. 예를 들어 원격 셸과 포트 전달을 서로 다른 채널로 구분하면서 하나의 SSH 연결을 공유하게 한다. 선지의 '전송 계층 프로토콜'을 '연결 프로토콜'로 고쳐야 한다. 전송 계층 프로토콜은 서버 인증·키 교환·암호화·무결성 보호 등을 담당하여 상위 프로토콜이 사용할 안전한 통신 기반을 제공한다.

NoteBase

문 20. 다음은 사용자 A와 B가 Diffie-Hellman 알고리즘에 의한 키 교환을 시도하려는 도중에 발생한 중간자 공격의 개념을 도시한 것이다. 공격자가 A와의 공유비밀키 K_A와 B와의 공유비밀키 K_B를 산출하는 계산식을 바르게 연결한 것은? (단, p는 소수, g는 p의 원시근, x, y, a, b는 p보다 작은 임의의 양의 정수)



① K_A: R_1^a p, K_B: R_2^b p

② K_A: R_1^b p, K_B: R_2^a p

③ K_A: R_2^a p, K_B: R_1^b p

④ K_A: R_2^b p, K_B: R_1^a p

정답 2번

간단한 해설

공격자는 가로챈 공개값에 **각 상대에게 사용한 자신의 비밀지수**를 적용한다. A에게 보낸 공개값의 비밀지수는 b, B에게 보낸 공개값의 비밀지수는 a이므로, K_A=R_1^b p, K_B=R_2^a p이다.

NoteBase

계산 과정

Diffie-Hellman에서는 공통의 소수 p와 원시근 g를 사용하며, 받은 공개값을 자신의 비밀지수만큼 거듭제곱한 뒤 p로 나눈 나머지를 공유비밀키로 삼는다. 그림에서 A의 비밀지수는 x, B의 비밀지수는 y이고, 공격자는 두 사람에게 서로 다른 공개값을 보내 각각 키를 공유한다.

A와의 키 계산: 공격자가 A에게 보낸 값은 R_4=g^b p이므로, 이 교환에 사용한 공격자의 비밀지수는 b이다. 공격자는 A로부터 가로챈 R_1=g^x p를 이용해 K_A=R_1^b p를 계산한다. 이를 전개하면 g^xb p이며, A가 받은 값으로 계산하는 R_4^x p도 같은 값이 된다.

B와의 키 계산: 공격자가 B에게 보낸 값은 R_3=g^a p이므로, 이 교환에 사용한 공격자의 비밀지수는 a이다. 공격자는 B로부터 가로챈 R_2=g^y p를 이용해 K_B=R_2^a p를 계산한다. 이를 전개하면 g^ya p이며, B가 받은 값으로 계산하는 R_3^y p도 같은 값이 된다.

따라서 공격자가 계산하는 식은 K_A=R_1^b p, K_B=R_2^a p이며, 이 연결은 ②와 일치한다.

NoteBase