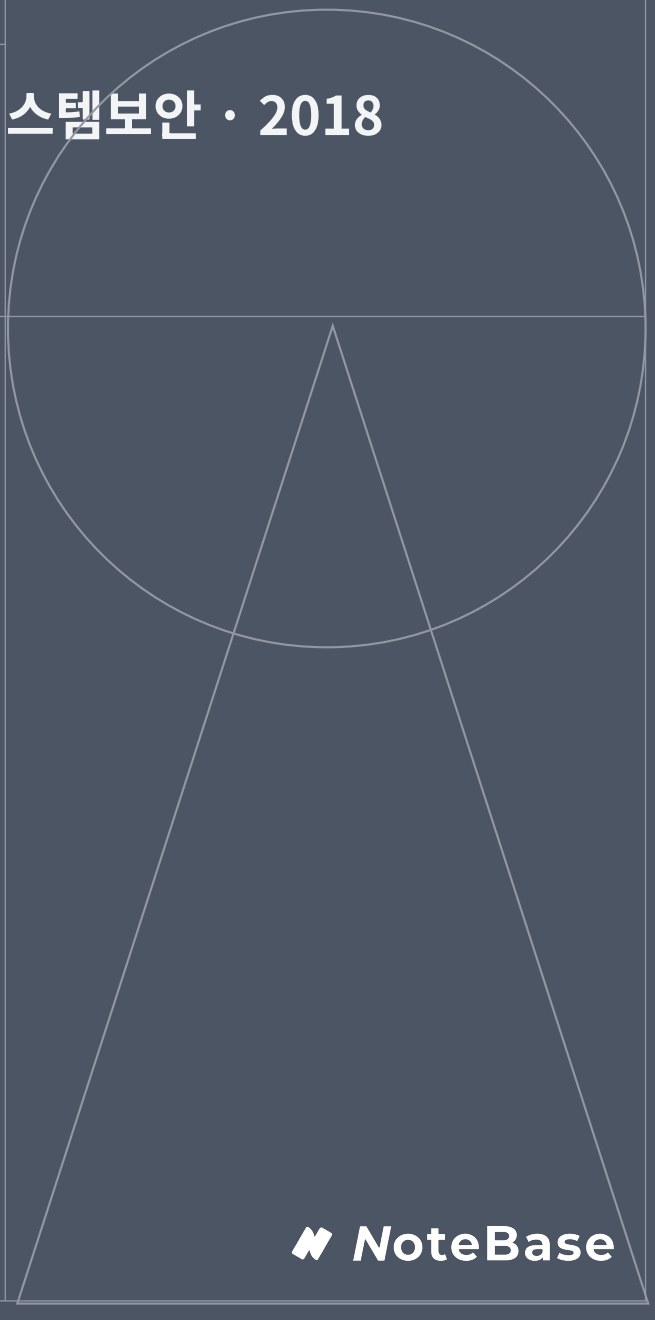
기출OX 해설집

공무원/국가직/9급 · 정보시스템보안 · 2018



NoteBase

기출OX 해설집

공무원/국가직/9급 · 정보시스템보안 · 2018

공무원/국가직/9급 · 정보시스템보안 · 2018

문 1. cookie 정보를 불법적으로 활용하여 사용자의 세션을 탈취하기 위해 사용되는 공격은?

① IP 스푸핑

② 버퍼 오버플로우

③ Cross Site Scripting

④ DoS

정답 3번

간단한 해설

XSS는 사용자의 브라우저에서 악성 스크립트를 실행시키는 공격이다. 이를 통해 **스크립트가 읽을 수 있는 세션 쿠키를 탈취**하면 사용자의 로그인 상태를 가로채는 데 악용할 수 있다.

공통 해설

Cross Site Scripting(XSS)은 웹 페이지에 악성 스크립트가 삽입되어 **사용자의 브라우저에서 해당 사이트의 권한으로 실행**되는 공격이다. 쿠키 정보를 이용한 세션 탈취는 이러한 스크립트 실행을 악용하는 대표적인 방식이다.

세션 쿠키에는 서버가 사용자의 세션을 식별하는 값이 담길 수 있다. 공격자가 악성 스크립트로 이 값을 빼낸 뒤 자신의 요청에 재사용하면, 서버가 공격자의 요청을 피해자의 로그인 세션에서 온 요청으로 처리할 수 있다. 다만 이처럼 쿠키 값을 직접 읽어 탈취하려면 **해당 쿠키에 스크립트가 접근할 수 있어야 한다**. HttpOnly 속성이 설정된 쿠키는 자바스크립트로 직접 읽을 수 없으므로, 이 방식의 쿠키 탈취가 제한된다.

NoteBase

문 2. 데이터베이스의 DDL(Data Definition Language) 질의문에 포함되지 않는 것은?

① ALTER DATABASE

② SELECT

③ DROP INDEX

④ CREATE TABLE

정답 2번

간단한 해설

SELECT는 저장된 **데이터를 조회**하는 명령이다. 데이터베이스 객체를 생성·변경·삭제하는 DDL에는 포함되지 않는다.

공통 해설

DDL(데이터 정의어)은 **데이터베이스 객체를 생성·변경·삭제**하는 언어이다. CREATE TABLE은 테이블을 생성하고, ALTER DATABASE는 데이터베이스의 속성을 변경하며, DROP INDEX는 인덱스를 삭제하므로 모두 DDL에 해당한다. 여기서 DROP INDEX는 테이블의 데이터 행을 지우는 것이 아니라 인덱스라는 객체를 제거하는 명령이다.

SELECT는 테이블 등에 저장된 데이터에서 필요한 내용을 조회하는 명령이다. **객체의 정의를 바꾸는 것이 아니라 데이터를 읽어 결과를 반환**하므로 DDL에 포함되지 않는다.

NoteBase

문 3. 공개된 웹 방화벽에 해당하는 것만을 모두 고른 것은?

ㄱ. WebKnight
ㄴ. ModSecurity
ㄷ. Snort
ㄹ. Nmap

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄷ, ㄹ

정답 1번

간단한 해설

WebKnight와 ModSecurity는 **웹 요청을 검사하여 악성 요청을 차단하는 웹 방화벽**이다. 공개된 보안 도구라도 네트워크 침입 탐지나 포트 탐색이 주된 기능이면 웹 방화벽으로 분류하지 않는다.

보기별 해설

웹 방화벽(WAF)은 HTTP 요청 등 웹 통신의 내용을 규칙에 따라 검사하여 웹 애플리케이션을 겨냥한 공격을 탐지·차단하는 보안 도구이다.

ㄱ. WebKnight: 공개 소스 웹 방화벽이다. IIS 웹 서버에서 ISAPI 필터로 동작하며, **웹 요청을 필터링하여** SQL 삽입이나 디렉터리 탐색 등의 공격을 차단한다.

ㄴ. ModSecurity: 공개 소스 웹 방화벽이다. **HTTP 트래픽을 검사하고 보안 규칙을 적용**하여 SQL 삽입, 크로스 사이트 스크립팅(XSS) 등 웹 애플리케이션 공격에 대응한다.

ㄷ. Snort: **네트워크 기반 침입 탐지·방지 시스템(IDS/IPS)**이다. 네트워크 패킷을 분석하여 규칙에 일치하는 악성 활동을 탐지하고, 인라인으로 배치하면 차단할 수도 있다. 웹 공격 탐지에 활용될 수 있지만 도구의 분류는 웹 방화벽과 구별된다.

ㄹ. Nmap: **네트워크 탐색 및 포트 스캔 도구**이다. 호스트의 응답과 열린 포트 등을 조사하여 네트워크 상태와 제공 서비스를 파악한다. 웹 요청을 필터링하여 웹 애플리케이션을 보호하는 웹 방화벽에 해당하지 않는다.

따라서 공개된 웹 방화벽에 해당하는 것은 ㄱ과 ㄴ이며, 이 조합은 ①에 해당한다.

NoteBase

문 4. OSI 7계층의 응용 계층과 관련된 공격은?

① Ping of Death
② SYN flooding
③ ARP spoofing
④ DNS spoofing

정답 4번

간단한 해설

DNS는 도메인 이름에 대응하는 IP 주소 등을 알려 주는 **응용 계층 프로토콜**이다. DNS spoofing은 이 이름 해석 과정에서 거짓 정보를 제공하여 사용자가 잘못된 주소로 접속하도록 유도하는 공격이다.

개념 구분

공격이 직접 악용하는 프로토콜과 기능을 기준으로 계층을 구분한다. DNS spoofing은 **응용 계층의 이름 해석 기능**을 속이는 공격이다. DNS 응답을 위조하는 등의 방법으로 도메인 이름에 잘못된 IP 주소를 대응시켜, 정상 사이트에 접속하려는 사용자를 공격자가 지정한 곳으로 유도할 수 있다. DNS 메시지가 UDP나 TCP로 운반되더라도, 공격 대상인 DNS의 이름 해석 기능은 응용 계층에 속한다.

Ping of Death는 비정상적으로 큰 패킷이 재조립되도록 조작한 ICMP 패킷 등으로 대상 시스템의 장애를 유발하는 공격이다. ICMP와 IP 패킷 처리를 악용하므로 **네트워크 계층**과 관련된다.

SYN flooding은 TCP 연결 설정 요청인 SYN을 대량으로 보내 서버에 미완료 연결 상태가 쌓이게 하는 공격이다. **전송 계층의 TCP 연결 설정 과정**을 악용한다.

ARP spoofing은 같은 로컬 네트워크에서 거짓 ARP 정보를 보내 IP 주소에 대응하는 MAC 주소를 속이는 공격이다. 도메인 이름을 해석하는 DNS와 달리, **로컬 통신에 사용할 MAC 주소의 대응 관계**를 조작하므로 데이터 링크 계층과 관련된 공격으로 구분한다.

NoteBase

문 5. 다음에서 설명하는 웹 공격 기술 유형은?

방화벽이 존재하는 시스템을 공격할 때 자주 사용된다. 일반적으로 웹 서버는 방화벽 내부에 존재하고, 80번 포트를 이용한 웹 서비스만을 제공하면 되기 때문에 방화벽에서의 인바운드 정책은 80번 포트와 필요한 포트만 빼고 다 막아 놓고, 아웃바운드 정책은 별다른 필터링을 수행하지 않는 경우가 많다. 이 웹 공격 기술은 바로 이런 허점을 이용한다.

① 인증 우회
② 리버스 텔넷
③ 패킷 변조
④ LAND

정답 2번

간단한 해설

리버스 텔넷은 **내부 서버가 외부 공격자 쪽으로 연결을 시작**하게 하여 원격 명령 통로를 만드는 기법이다. 외부에서 들어오는 연결을 막아도 아웃바운드 연결을 제한하지 않으면 이 통로가 형성될 수 있다.

공통 해설

리버스 텔넷은 공격 대상 서버에서 텔넷을 이용해 외부 공격자 시스템으로 연결하게 하고, 그 연결을 통해 명령과 실행 결과를 주고받는 기법이다. 공격자가 서버에서 관련 프로그램이나 명령을 실행할 수 있어야 하며, 해당 외부 연결이 방화벽에서 허용되어야 한다. 핵심은 **연결을 시작하는 쪽이 방화벽 내부의 서버**라는 점이다.

지문에서는 인바운드 연결을 80번 포트와 필요한 포트로 제한하지만, 아웃바운드에는 별다른 필터링을 하지 않는다. 따라서 웹 서버에서 외부로 시작하는 연결을 이용하면, 외부에서 내부로 새 연결을 시도할 때 받는 제한을 우회하여 명령 통로를 만들 수 있다. 이처럼 연결 방향과 아웃바운드 정책의 허점을 이용하는 공격이 리버스 텔넷이다. 80번 포트는 지문에서 웹 서비스용으로 허용한 포트이며, 리버스 텔넷의 외부 연결이 반드시 80번 포트를 사용한다는 뜻은 아니다.

NoteBase

문 6. ㉠~㉢에 들어갈 리눅스 부팅 순서를 바르게 연결한 것은?

POST (Power on Self Test) 수행 → (㉠) → (㉡) → (㉢) → 실행 레벨에 따른 서비스 실행

① ㉠: 부트 로더 실행, ㉡: 기본 부팅 관련 설정 사항 로드, ㉢: MBR 로드
② ㉠: 기본 부팅 관련 설정 사항 로드, ㉡: MBR 로드, ㉢: 부트 로더 실행
③ ㉠: MBR 로드, ㉡: 부트 로더 실행, ㉢: 기본 부팅 관련 설정 사항 로드
④ ㉠: 부트 로더 실행, ㉡: MBR 로드, ㉢: 기본 부팅 관련 설정 사항 로드

정답 2번

간단한 해설

부팅 장치를 찾으려면 먼저 **기본 부팅 설정**을 읽어야 한다. 선택한 장치의 **MBR을 메모리에 로드한 뒤** 그 안의 부팅 코드를 실행하여 부트 로더로 부팅을 이어 간다.

올바른 순서

POST 수행 → ㉠ 기본 부팅 관련 설정 사항 로드 → ㉡ MBR 로드 → ㉢ 부트 로더 실행 → 실행 레벨에 따른 서비스 실행

POST 수행: 문제는 BIOS와 MBR을 사용하는 전통적인 리눅스 부팅 과정을 전제로 한다. 전원을 켜면 BIOS가 메모리 등 주요 하드웨어의 이상 유무를 검사하여 부팅을 진행할 수 있는지 확인한다.

㉠ 기본 부팅 관련 설정 사항 로드: BIOS가 부팅 장치의 우선순위 등 기본 설정을 읽는다. 이 설정에 따라 **어느 장치에서 부팅할지** 결정해야 그 장치의 부팅 정보를 읽을 수 있다.

㉡ MBR 로드: 선택한 디스크의 첫 번째 섹터에 있는 MBR(Master Boot Record)을 메모리에 읽어 들인다. MBR에는 초기 부팅 코드와 파티션 정보가 있으며, 부팅 코드를 실행하려면 **먼저 메모리에 적재**해야 한다.

㉢ 부트 로더 실행: BIOS가 메모리에 적재한 부팅 코드로 제어를 넘기면 부트 로더의 실행이 시작된다. 부트 로더는 필요한 후속 단계를 거쳐 **리눅스 커널을 메모리에 적재하고 제어를 넘긴다**.

실행 레벨에 따른 서비스 실행: 커널이 시스템을 초기화하고 init을 실행하면, init이 지정된 실행 레벨에 맞는 서비스를 시작한다. 따라서 실행 레벨에 따른 서비스 실행은 **커널과 초기화 프로세스가 실행된 이후**에 이루어진다.

NoteBase

문 7. 리눅스 계열 시스템에서 /bin/sh의 정보를 출력한 파일에 대한 설명으로 옳지 않은 것은?

-r-Sr-xr-x root sys 31508 2017년 7월 21일 /bin/sh

① 이 파일은 모든 사용자가 실행권한을 가지고 있다.

② 이 파일의 소유자는 해당 파일을 읽거나 실행시킬 수 있다.

③ 이 파일에는 setgid가 설정되어 있다.

④ 이 파일을 실행 중인 일반 사용자는 root 권한을 가질 수 있다.

정답 3번

간단한 해설

소유자 실행 위치의 'S'는 **setuid가 설정**되어 있지만 **소유자 실행 비트는 꺼져 있음**을 뜻한다. setgid는 소유 그룹의 실행 위치에 's' 또는 'S'로 표시하므로, 제시된 파일에는 setgid가 설정되어 있지 않다.

선지별 해설

권한 문자열 '-r-Sr-xr-x'는 일반 파일을 뜻하는 '-', 소유자 권한 r-S, 소유 그룹 권한 r-x, 기타 사용자 권한 r-x로 나누어 읽는다. 파일 소유자는 root이고 소유 그룹은 sys이다.

① O: 소유 그룹과 기타 사용자의 권한은 모두 r-x이므로 일반 사용자는 실행할 수 있다. 소유자 root의 실행 비트는 꺼져 있지만, root는 관리자 권한으로 실행 권한 검사를 우회할 수 있다. 이때 일반 파일에는 **적어도 하나의 실행 비트가 설정**되어 있어야 하는데, 이 파일은 그룹과 기타 사용자의 실행 비트가 켜져 있어 그 조건을 충족한다.

② O: 소유자 root의 권한 r-S에서 r은 읽기 권한을 뜻한다. 대문자 S이므로 소유자 실행 비트 자체는 꺼져 있지만, **소유자가 root**이고 다른 사용자 범주의 실행 비트가 켜져 있으므로 관리자 권한으로 실행할 수 있다. 소유자가 일반 사용자였다면 r-S만으로 실행할 수 있다는 뜻은 아니다.

③ X: 특수 권한은 s 또는 S가 나타난 위치로 구별한다. 제시된 S는 소유자 권한의 실행 위치에 있으므로 **setgid가 아니라 setuid**가 설정된 것이다. 소유 그룹의 권한은 r-x로 표시되어 있으며, 그 실행 위치에는 s나 S가 없으므로 setgid는 설정되어 있지 않다.

④ O: setuid가 적용되는 프로그램을 실행하면 실행 프로세스의 **유효 사용자 ID가 파일 소유자의 ID**로 바뀐다. 이 파일의 소유자는 root이므로 일반 사용자가 실행해도 프로세스가 root 권한을 가질 수 있다. 이는 사용자의 계정 자체가 root로 바뀐다는 뜻은 아니다. 다만 nosuid로 마운트된 파일시스템에서는 setuid가 적용되지 않으며, 셸이 실행 후 스스로 권한을 낮추는 경우에도 root 권한이 계속 유지되지는 않는다.

NoteBase

문 8. 리눅스에서 지금 어떤 프로세스가 실행 중인지를 확인하기 위해 ps라는 명령어를 사용한다. 프로세스를 실행할 때, 백그라운드(background)와 포그라운드(foreground) 프로세스에 대한 설명으로 옳지 않은 것은?

① 포그라운드 프로세스는 키보드로부터 입력을 받아서 결과를 직접 화면에 출력한다.

② 백그라운드 프로세스를 실행시키기 위해서는 명령어 뒤에 &를 붙여야 한다.

③ 포그라운드 프로세스는 일반적으로 한 명령어가 실행되는 동안 다른 명령어를 실행시킬 수 없다.

④ 백그라운드 프로세스는 현재 터미널에서 무엇을 하고 있는지와 상관없이 결과를 그대로 화면에 출력할 수 없다.

정답 4번

간단한 해설

백그라운드 프로세스도 표준 출력이 터미널에 연결되어 있으면 **화면에 결과를 출력할 수 있다**. 따라서 사용자가 다른 명령을 입력하거나 실행하는 중에도 그 출력이 화면에 섞여 나타날 수 있다.

선지별 해설

① O: 포그라운드 프로세스는 터미널에서 사용자와 상호작용하며 실행된다. 입출력을 별도로 바꾸지 않은 일반적인 경우에는 **키보드로 입력받고 화면으로 출력**할 수 있다. 다만 모든 포그라운드 프로그램이 반드시 입력을 요구하는 것은 아니며, 리다이렉션을 사용하면 파일에서 입력받거나 파일로 출력할 수도 있다.

② O: 명령을 처음부터 백그라운드로 실행하려면 **명령어 끝에 &**를 붙인다. 예를 들어 sleep 10 &를 실행하면 셸은 해당 작업이 끝나기를 기다리지 않고 다음 명령을 받을 수 있다. 선지는 이러한 실행 형식을 설명한 것이다. 이미 포그라운드에서 실행 중인 작업은 Ctrl+Z로 중지한 뒤 bg로 백그라운드에서 재개할 수도 있으므로, &가 백그라운드 실행으로 전환하는 유일한 방법이라는 뜻은 아니다.

③ O: 대화형 셸은 일반적으로 **포그라운드 명령이 끝날 때까지 기다린다**. 따라서 그 명령이 실행 중인 동안에는 같은 셸의 프롬프트에서 다음 명령을 실행할 수 없다. 이는 다른 터미널이나 이미 실행 중인 백그라운드 작업까지 실행되지 못한다는 뜻은 아니다.

④ X: 백그라운드로 실행하더라도 표준 출력이 터미널에 연결되어 있으면 **결과를 그대로 화면에 출력할 수 있다**. 셸이 다음 명령을 받는 동안에도 해당 작업의 출력이 나타날 수 있으므로, 선지의 '출력할 수 없다'를 '출력할 수 있다'로 고쳐야 한다. 다만 출력을 파일 등으로 리다이렉션했거나 stty tostop으로 백그라운드 작업의 터미널 출력을 제한한 경우에는 달라진다.

NoteBase

문 9. AES(Advanced Encryption Standard) 암호화 라운드 수행 시 마지막 라운드를 제외하고, 각 라운드에서 수행되는 4단계 처리 순서가 옳은 것은?

① Add round key → Substitute bytes → Shift rows → Mix columns

② Add round key → Substitute bytes → Mix columns → Shift rows

③ Substitute bytes → Shift rows → Mix columns → Add round key

④ Substitute bytes → Mix columns → Shift rows → Add round key

정답 3번

간단한 해설

AES는 바이트를 치환한 뒤 **행을 이동하고 열을 혼합**하여 변화가 퍼지게 한 다음, 라운드 키를 결합한다. 암호화 시작 시 한 번 수행하는 초기 키 더하기와 달리, 각 라운드의 **Add round key는 마지막 단계**이다.

올바른 순서

Substitute bytes → Shift rows → Mix columns → Add round key

Substitute bytes(바이트 치환): 처리 중인 데이터를 나타내는 상태 배열의 각 바이트를 S-box라는 치환표에 따라 다른 값으로 바꾼다. 이 비선형 치환 결과가 다음 행 이동 단계의 입력이 된다.

Shift rows(행 이동): 치환된 상태 배열에서 각 행의 바이트를 행마다 서로 다른 칸 수만큼 왼쪽으로 순환 이동한다. 첫 행은 그대로 둔다. 이 과정에서 바이트가 속한 열이 달라지므로, **열을 혼합하기 전에** 바이트를 재배치하는 역할을 한다.

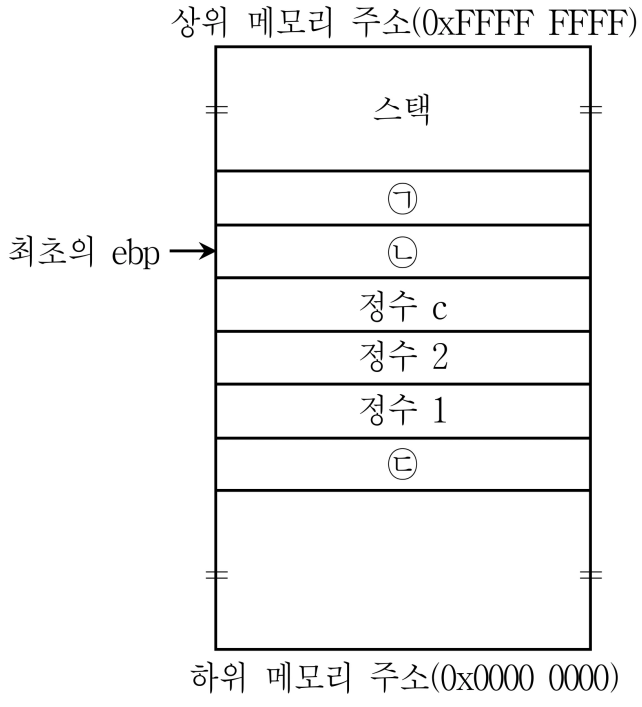
Mix columns(열 혼합): 행 이동이 끝난 상태 배열에서 각 열의 네 바이트를 정해진 선형 변환으로 혼합한다. 앞 단계에서 서로 다른 열로 이동한 바이트들이 새 열의 다른 바이트들과 함께 섞여 변화가 확산된다. 마지막 라운드에서는 이 단계를 생략하지만, 문제에서 묻는 나머지 라운드에는 포함한다.

Add round key(라운드 키 더하기): 열 혼합 결과와 해당 라운드 키를 비트 단위 **XOR(배타적 논리합)**로 결합한다. 라운드 키는 원래 암호 키를 확장하여 생성한다. 첫 라운드에 들어가기 전에도 초기 키 더하기를 한 번 수행하지만, 그것을 매 라운드의 첫 단계로 보는 것은 잘못이다. 각 라운드에서는 앞선 변환을 마친 상태에 키를 결합하며, 마지막 라운드도 열 혼합만 생략하고 이 키 결합으로 마무리한다.

NoteBase

문 10. 다음은 C 프로그램 소스의 일부분이다. x86시스템에서 프로그램 내의 function 함수를 호출시 스택 프레임의 값 ㉠~㉢에 들어갈 용어로 옳게 짝지은 것은?

```
void main() {
    int c;
    c=function(1,2);
}
int function(int a, int b) {
    a=a+b;
    return a;
}
```



① ㉠: 저장된 프레임 포인터, ㉡: function 함수의 반환주소, ㉢: 반환주소

② ㉠: function 함수의 반환주소, ㉡: 저장된 프레임 포인터, ㉢: 반환주소

③ ㉠: 저장된 프레임 포인터, ㉡: 반환주소, ㉢: function 함수의 반환주소

④ ㉠: 반환주소, ㉡: 저장된 프레임 포인터, ㉢: function 함수의 반환주소

정답 4번

NoteBase

간단한 해설

함수를 호출할 때 저장하는 반환주소와, 프레임을 만들 때 보관하는 이전 ebp는 서로 다르다. 그림의 '최초의 ebp'가 가리키는 ㉡은 **저장된 프레임 포인터**이고, 인수 1 아래의 ㉢은 **function에서 main으로 돌아갈 주소**이다.

빈칸별 해설

그림은 ebp를 프레임의 기준으로 사용하는 x86 스택 구조를 나타낸다. 스택은 그림의 위쪽인 높은 주소에서 아래쪽인 낮은 주소로 자란다. main의 지역변수 c보다 아래에 놓인 정수 2와 정수 1은 function에 전달하는 인수이다.

㉠: 반환주소. main의 프레임에서 저장된 ebp보다 높은 주소에 있는 칸이다. **main이 실행을 마친 뒤 돌아갈 위치**를 저장하며, 이후에 이루어지는 function 호출의 반환주소와 구별된다.

㉡: 저장된 프레임 포인터. main의 프레임을 만들 때 이전 ebp 값을 스택에 보관하고, ebp가 그 저장 위치를 가리키게 한다. 그림의 **'최초의 ebp' 화살표가 가리키는 칸**이므로 저장된 프레임 포인터에 해당한다. 이 값은 main의 프레임을 해제할 때 이전 프레임의 기준을 복원하는 데 사용한다.

㉢: function 함수의 반환주소. 그림에서는 인수 2와 1을 스택에 넣은 뒤 function을 호출하므로, 인수 1보다 낮은 주소에 호출 후 복귀할 주소가 저장된다. 이는 **function 실행을 마친 뒤 main에서 실행을 이어 갈 위치**이며, function의 시작 주소나 계산한 반환값을 뜻하지 않는다.

㉠은 반환주소, ㉡은 저장된 프레임 포인터, ㉢은 function 함수의 반환주소이므로 ④의 조합과 일치한다.

NoteBase

문 11. VPN(Virtual Private Network)의 터널링 모드 수립 시 사용되는 암호화 프로토콜이 아닌 것은?

① NNTP

② IPSec

③ L2TP

④ PPTP

정답 1번

간단한 해설

NNTP는 **뉴스 기사를 배포·조회·게시하는 응용 계층 프로토콜**로, VPN 터널을 수립하는 프로토콜이 아니다. 다만 VPN의 터널링과 암호화는 서로 다른 기능이므로, 터널링 프로토콜이 모두 자체 암호화 기능을 갖는 것은 아니다.

선지별 해설

터널링은 원래 패킷이나 프레임을 다른 프로토콜의 데이터 부분에 넣어 전송하는 기술이고, 암호화는 데이터 내용을 알아볼 수 없도록 변환하는 기술이다. 문항의 '암호화 프로토콜'이라는 표현을 모든 후보가 자체적으로 암호화한다는 뜻으로 받아들이면 안 된다. 여기서는 VPN 터널링 및 그 보안 구성에 쓰이는 프로토콜인지 구별한다.

① X: NNTP(Network News Transfer Protocol)는 뉴스 서버 사이에서 기사를 교환하거나 클라이언트가 기사를 읽고 게시하는 데 사용한다. **뉴스 기사 전송용 응용 계층 프로토콜**이며, VPN 터널을 구성하거나 터널 내부의 패킷을 암호화하는 역할을 하지 않는다.

② O: IPSec은 IP 계층에서 통신을 보호하는 보안 프로토콜 체계이다. 터널 모드에서는 **원래 IP 패킷 전체를 캡슐화**하여 새로운 IP 헤더를 붙인다. ESP의 암호화 기능을 사용하면 원래 IP 헤더와 데이터를 함께 보호할 수 있어 VPN 구성에 이용된다.

③ O: L2TP(Layer Two Tunneling Protocol)는 PPP 프레임을 다른 네트워크를 통해 전달하는 2계층 터널링 프로토콜로, VPN 터널 구성에 사용된다. 다만 **L2TP 자체는 사용자 데이터의 암호화를 제공하지 않으므로**, 암호화가 필요하면 IPSec과 결합한 L2TP/IPSec 등의 방식으로 구성한다.

④ O: PPTP(Point-to-Point Tunneling Protocol)는 PPP 프레임을 IP 네트워크를 통해 전달하여 VPN 터널을 구성하는 프로토콜이다. GRE를 이용해 PPP 프레임을 캡슐화하며, 데이터 암호화에는 MPPE 같은 별도 기능을 사용한다. 따라서 **터널을 만드는 PPTP와 데이터를 암호화하는 기능**은 구별해야 한다.

NoteBase

문 12. 리눅스 시스템의 네트워크 관리 도구 및 서비스에 대한 설명으로 옳지 않은 것은?

① ifconfig — 네트워크 인터페이스의 IP 주소 설정

② traceroute — 최종 목적지 컴퓨터까지 중간에 거치는 여러 개의 라우터에 대한 경로 및 응답속도를 표시

③ fping — 네트워크 연결 상태, 라우팅 테이블, 인터페이스 관련 통계 정보 출력

④ tcpdump — 네트워크 모니터링 및 패킷 분석을 위해 사용되는 도구로, 패킷 필터 기능을 통해서, 특정 침입자의 침입 경로에 따라 원하는 트래픽만을 감시

정답 3번

간단한 해설

③의 연결 상태·라우팅 테이블·인터페이스 통계 출력은 **netstat**의 기능이다. **fping**은 여러 호스트에 ICMP Echo Request를 보내 응답 여부를 확인하는 도구이다.

선지별 해설

① O: ifconfig는 네트워크 인터페이스의 설정을 조회하거나 변경한다. 특정 인터페이스에 **IP 주소를 설정**할 수 있으며, 넷마스크와 인터페이스의 활성화·비활성화 상태 등도 설정한다.

② O: traceroute는 탐사 패킷의 TTL을 단계적으로 늘려 목적지까지 거치는 라우터를 확인한다. 각 단계에서 응답한 라우터의 주소와 **탐사 패킷의 왕복 시간**을 표시하므로 경로와 응답 지연을 파악할 수 있다. 여기서 응답속도는 데이터 전송률이 아니라 응답에 걸린 시간을 뜻한다.

③ X: 네트워크 연결 상태, 라우팅 테이블, 인터페이스 통계를 출력하는 도구는 fping이 아니라 **netstat**이다. netstat는 옵션에 따라 연결 상태를 조회하고, -r로 라우팅 테이블을, -i로 인터페이스 통계를 출력한다. fping은 여러 호스트에 ICMP Echo Request를 보내 각 호스트의 응답 여부를 확인하며, 옵션에 따라 응답 시간과 패킷 손실 등도 확인할 수 있다.

④ O: tcpdump는 네트워크 인터페이스에서 패킷을 캡처하고 그 내용을 출력하여 트래픽을 분석하는 도구이다. **출발지·목적지 주소, 포트, 프로토콜 등의 필터 조건**을 지정하면 관심 있는 트래픽만 골라 감시할 수 있다. 침입에 관련된 주소나 통신 조건을 알고 있다면 해당 조건에 맞는 패킷을 수집하여 분석한다.

NoteBase

문 13. TCP 대신에 UDP 69번 포트를 사용하고, 사용자 인증 절차를 요구하지 않기 때문에 누구나 호스트로부터 파일을 가져갈 수 있도록 고안된 프로토콜은?

① HTTP

② FTP

③ TFTP

④ SOAP

정답 3번

간단한 해설

TFTP는 UDP를 이용하는 간단한 파일 전송 프로토콜로, 최초 요청을 **서버의 69번 포트**로 보내며 사용자 인증 절차를 두지 않는다.

공통 해설

TFTP(Trivial File Transfer Protocol)는 파일 읽기·쓰기를 위한 간단한 파일 전송 프로토콜이다. TCP 대신 UDP를 사용하고, **최초 읽기·쓰기 요청을 서버의 69번 포트로 보낸다**. 이후에는 서버와 클라이언트가 해당 전송에 사용할 포트로 데이터를 교환하므로, 모든 데이터가 계속 서버의 69번 포트를 거치는 것은 아니다.

TFTP에는 **사용자 이름과 비밀번호를 확인하는 인증 절차가 없다**. 따라서 문제에 제시된 전송 방식·포트 번호·인증 특성은 TFTP에 대응한다. 다만 인증 절차가 없다는 것이 서버의 모든 파일을 무조건 가져갈 수 있다는 뜻은 아니며, 실제로 전송할 수 있는 파일은 서버의 공개 설정과 접근 권한에 따라 제한된다.

NoteBase

문 14. Windows 레지스트리 키에 대한 설명으로 옳지 않은 것은?

① HKEY_CLASSES_ROOT – 파일의 각 확장자에 대한 정보와 파일과 프로그램 간의 연결에 대한 정보 저장

② HKEY_CURRENT_USER – 윈도우가 설치된 컴퓨터 환경 설정에 대한 정보 저장

③ HKEY_LOCAL_MACHINE – 설치된 하드웨어와 소프트웨어 설치 드라이버 설정에 대한 정보 저장

④ HKEY_USERS – 디스플레이와 프린터에 관한 정보 저장

정답 4번

간단한 해설

HKEY_USERS는 **사용자별 프로필 정보**를 모아 놓은 키이다. ④의 디스플레이·프린터 설명은 **현재 하드웨어 구성**을 나타내는 HKEY_CURRENT_CONFIG와 구분해야 한다.

선지별 해설

① O: HKEY_CLASSES_ROOT에는 파일 확장자와 파일 형식, 해당 형식을 처리할 프로그램의 연결 정보가 들어 있다. 파일을 열 때 **어떤 프로그램으로 처리할지** 결정하는 데 이용된다.

② O: HKEY_CURRENT_USER에는 **현재 로그인한 사용자에게 적용되는 환경 설정**이 저장된다. 화면 색상, 제어판 설정, 사용자 환경 변수 등이 해당한다. 컴퓨터 전체에 공통으로 적용되는 설정은 HKEY_LOCAL_MACHINE과 구별한다.

③ O: HKEY_LOCAL_MACHINE에는 **컴퓨터 전체에 적용되는** 하드웨어·소프트웨어 구성 정보가 저장된다. 설치된 장치, 장치 드라이버와 서비스의 설정 등이 포함되므로 특정 사용자의 개인 설정을 담는 HKEY_CURRENT_USER와 적용 범위가 다르다.

④ X: HKEY_USERS는 컴퓨터에 로드된 **사용자 프로필들의 설정**을 담는 키이며, HKEY_CURRENT_USER는 그중 현재 사용자의 설정을 가리킨다. 디스플레이·프린터 등의 현재 하드웨어 프로필에 관한 구성 정보는 **HKEY_CURRENT_CONFIG**로 설명해야 한다. 화면이나 프린터의 사용자별 기본 설정이 사용자 프로필에 포함될 수는 있지만, 그것이 HKEY_USERS의 전체 역할을 뜻하지는 않는다.

NoteBase

문 15. 다음 보안 운영체제에 대한 설명에서 괄호 안에 들어갈 용어는?

보안커널의 중요한 부분으로 객체에 대한 접근 통제 기능을 수행하고 감사, 식별 및 인증, 보안 매개변수 설정 등과 같은 다른 보안 메커니즘과 데이터를 교환하면서 상호작용을 한다.
주체와 객체 사이에서 비인가된 접속이나 불법적인 자료 변조를 막기 위해 ( )은(는) DB로부터 주체의 접근 권한을 확인하기 위해 사용된다.

① 참조 모니터(Reference Monitor)
② 신뢰 컴퓨팅 베이스(Trusted Computing Base)
③ 로컬 프로시저 호출 관리자(Local Procedure Call Manager)
④ 코어(Core)

정답 1번

간단한 해설

주체가 객체에 접근할 때 **접근 권한을 확인하여 허용·거부를 결정**하는 것은 참조 모니터의 역할이다. 지문의 DB는 이 판단에 필요한 권한 정보를 제공한다.

공통 해설

참조 모니터(Reference Monitor)는 주체와 객체 사이의 접근을 중재하여 접근 통제 정책을 적용하는 개념이다. 주체는 접근을 요청하는 사용자나 프로세스이고, 객체는 보호 대상인 파일이나 데이터 등이다. 참조 모니터는 **요청한 접근이 허가된 것인지** 권한 정보와 대조하여 허용하거나 거부한다. 지문의 'DB로부터 주체의 접근 권한을 확인'하고 '객체에 대한 접근 통제 기능을 수행'한다는 설명이 이에 해당한다.

예를 들어 어떤 사용자에게 파일 읽기만 허용되어 있다면, 그 사용자의 읽기 요청은 허용하고 파일 수정 요청은 거부한다. 사용자의 신원이 확인되었다고 해서 모든 접근을 허용하는 것은 아니며, **요청한 동작에 대한 권한**까지 확인해야 한다. 참조 모니터를 우회하는 접근이 가능하면 이러한 통제를 피할 수 있으므로, 보호 대상에 대한 모든 접근을 중재해야 한다.

NoteBase

문 16. SSL(Secure Socket Layer) 핸드셰이크 프로토콜 처리에서 클라이언트와 서버 사이의 논리 연결을 설립하는 데 필요한 교환 단계를 순서대로 바르게 나열한 것은?

ㄱ. 서버는 인증서, 키 교환을 보내고 클라이언트에게 인증서를 요청한다.
ㄴ. 프로토콜 버전, 세션 ID, 암호 조합, 압축 방법 및 초기 난수를 포함하여 보안 능력을 수립한다.
ㄷ. 암호 조합을 변경한다.
ㄹ. 클라이언트는 인증서와 키 교환을 보낸다.

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

정답 2번

간단한 해설

먼저 사용할 보안 방식을 합의하고 서버와 클라이언트가 인증·키 교환에 필요한 정보를 주고받아야 한다. 그 뒤 **합의한 암호 설정을 실제 통신에 적용**하므로, 클라이언트의 키 교환이 암호 조합 변경보다 앞선다.

올바른 순서

ㄴ → ㄱ → ㄹ → ㄷ

ㄴ. 보안 능력 수립: ClientHello와 ServerHello를 교환하여 프로토콜 버전, 세션 ID, 암호 조합, 압축 방법을 정하고 초기 난수를 교환한다. 이후 인증과 키 교환이 이때 **합의한 방식에 따라** 진행되므로 가장 먼저 수행한다.

ㄱ. 서버 인증 및 키 교환: 서버가 자신의 인증서를 보내고, 선택한 키 교환 방식에 필요한 경우 ServerKeyExchange 메시지를 보낸다. 클라이언트 인증이 필요하면 인증서도 요청한다. 클라이언트는 이 정보를 받아 서버를 확인하고 자신의 응답을 준비하므로 **서버 측 정보 제공이 먼저**이다.

ㄹ. 클라이언트 인증 및 키 교환: 서버의 요청에 따라 클라이언트가 인증서를 보내고, ClientKeyExchange 메시지로 키 설정에 필요한 정보를 보낸다. 인증서 전송은 클라이언트 인증이 요청된 경우에 이루어지며, 키 교환을 통해 **양측이 통신에 사용할 키를 준비**한다.

ㄷ. 암호 조합 변경: ChangeCipherSpec 메시지로 이후 전송할 데이터를 앞서 합의한 암호 알고리즘과 키로 보호하겠다고 알린다. 이는 새 알고리즘을 다시 협상하는 것이 아니라 **준비된 암호 설정을 활성화**하는 것이므로 키 교환 뒤에 이루어진다. 이어 Finished 메시지로 핸드셰이크 결과를 확인한다.

NoteBase

문 17. 인터넷 메일 구조의 핵심요소에 대한 설명으로 옳지 않은 것은?

① MUA – 사용자 액터(actor)와 사용자 응용프로그램을 대신하여 동작한다.

② MSA - 원격서버로부터 POP3 또는 IMAP를 사용하여 메시지를 추출한다.

③ MDA - 메시지를 MHS에서 MS로 메시지를 전달한다.

④ MTA - 메시지가 목적지 MDA에 도달할 때까지 중계 역할을 한다.

정답 2번

간단한 해설

MSA는 발신 MUA가 제출한 메시지를 받아 **메일 전송 시스템에 넘기는** 역할을 한다. POP3나 IMAP로 원격 서버의 메시지를 가져오는 것은 **수신 메시지 접근 기능**이다.

선지별 해설

① O: MUA(메시지 사용자 에이전트)는 **사용자와 사용자 응용프로그램을 대신하여** 메일 서비스를 이용하는 구성 요소이다. 발신 측에서는 메시지를 작성해 제출하고, 수신 측에서는 받은 메시지를 표시하거나 답장하는 등의 처리를 수행한다.

② X: MSA(메일 제출 에이전트)는 **발신 MUA가 제출한 메시지를 접수**하여 메일 처리 시스템(MHS)으로 넘긴다. 일반적으로 SMTP 기반의 제출 프로토콜을 사용한다. POP3나 IMAP로 원격 서버의 메시지를 가져오거나 조회하는 것은 수신 측 MUA 또는 메시지 저장소(MS)의 접근 기능이므로, 이를 MSA의 역할로 설명한 부분이 잘못되었다.

③ O: MDA(메일 배달 에이전트)는 MHS에서 운반한 메시지를 **수신자의 사서함으로 최종 배달**하는 역할을 한다. MS는 메시지를 보관하는 저장소이므로, MHS에서 수신자의 MS로 메시지를 전달한다는 설명은 MDA의 배달 기능에 해당한다.

④ O: MTA(메시지 전송 에이전트)는 메시지의 다음 전달 지점을 정해 **목적지 방향으로 중계**한다. 필요한 경우 여러 MTA를 거치며, 마지막 MTA가 목적지 MDA에 메시지를 넘기면 MDA가 최종 배달을 담당한다.

NoteBase

문 18. 웹을 사용할 때 직면하는 보안위협과 그에 대한 대응수단으로 옳지 않은 것은?

① 보안위협: 사용자 데이터 변조, 대응수단: 암호학적인 체크섬(checksum)

② 보안위협: 네트워크상의 도청, 대응수단: 암호화, 웹 프록시(proxy)

③ 보안위협: 위조 요청으로 시스템 과부하 걸기, 대응수단: 포트 스캔

④ 보안위협: 정당한 사용자로 위장, 대응수단: 암호학적인 기술

정답 3번

간단한 해설

포트 스캔은 대상 시스템의 열린 포트와 서비스 상태를 탐색하는 기법이다. 위조 요청으로 자원을 소모시키는 서비스 거부 공격에는 **공격 트래픽 필터링과 요청량 제한** 등의 대응이 필요하다.

선지별 해설

① O: 사용자 데이터 변조는 데이터가 허가 없이 변경되는 무결성 침해이다. 비밀키를 이용하는 메시지 인증 코드(MAC) 같은 **암호학적 체크섬**을 검증하면 데이터의 변경이나 위조를 탐지할 수 있다. 단순히 해시값만 함께 전송하면 공격자가 데이터와 해시값을 모두 바꿀 수 있으므로, 검증값 자체의 신뢰성도 확보해야 한다.

② O: 네트워크상의 도청은 전송 내용을 몰래 읽는 기밀성 침해이다. 암호화는 전송 내용을 암호문으로 바꾸어 도청자가 내용을 알아내기 어렵게 한다. 웹 프록시는 통신을 중계하며 HTTPS 제공 등 보호 기능을 적용하는 수단으로 활용할 수 있지만, **단순 중계만으로 도청이 방지되는 것은 아니며** 보호할 통신 구간에 암호화가 적용되어야 한다.

③ X: 위조 요청으로 시스템에 과부하를 주는 것은 정상 사용자의 이용을 방해하는 서비스 거부(DoS) 공격으로, 가용성을 침해한다. 포트 스캔은 **열린 포트와 서비스 상태를 탐색**하는 기법이므로 과도한 요청을 차단하거나 처리 부하를 줄이는 직접적인 대응수단이 아니다. 대응수단은 공격 트래픽 필터링, 요청량 제한 등으로 고쳐야 한다.

④ O: 정당한 사용자로 위장하는 공격에는 상대방의 신원을 확인하는 **인증**이 필요하다. 예를 들어 신뢰할 수 있는 인증서의 공개키로 상대방이 새로 제시된 난수에 한 전자서명을 검증하면, 대응하는 개인키를 보유했는지 확인할 수 있다. 이처럼 암호학적 기술을 이용한 인증은 사용자 위장에 대응하는 수단이다.

NoteBase

문 19. 괄호 안에 공통으로 들어갈 용어는?

◯ ( )코드는 Windows 호스트 프로그램의 권한으로 실행된다.
◯ ( )인젝션은 다른 프로세스의 주소 공간 내에서 ( )를(을) 강제로 로드시킴으로써 코드를 실행시키는 기술이다.

① 동적 링크 라이브러리(DLL)
② 보안 식별자(Security Identifier)
③ 일차 도메인 컨트롤러(Primary Domain Controller)
④ NFS(Network File System)

정답 1번

간단한 해설

DLL은 자신을 로드한 **호스트 프로세스의 권한**으로 실행된다. 다른 프로세스에 DLL을 강제로 로드하여 그 안의 코드를 실행시키는 기술이 **DLL 인젝션**이다.

공통 해설

동적 링크 라이브러리(DLL)는 프로그램이 불러와 사용하는 코드와 데이터를 담은 라이브러리이다. DLL의 코드는 별도의 독립 프로세스로 실행되는 것이 아니라, **자신을 로드한 호스트 프로세스의 주소 공간에서 그 프로세스의 권한으로 실행**된다. 이것이 첫 번째 문장의 설명에 해당한다.

DLL 인젝션은 다른 프로세스의 주소 공간에 DLL을 강제로 로드하여 그 DLL에 담긴 코드를 실행시키는 기술이다. 주입된 코드가 대상 프로세스 안에서 실행되므로, 첫 번째 문장의 실행 방식과 두 번째 문장의 인젝션 설명이 연결된다. 두 문장은 모두 프로세스에 로드되어 실행되는 라이브러리를 가리키므로, 세 괄호에 공통으로 들어갈 용어는 동적 링크 라이브러리(DLL)이다.

NoteBase

문 20. 리눅스 계열 운영체제에서 매주 금요일 오후 6시 50분에 /usr/adm/backuplog.sh에 위치한 쉘 스크립트를 실행시키기 위해 crontab에 입력할 내용은?

① * 50 18 * 5 -exec {/usr/adm/backuplog.sh}

② 50 18 * * 5 /usr/adm/backuplog.sh

③ 50 18 * * 5 -exec {/usr/adm/backuplog.sh}

④ 5 * * 18 50 /usr/adm/backuplog.sh

정답 2번

간단한 해설

crontab은 **분·시·일·월·요일** 순서로 실행 시점을 지정한다. 오후 6시 50분은 '50 18', 매주 금요일은 일·월을 제한하지 않는 '* *'와 요일 값 '5'로 나타내며, 그 뒤에 실행할 스크립트 경로를 적는다.

구성 요소별 해설

사용자 crontab의 작업 한 줄은 '분 시 일 월 요일 실행 명령' 형식이다. 앞의 다섯 필드가 실행 시점을 정하고, 그 뒤에는 해당 시점에 실행할 명령을 적는다.

분: 첫 번째 필드에는 **50**을 적는다. 실행 시각의 분을 뜻하며, 허용 범위는 0~59이다. '*'를 적으면 특정 분을 지정하지 않고 매분을 대상으로 한다.

시: 두 번째 필드에는 **18**을 적는다. 시간은 0~23의 24시간제로 지정하므로 오후 6시는 18시이다. 이 자리에 50을 적으면 허용 범위를 벗어난다.

일: 세 번째 필드에는 *****를 적는다. 매월 특정 날짜에만 실행하는 조건이 없으므로 날짜를 제한하지 않는다.

월: 네 번째 필드에도 *****를 적는다. 특정 월에만 실행하는 조건이 없으므로 모든 월을 대상으로 한다. 월의 숫자 범위는 1~12이므로 18은 사용할 수 없다.

요일: 다섯 번째 필드에는 **5**를 적는다. 일요일을 0으로 하여 월요일 1, 화요일 2, 수요일 3, 목요일 4, 금요일 5로 나타낸다. 일·월을 '*'로 두고 요일을 5로 지정하면 매주 금요일에 실행한다. 50은 유효한 요일 값이 아니다.

실행 명령: 시간 필드 다음에 **/usr/adm/backuplog.sh**를 적는다. 스크립트 경로 앞에 '-exec'를 붙이거나 경로를 중괄호로 감싸는 것은 crontab의 실행 지정 문법이 아니다.

이 구성 요소를 연결하면 '50 18 * * 5 /usr/adm/backuplog.sh'가 되며, ②의 입력 내용과 일치한다.

NoteBase