2018-지방교행-정보보호론-A형-해설

대방고시 전산직/계리직, 하이클래스 군무원 곽후근(gobarian@gmail.com)

1. 다음 설명을 모두 만족하는 정보보호의 목표는?

○ 인터넷을 통해 전송되는 데이터 암호화 ○ 데이터베이스와 저장 장치에 저장되는 데이터 암호화 ○ 인가된 사용자들만이 정보를 볼 수 있도록 암호화

① 가용성
② 기밀성
③ 무결성
④ 신뢰성

정답 체크 :
(2) 기밀성 : 인가된 사람, 인가된 프로세스, 인가된 시스템만이 알 필요성에 근거하여 시스템에 접근해야 함을 의미한다. 기밀성을 위해 암호화와 접근 통제 등을 수행한다.

오답 체크 :
(1) 가용성 : 정보의 내용이 불법적으로 생성 또는 변경되거나 삭제되지 않도록 보호되어야 하는 성질의 의미한다. 가용성을 위해 백업 등을 수행한다.
(3) 무결성 : 인가된 사용자가 자원이 필요할 때 지체 없이 원하는 객체 또는 자원에 접근하여 사용할 수 있어야 하는 성질을 의미한다. 무결성을 위해 접근 제어, 인증 등을 수행한다.
(4) 신뢰성 : 의도된 행위에 대한 결과의 일관성을 유지하는 것으로 정보나 정보시스템을 사용함에 있어서 일관되게 오류의 발생 없이 계획된 활동을 수행하여 결과를 얻을 수 있도록 하는 환경을 유지하는 것이다.
Tip! : 정보보호의 3대 목표는 기밀성, 무결성, 가용성이고, 6대 목표는 책임추적성, 인증성, 신뢰성(또는 부인방지)이다.

2. 다음은 신문기사의 일부이다. 빈칸 ㉠에 공통으로 들어갈 용어로 옳은 것은?

㉠ 은(는) 하나의 PC로 제어되는 대규모 온라인 기기 모음이며, 악성 소프트웨어를 이용해 빼앗은 다수의 좀비 컴퓨터로 구성되는 네트워크라고 볼 수 있다. 일반적으로 PC, 공유기, 스마트 폰, 웹캠, 태블릿 등을 악성코드에 감염시켜 사용한다. ㉠ 은(는) 특정 온라인 서버를 표적으로 다운시키거나 대규모 스팸 캠페인을 전달하는 DDoS 공격에 사용할 수 있다. 또한 사용자는 자신의 기기에 있는 악성코드를 인식하지 못하기 때문에 사생활 침해 사기에 개인 정보를 쉽게 도용당할 수 있다.

— 2017년 ○월 ○일자 —

① 웜(worm)
② 봇넷(botnet)
③ 루트킷(rootkit)

④ 랜섬웨어(ransomware)

정답 체크 :

(2) 봇넷 : DDoS에서 악성 코드(봇)에 감염된 좀비 PC로 구성된 네트워크이다.

오답 체크 :

(1) 웜 : 인터넷 또는 네트워크를 통해서 컴퓨터에서 컴퓨터로 전파되는 악성 프로그램이다. 윈도우의 취약점 또는 응용 프로그램의 취약점을 이용하거나 이메일이나 공유 폴더를 통해 전파되며, 최근에는 공유 프로그램(P2P)을 이용하여 전파되기도 한다. 바이러스와 달리 스스로 전파되는 특성이 있다.

(3) 루트킷 : 시스템 침입 후 침입 사실을 숨긴 채 차후의 침입을 위한 백도어, 트로이목마 설치, 원격 접근, 내부 사용 흔적 삭제, 관리자 권한 획득 등 주로 불법적인 해킹에 사용되는 기능들을 제공하는 프로그램의 모음이다.

(4) 랜섬웨어 : 컴퓨터 시스템을 감염시켜 접근을 제한하고 일종의 몸값을 요구하는 악성 소프트웨어의 한 종류이다. 컴퓨터로의 접근이 제한되기 때문에 제한을 없애려면 해당 악성 프로그램을 개발한 자에게 지불을 강요받게 된다.

3. 다음 스푸핑(spoofing) 공격에 대한 설명 (가)~(다)를 바르게 짝지은 것은?

(가) 공격 대상이 잘못된 IP 주소로 웹 접속 유도 (나) 권한 획득을 위하여 다른 사용자의 IP 주소 강탈 (다) MAC 주소를 속여 클라이언트에서 서버로 가는 패킷이나 그 반대 패킷의 흐름을 왜곡

	(가)	(나)	(다)
①	IP 스푸핑	ARP 스푸핑	DNS 스푸핑
②	ARP 스푸핑	IP 스푸핑	DNS 스푸핑
③	ARP 스푸핑	DNS 스푸핑	IP 스푸핑
④	DNS 스푸핑	IP 스푸핑	ARP 스푸핑

정답 체크 :

(4)

(가) DNS 스푸핑 : 공격자가 실제 DNS 서버보다 빨리 공격 대상에게 DNS Response 패킷을 보내, 공격 대상이 잘못된 IP 주소로 웹 접속을 하도록 유도하는 공격이다.

(나) IP 스푸핑 : 트러스트(Trust)로 접속하고 있는 클라이언트에 DoS 공격을 수행해 클라이언트가 사용하는 IP가 네트워크에 출현하지 못하도록 한 뒤, 공격자 자신이 해당 IP로 설정을 변경한 후 서버에 접속하는 형태로 이루어진다.

(다) ARP 스푸핑 : 근거리 통신망(LAN) 하에서 주소 결정 프로토콜(ARP) 메시지를 이용하여 상대방의 데이터 패킷을 중간에서 가로채는 중간자 공격 기법이다. 즉, 공격자가 가짜 MAC 주소를 서버와 클라이언트에게 알려준다.

4. 리눅스 시스템에서 침해사고 분석 시 wtmp 로그파일에서 확인할 수 있는 정보로 <보기>에서 옳은 것만을 모두 고른 것은?

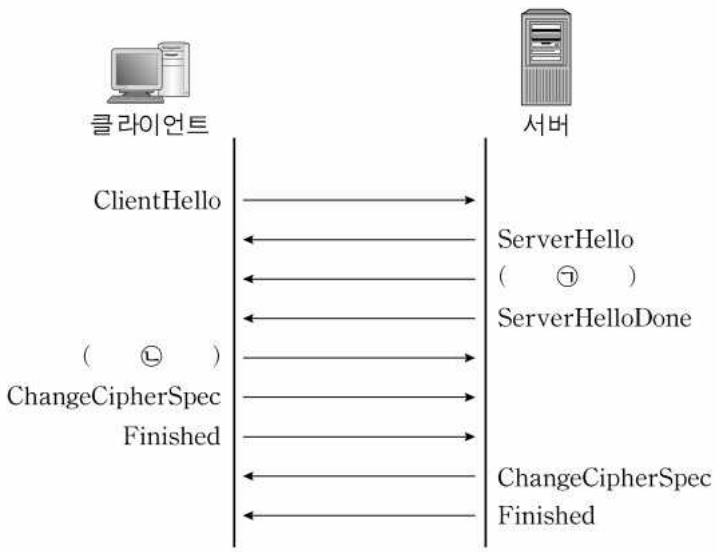
< 보기 > ㄱ. 재부팅 시간 정보

ㄴ. 사용자의 로그인/로그아웃 정보 ㄷ. 로그인에 실패한 사용자의 IP 주소

① ㄱ
② ㄴ
③ ㄱ, ㄴ
④ ㄱ, ㄴ, ㄷ

정답 체크 :
(3)
(ㄱ) 재부팅 시간 정보 : wtmp에서 확인할 수 있다.
(ㄴ) 사용자의 로그인/로그아웃 정보 : wtmp에서 확인할 수 있다.
오답 체크 :
(1) (ㄴ)이 없다.
(2) (ㄱ)이 없다.
(4) (ㄷ) 로그인에 실패한 사용자의 IP 주소 : loginlog에서 확인할 수 있다.

5. 그림은 SSL/TLS에서 상호인증을 요구하지 않는 경우의 핸드쉐이크(handshake) 과정이다. ㉠, ㉡에 들어갈 SSL/TLS 메시지를 바르게 짝지은 것은?



	㉠	㉡
①	ClientRequest	ClientHelloDone
②	ServerKeyExchange	ClientHelloDone
③	ClientKeyExchange	ServerKeyExchange
④	ServerKeyExchange	ClientKeyExchange

정답 체크 :
(4)
SSL/TLS에서 상호인증을 요구한 경우의 핸드쉐이크 과정이다. 문제의 조건은 상호인증을 요구하지 않은 경우이므로 아래의 과정에서 (3), (5), (7), (9)를 제외하면 문제에서 요구한 핸드쉐이크 과정이 된다.

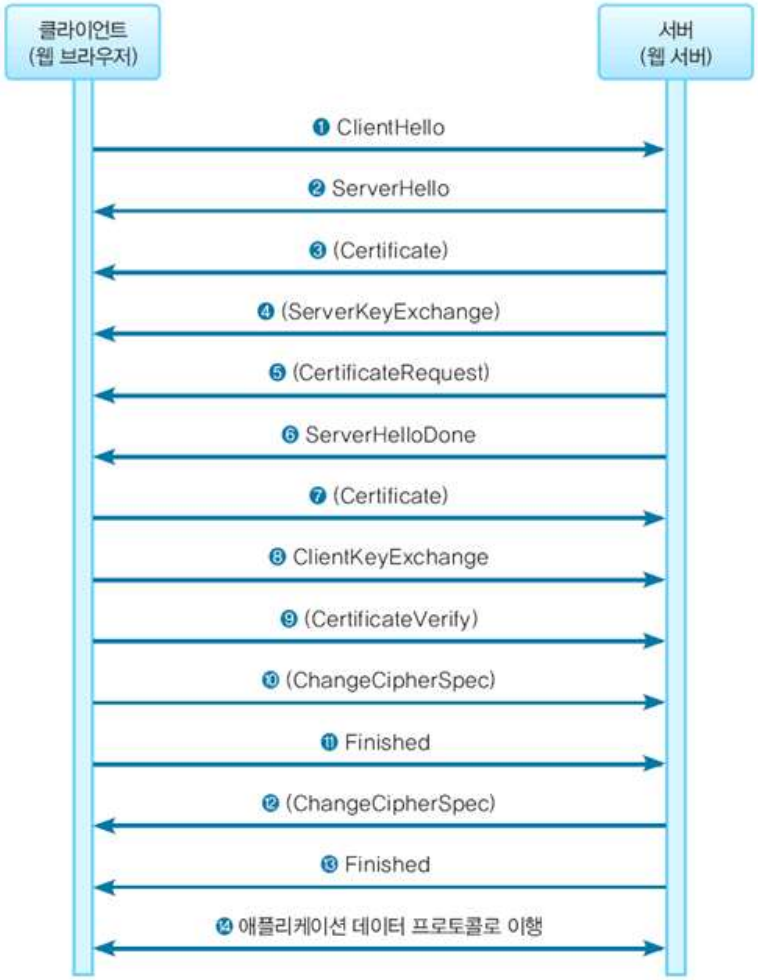


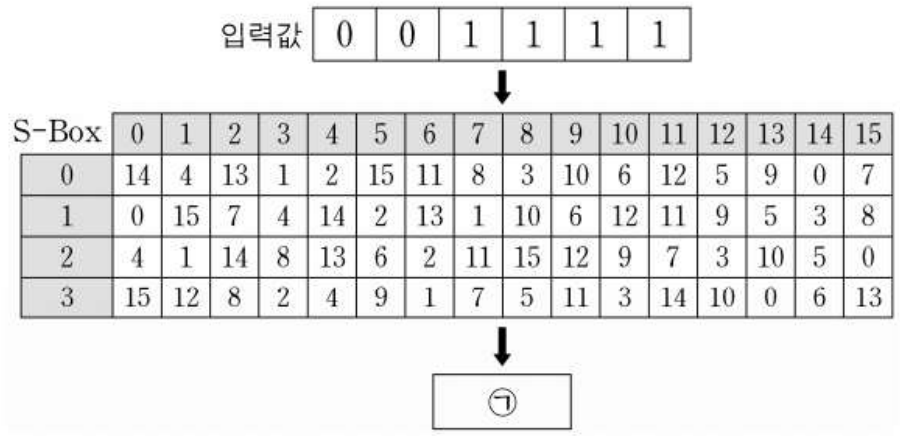
그림 15-7 • TLS 핸드쉐이크 프로토콜

각 과정을 테이블로 정리하면 다음과 같다.

(1) ClientHello	사용하는 버전 번호, 현재 시각, 클라이언트 랜덤값, 세션 ID, 사용하는 암호 스위트 목록, 사용하는 압축 방법 목록을 보낸다.
(2) ServerHello	사용하는 버전 번호, 현재 시각, 서버 랜덤값, 세션 ID, 사용하는 암호 스위트 목록, 사용하는 압축 방법 목록을 보낸다.
(3) Certificate	인증서 목록을 보낸다.
(4) ServerKeyExchange	키 교환을 위한 정보로서 (3)의 Certificate 메시지만으로는 정보가 부족할 때, 클라이언트에게 필요한 정보를 전달한다.
(5) CertificateRequest	서버가 클라이언트에게 인증서를 요구한다.
(6) ServerHelloDone	서버가 보낸 메시지의 끝을 나타낸다.
(7) Certificate	서버의 CertificateRequest 메시지에 대한 응답으로 클라이언트가 서버에게 자신의 인증서를 전송한다.
(8) ClientKeyExchange	(4)의 ServerKeyExchange 메시지에 대응하여 적합한 키 교환 알고리즘을 선정하여 필요한 정보를 전송한다.
(9) CertificateVerify	서버로부터 CertificateRequest를 받은 뒤 클라이언트는 자신의 인증서 속 공개키와 쌍이 되는 정당한 개인 키를 가지고 있

	다는 것을 서버에게 주장하는 것이다.
(10) ChangeChiperSpec	이 메시지를 이용해서 암호를 변경할 수 있다.
(11) Finished	핸드쉐이크 프로토콜 종료를 요청한다.
(12) ChangeChiperSpec	서버가 클라이언트에게 암호를 교환하자고 메시지를 전송한다.
(13) Finished	서버도 클라이언트에게 Finished 메시지를 전송한다.

6. 그림은 DES(Data Encryption Standard)에서 S-Box를 통과하는 과정이다. 입력 값이 001111 (2) 일 때 출력 비트 ㉠은?

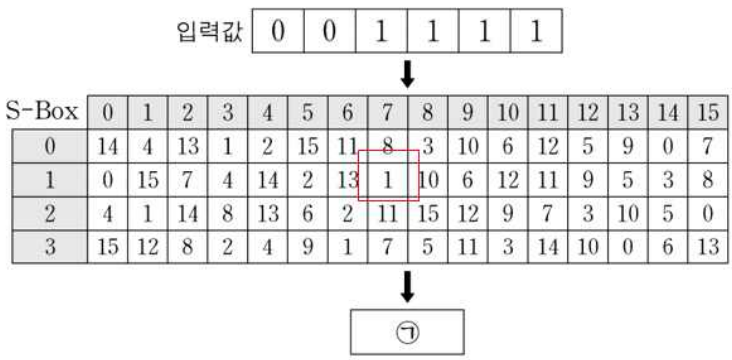


① 0001 (2)
② 0010 (2)
③ 0110 (2)
④ 0111 (2)

정답 체크 :

(1)

S-Box에서는 입력 값의 중간 4비트(0111)로 S-Box의 수평 값(7)을 결정하고, 입력 값의 양쪽 끝 2비트(01)로 S-Box의 수직 값(1)을 결정한다. 결론적으로 아래 그림과 같이 1이 출력된다.



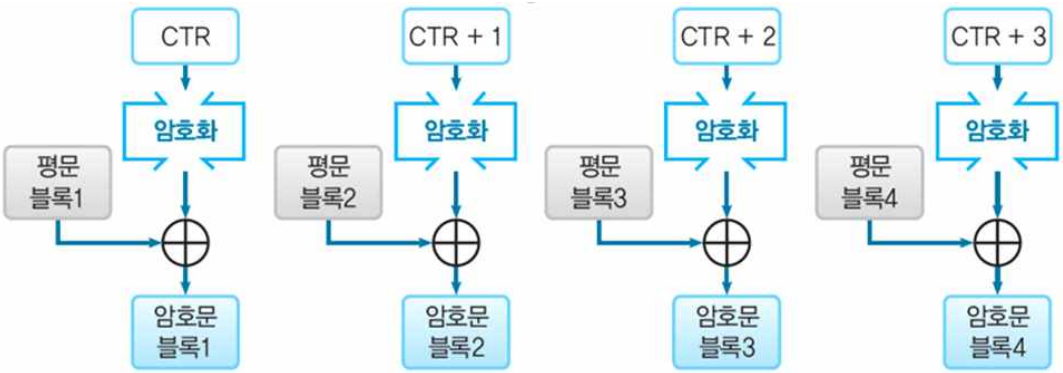
7. 블록암호 운영모드 중 CTR(counter) 모드에 대한 설명으로 <보기>에서 옳은 것만을 모두 고른 것은?

< 보기 >
ㄱ. 운영모드에서 시프트 레지스터를 사용한다.

ㄴ. 패딩이 필요 없으며 평문 블록과 키 스트림을 XOR 연산하여 암호문을 생성한다.
ㄷ. 암호화는 각 블록에 독립적으로 적용되기 때문에, 블록 단위 에러 발생 시 해당 블록에 영향을 준다.

① ㄱ
② ㄱ, ㄴ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ

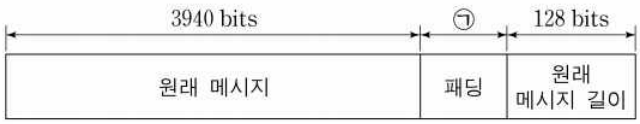
정답 체크 :
(3)
(ㄴ) 아래 그림처럼 평문 블록을 자체를 암호화하는 것이 아니기 때문에 패딩이 필요 없고, 평문 블록과 키 스트림(Counter)을 암호화한 값과 XOR 연산하여 암호문을 생성한다.



(a) CTR 모드에 의한 암호화

(ㄷ) 위의 그림처럼 암호화는 각 블록에 독립적으로 적용되기 때문에, 블록 단위 에러 발생 시 해당 블록에만 영향을 준다. 즉, 에러가 전파되지 않는다.
오답 체크 :
(1), (2), (4)
(ㄱ) CFB, OFB에서 시프트 레지스터를 사용한다.

8. 해시함수 SHA-512를 이용하여 해시값을 구하려고 한다. 원래 메시지가 3940 비트 일때, 그림에서 ㉠ 패딩의 비트 수는?



① 24
② 28
③ 32
④ 36

정답 체크 :
(2)
SHA-512는 블록의 크기(해시 연산을 수행하는 기본 단위)가 1024비트이다. 그러므로 패딩의 결과가 1024로 나누어 떨어져야 한다. 28비트의 패딩을 추가하면 4096(=3940+28+12)비트가

되어 1024비트 4개의 블록으로 나누어진다. 아래 그림은 메시지로부터 해시값이 구해지는 과정을 보여준다. 여기서, IV는 초기화 벡터를 의미하고, H_i는 중간 해시값, H_N은 최종 해시값 512비트를 의미한다. 연산 중에 +는 512비트가 64비트 8개로 분리되어 계산되는 것을 의미하고, 각 64비트는 F의 출력(64비트)과 H_i(64비트)가 더해지므로 64비트를 초과할 가능성을 염두에 두어 mod 2^{64} 나머지 연산을 수행한다.

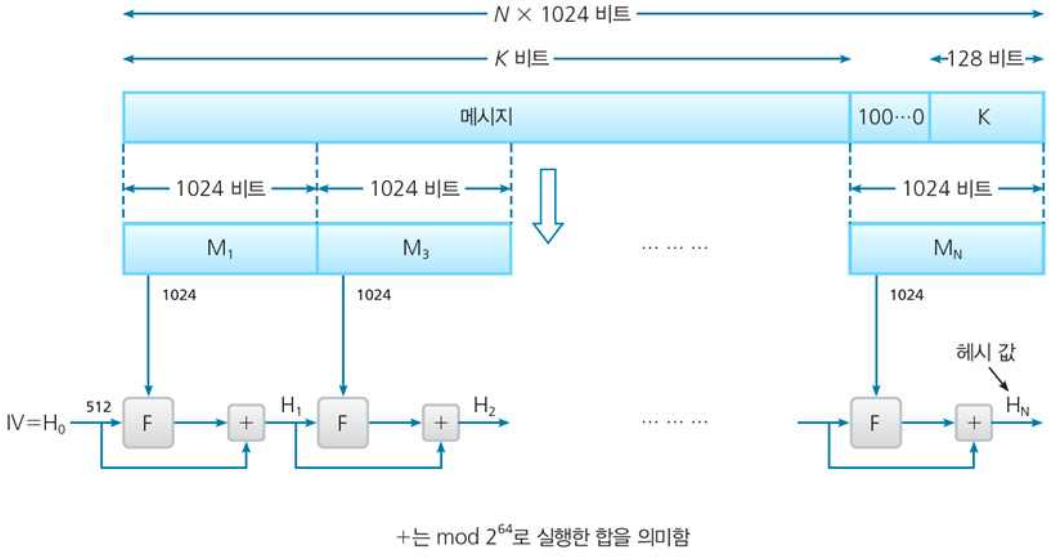


그림 8-10 • SHA-512를 사용한 해시 값 생성

9. 데이터베이스 서버와 어플리케이션 서버로 분리하여 운용할 경우, 데이터베이스 암호화 방식 중 암·복호화가 데이터베이스 서버에서 수행되는 방식으로 <보기>에서 옳은 것만을 모두 고른 것은?

< 보기 >
ㄱ. API 방식
ㄴ. 플러그-인 방식
ㄷ. 필터(filter) 방식
ㄹ. TDE(Transparent Data Encryption) 방식

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄴ, ㄷ, ㄹ

정답 체크 :
(3)
(ㄴ) 플러그-인 방식 : DB 서버에서 수행된다. DB 서버에 설치하는 방식으로써 적용성이 뛰어나지만, 암·복호화 시 DB 서버의 CPU를 사용하기 때문에 부하가 발생한다.
(ㄹ) TDE 방식 : DB 서버에서 수행된다. DBMS에 내장 또는 옵션으로 제공되는 암호화 기능을 이용하는 방식으로써, DBMS 종류 및 버전에 따라 지원이 가능하다.
오답 체크 :
(1), (2), (4)

(ㄱ) API 방식 : 어플리케이션 서버에서 수행된다. 응용 프로그램 서버에 설치하는 방식으로써, 응용 프로그램의 수정을 동반한다.
(ㄷ) 필터 방식 : 어플리케이션 서버에서 수행된다. 독립된 프로세스로 구동하여 어플리케이션과 DBMS 중간에서 암·복호화 처리를 하는 방식이다.
Tip! : 데이터베이스 암호화 방식을 테이블로 정리하면 다음과 같다. 테이블에서 Plug-in, API, Hybrid는 컬럼(column) 암호화 방식이고, TDE, 파일 암호화는 블록(block) 암호화 방식이다.

유형	운영 형태	특징
Plug-in	DB 서버	구축 시 일부 어플리케이션 수정이 필요하며 DB 서버의 성능에 대한 검토 필요
API	DB & 어플리케이션 서버	Plug-in 방식에 비해 DB 서버에 영향을 주지 않으나 구축 시 어플리케이션의 수정 필요
Hybrid (Plug-in+API)	DB & 어플리케이션 서버	Plug-in과 API 방식이 조합된 형식임
TDE 방식	DB 서버	일반적으로 어플리케이션 수정이 필요 없음, DB 등 지원 가능 여부에 대한 고려 필요
파일 암호화	DB & 어플리케이션 서버	일반적으로 어플리케이션 수정이 필요 없음, OS, 스토리지 등 지원 가능 확인 필요

10. 다음의 개인정보보호법 제17조 ①항에 따라 개인정보처리자가 정보주체의 개인정보를 수집한 목적범위 안에서 제3자에게 제공할 수 있는 경우로 <보기>에서 옳은 것만을 모두 고른 것은?

제17조(개인정보의 제공) ① 개인정보처리자는 다음 각 호의 어느 하나에 해당되는 경우에는 정보주체의 개인 정보를 제3자에게 제공(공유를 포함한다. 이하 같다)할 수 있다.

< 보기 > ㄱ. 정보주체와의 계약의 체결 및 이행을 위하여 불가피하게 필요한 경우 ㄴ. 공공기관이 법령 등에서 정하는 소관 업무의 수행을 위하여 불가피한 경우 ㄷ. 법률에 특별한 규정이 있거나 법령상 의무를 준수하기 위하여 불가피한 경우

① ㄱ
② ㄷ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ
정답 체크 :
(3)
(ㄴ), (ㄷ) : "개인정보 보호법" 제17조(개인정보의 제공) 상 개인정보처리자는 다음 각 호의 어느 하나에 해당되는 경우에는 정보주체의 개인정보를 제3자에게 제공(공유를 포함한다. 이하 같다)할 수 있다. 1. 정보주체의 동의를 받은 경우, 2. 제15조제1항제2호(법률에 특별한 규정이 있거나 법령상 의무를 준수하기 위하여 불가피한 경우)·제3호(공공기관이 법령 등에서 정하는 소관 업무의 수행을 위하여 불가피한 경우) 및 제5호(정보주체 또는 그 법정대리인이 의

사표시를 할 수 없는 상태에 있거나 주소불명 등으로 사전 동의를 받을 수 없는 경우로서 명백히 정보주체 또는 제3자의 급박한 생명, 신체, 재산의 이익을 위하여 필요하다고 인정되는 경우)에 따라 개인정보를 수집한 목적 범위에서 개인정보를 제공하는 경우

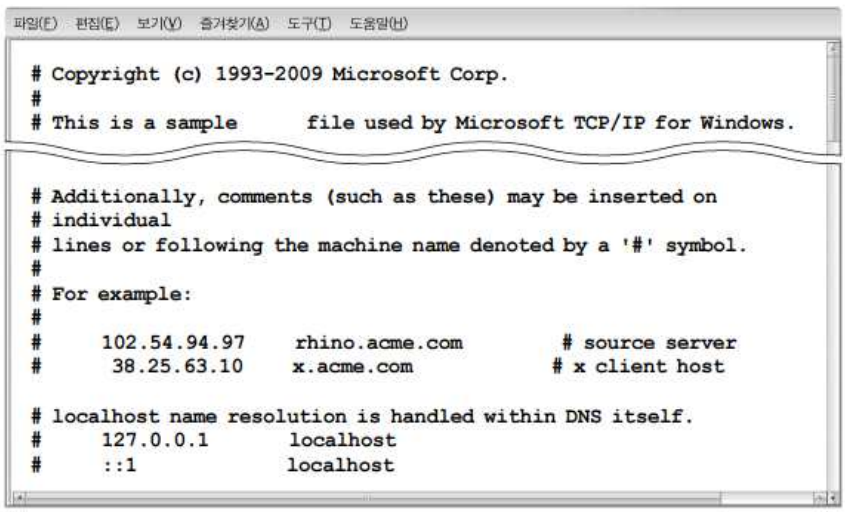
오답 체크 :

(1), (4)

(ㄱ) : "개인정보 보호법" 제15조(개인정보의 수집·이용)에 따르면, 개인정보처리자는 다음 각 호의 어느 하나에 해당하는 경우에는 개인정보를 수집할 수 있으며 그 수집 목적의 범위에서 이용할 수 있다. 1. 정보주체의 동의를 받은 경우, 2. 법률에 특별한 규정이 있거나 법령상 의무를 준수하기 위하여 불가피한 경우, 3. 공공기관이 법령 등에서 정하는 소관 업무의 수행을 위하여 불가피한 경우, 4. 정보주체와의 계약의 체결 및 이행을 위하여 불가피하게 필요한 경우, 5. 정보주체 또는 그 법정대리인이 의사표시를 할 수 없는 상태에 있거나 주소불명 등으로 사전 동의를 받을 수 없는 경우로서 명백히 정보주체 또는 제3자의 급박한 생명, 신체, 재산의 이익을 위하여 필요하다고 인정되는 경우, 6. 개인정보처리자의 정당한 이익을 달성하기 위하여 필요한 경우로서 명백하게 정보주체의 권리보다 우선하는 경우. 이 경우 개인정보처리자의 정당한 이익과 상당한 관련이 있고 합리적인 범위를 초과하지 아니하는 경우에 한한다.

(2) (ㄴ)이 없다.

11. 그림은 DNS 보다 우선 적용되는 파일로, 해커는 이 파일을 변조하여 파밍(pharming)에 사용할 수 있다. 이 파일명으로 옳은 것은?



① hosts

② networks

③ protocol

④ services

정답 체크 :

(1) hosts : 아래 그림과 같이 IP와 도메인 이름을 가지고 있다.

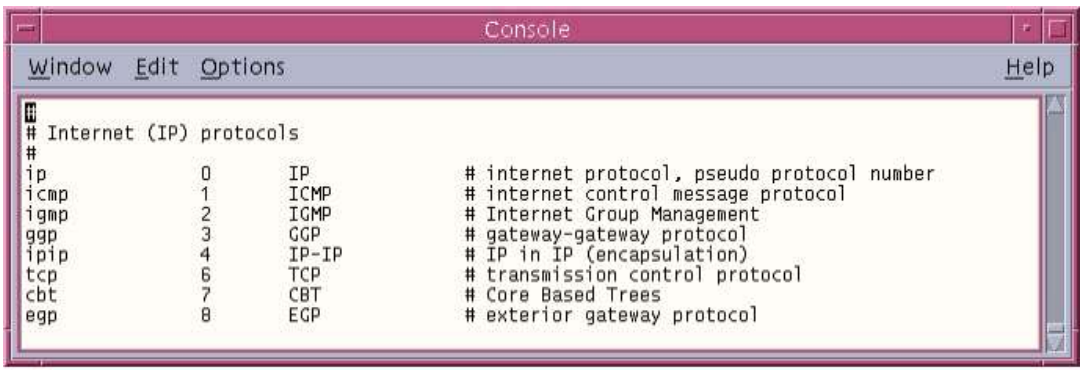
```
# space.
#
# Additionally, comments (such as these) may be inser
# lines or following the machine name denoted by a '#
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # sou
#       38.25.63.10     x.acme.com              # x c

# localhost name resolution is handled within DNS its
#   127.0.0.1       localhost
#   ::1             localhost
127.0.0.1 activation.cloud.techsmith.com
127.0.0.1 www.paul.com
```

오답 체크 :

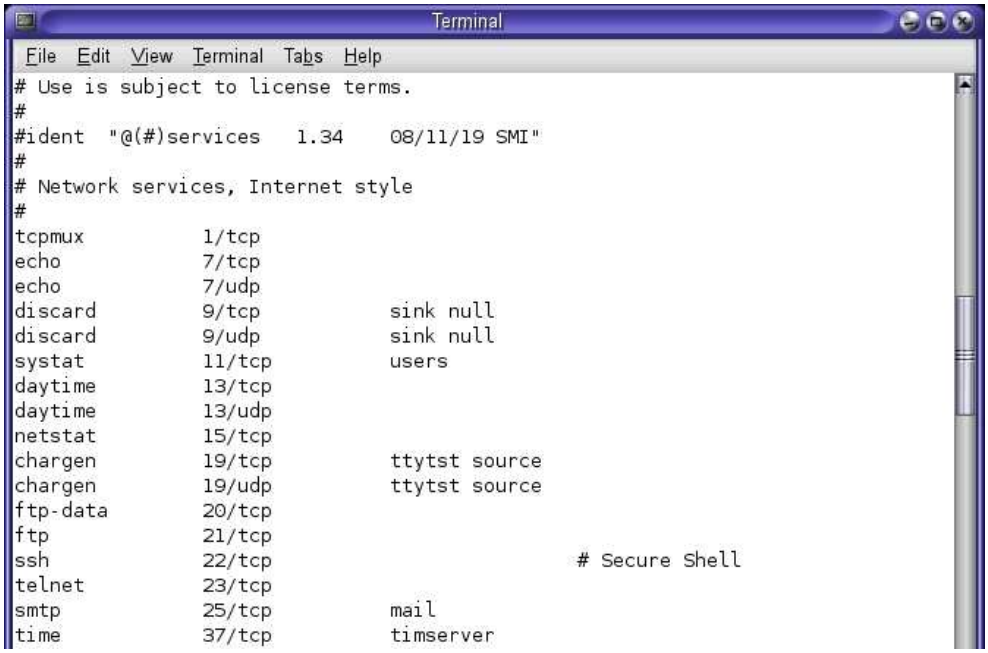
(2) networks : 아래 그림과 같이 네트워크 대역에 대한 이름(심볼)을 가지고 있다.

```
root@fw-test:~# cat /etc/networks
default         0.0.0.0
loopback        127.0.0.0
link-local      169.254.0.0
google-dns      8.8.4.4
root@fw-test:~#
```

(3) protocol : 아래 그림과 같이 프로토콜 종류와 번호를 가지고 있다.



(4) services : 아래 그림과 같이 데몬 이름, 포트 번호, 프로토콜을 가지고 있다.



12. 다음 설명을 모두 만족하는 공개키 기반구조(PKI)의 구성요소는?

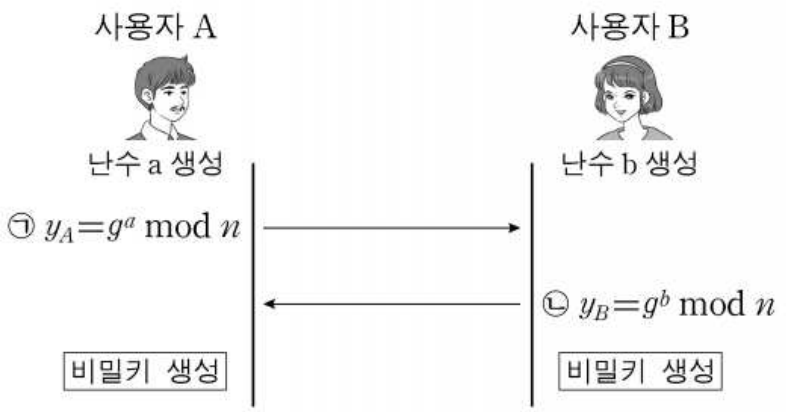
◦ LDAP을 이용하여 X.500 디렉터리 서비스 제공 ◦ 인증서와 사용자 관련 정보, 상호 인증서 쌍, CRL 등을 저장하고 검색하는 데이터베이스

① 사용자(user)
② 저장소(repository)
③ 등록기관(registration authority)
④ 인증기관(certification authority)

정답 체크 :
(2) 저장소 : 인증서를 보존한다. PKI 이용자가 인증서를 입수할 수 있도록 한 데이터베이스이다. LDAP(디렉터리 서비스를 조회하고 수정하는 프로토콜)을 이용하여 X.500(전자 디렉터리 서비스를 전달하는 일련의 컴퓨터 네트워크 표준) 디렉터리 서비스를 제공한다. 여기서 디렉터리 서비스란 컴퓨터 네트워크의 사용자와 네트워크 자원에 대한 정보를 저장하고 조직하는 응용 소프트웨어(응용 프로그램들의 모임)를 의미한다.

오답 체크 :
(1) 사용자 : PKI를 사용해서 자신의 공개키를 등록하고 싶어 하는 사람과 등록되어 있는 공개키를 사용하고 싶어 하는 사람을 의미한다.
(3) 등록기관 : 인증기관(CA)의 일 중 「공개키의 등록과 본인에 대한 인증」을 대행하는 기관이다.
(4) 인증기관 : 인증기관은 인증서의 관리를 행하는 기관으로, 키 쌍을 작성한다(이용자가 작성하는 경우도 있다). 그리고 공개키 등록 때 본인을 인증하고, 인증서를 작성해서 발행하거나 인증서를 폐지한다.

[13 ~ 14] 그림은 Diffie-Hellman의 키 교환 방법이다. 다음 그림을 보고 물음에 답하시오.



13. 위 그림의 식 ㉠, ㉡에서 n이 7일 때, g로 사용할 수 있는 것은?

① 2

② 3

③ 4

④ 7

정답 체크 :

(3) g는 n의 prime root modulo(원시근)이다. 그러므로 3이 원시근이라면 3의 (1부터 6)승을 n으로 모듈로 한 결과가 0과 7을 제외한 모든 숫자가 나와야 한다. 3의 (1부터 6)승을 n으로 차례대로 모듈로 해보면 (3, 2, 6, 4, 5, 1)이라는 숫자가 나오므로 원시근의 조건을 만족한다. 예를 들어, 3^1 mod 7 = 1, 3^2 = mod 7 = 2 등과 같이 계산하면 된다.

오답 체크 :

(1) 2는 (1부터 6)승 모듈로(n) 결과가 0과 7을 제외한 모든 숫자가 나오지 않으므로 원시근이 될 수 없다.

(3) 4는 (1부터 6)승 모듈로(n) 결과가 0과 7을 제외한 모든 숫자가 나오지 않으므로 원시근이 될 수 없다.

(4) g는 원시근의 조건에 의해 n보다 작은 수(0부터 6)어야 합니다. 그러므로 7은 답에서 제외한다.

14. 위 그림에서 사용자 A, B가 생성하는 비밀키 값과 동일한 값을 구하는 식은? (단, mod는 나머지를 구하는 연산자이고, ϕ(n)는 오일러의 Totient 함수이다.)

① g^{axb} mod n

② g^{a+b} mod n

③ g^{axb} mod ϕ(n)

④ g^{a+b} mod ϕ(n)

정답 체크 :

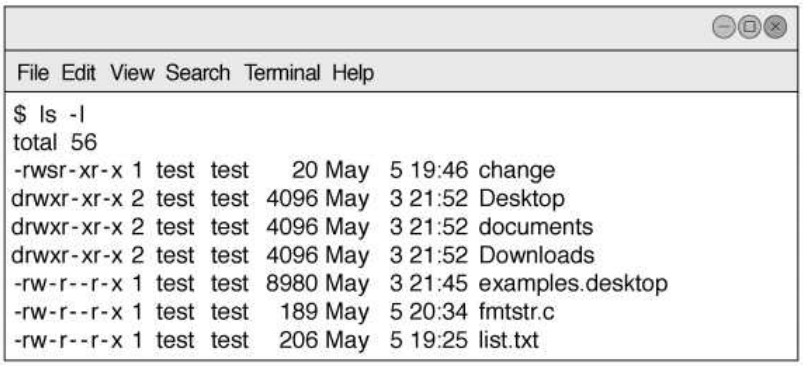
(1)

사용자 A : 사용자 B로부터 수신한 $y_B = g^b$ mod n에 자신의 난수 a를 합쳐 비밀키 = y_B^a = $(g^b \bmod n)^a$ = g^{ab} mod n을 계산한다.

사용자 B : 사용자 A로부터 수신한 $y_A = g^a$ mod n에 자신의 난수 b를 합쳐 비밀키 = y_A^b = $(g^a \bmod n)^b$ = g^{ba} mod n을 계산한다.

결론적으로, 사용자 A, B는 동일한 비밀키 g^{ab} mod n을 가지게 된다.

15. 그림은 리눅스에서 ls - l 명령을 실행한 결과이다. change 파일에 대한 설명으로 옳은 것은?



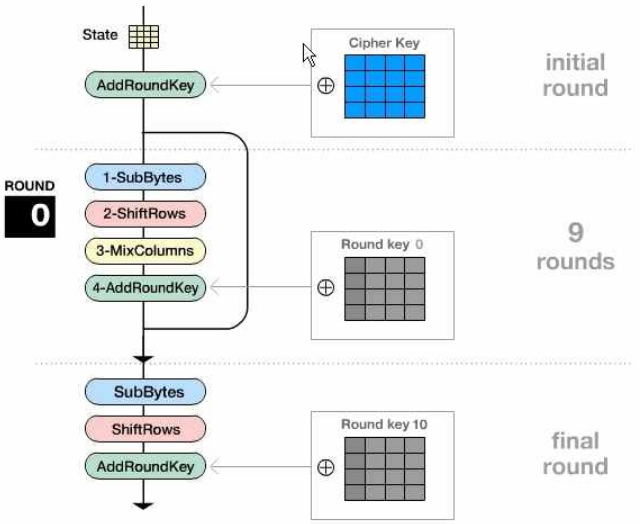
① change 파일은 setGID 비트가 설정되어 있다.
② change 파일의 접근 권한을 8진수로 표현하면 754이다.
③ test 외의 사용자는 change 파일에 대해 쓰기 권한을 가진다.
④ change 파일은 test 외의 사용자가 실행할 때 유효 사용자 ID(effective UID)는 test가 된다.
정답 체크 :
(4) change 파일은 setUID 비트가 설정되어 있으므로 test 외의 사용자가 실행할 때 유효 사용자 ID(effective UID)는 test가 된다.
오답 체크 :
(1) change 파일은 setUID 비트가 설정되어 있다.
(2) change 파일의 접근 권한을 8진수로 표현하면 4755이다.
(3) test 외의 사용자는 change 파일에 대해 읽기와 실행 권한을 가진다.

16. AES(Advanced Encryption Standard) 알고리즘에서 사용되는 함수들이다. 암호화 과정의 마지막 라운드 에서 수행되는 함수를 <보기>에서 옳은 것만을 모두 골라, 호출 순서대로 바르게 나열한 것은?

```
< 보기 >
ㄱ. SubBytes( )          /* 바이트 치환 */
ㄴ. ShiftRows( )         /* 행 이동 */
ㄷ. MixColumns( )        /* 열 혼합 */
ㄹ. AddRoundKey( )       /* 라운드 키 더하기 */
```

① ㄱ-ㄷ
② ㄱ-ㄴ-ㄹ
③ ㄴ-ㄱ-ㄹ
④ ㄹ-ㄱ-ㄴ-ㄷ
정답 체크 :
(2)
AES의 10라운드 암호화 과정을 그림으로 나타내면 다음과 같다. 그림에서 Round 0과

Round Key 0은 Round 1과 Round Key 1로 해석해도 무방하다.



여기서, SubByte는 바이트 대체를 의미하며, ShiftRows는 행 이동을 의미한다. 그리고 MixColumns는 열 섞기를 의미하고, AddRoundKey는 라운드 키와 XOR를 의미한다.

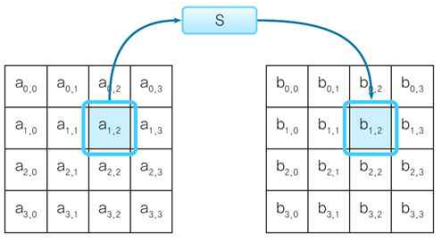


그림 4-11 • SubBytes(바이트 대체)

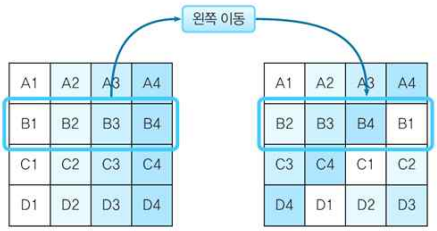


그림 4-12 • ShiftRows(행 이동)

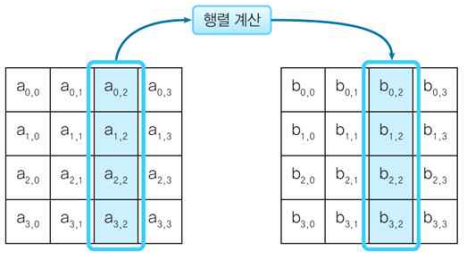


그림 4-13 • MixColumns(열 섞기)

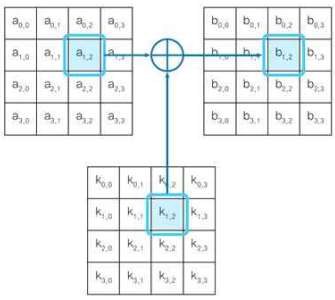


그림 4-14 • AddRoundKey(라운드 키와 XOR)

17. 네트워크 기반 침입탐지시스템(Intrusion Detection System)의 특징에 대한 설명으로 <보기>에서 옳은 것만을 모두 고른 것은?

< 보기 >
ㄱ. 어플리케이션 서버에 설치되어 관리가 간단하다.
ㄴ. 네트워크상의 패킷을 분석하여 침입을 탐지한다.
ㄷ. 방화벽 내부의 내부 네트워크와 방화벽 외부의 DMZ에 모두 배치 가능하다.

① ㄱ

② ㄴ

③ ㄱ, ㄷ

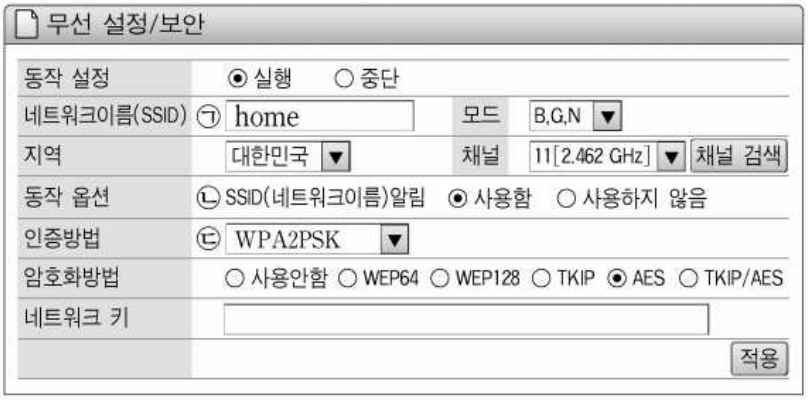
④ ㄴ, ㄷ
정답 체크 :
(4)
(ㄴ) IDS는 passive(수동적) 보안 장비로, 네트워크상의 패킷을 7계층(payload or content) 레벨에서 분석하여 침입을 탐지한다.
(ㄷ) IDS는 호스트 IDS와 네트워크 IDS가 존재한다. 호스트 IDS는 보호 대상이 되는 PC에 설치하며, 네트워크 IDS는 패킷의 전송 경로가 되는 네트워크의 어디에든(방화벽 내부, 방화벽 외부 등) 설치가 가능하다.
오답 체크 :
(1), (3)
(ㄱ) 해당 설명은 소프트웨어 방화벽에 대한 설명이고, 소프트웨어 IDS(Snort)를 어플리케이션 서버에 설치할 수 있다고 하더라도 관리가 간단하지는 않다(웜과 바이러스를 차단하기 위해 시그너처를 관리하고, 관련 규칙들을 계속해서 갱신해주어야 한다).
(2) (ㄷ)이 없다.

18. 다음 설명을 모두 만족하는 OTP(One-Time Password) 생성 방식은?

○ 해시체인 방식으로 계산된다. ○ 생성된 일회용 패스워드의 사용 횟수가 제한된다. ○ 검증 시 계산량이 적기 때문에 스마트카드와 같은 응용에 적합하다.

① S/KEY 방식
② 시간 동기화 방식
③ 이벤트 동기화 방식
④ Challenge-Response 방식
정답 체크 :
(1) S/KEY : 클라이언트에서 정한 임의의 비밀키를 서버로 전송한다. 클라이언트로부터 받은 비밀키를 첫 값으로 사용하여, 해시 체인 방식으로, 이전 결과 값에 대한 해시 값을 구하는 작업을 n번 반복한다. 그렇게 생성된 n개의 OTP를 서버에 저장한다. 클라이언트에서 정한 OTP에 해시 함수를 n-i번 중첩 적용하여 서버로 전송한다. 서버에서는 클라이언트로부터 받은 값에 해시 함수를 한 번 적용하여, 그 결과가 서버에 저장된 n-i+1번째 OTP와 일치하는지 검사한다. 일치하면 인증에 성공한 것으로, 카운트를 1 증가시킨다.
오답 체크 :
(2) 시간 동기화 : 클라이언트는 현재 시각을 입력값으로 OTP를 생성해 서버로 전송하고, 서버 역시 같은 방식으로 OTP를 생성하여 클라이언트가 전송한 값의 유효성을 검사한다. 하지만 클라이언트와 서버의 시간 동기화가 정확하지 않으면 인증에 실패하게 된다는 단점이 있으며, 이를 보완하기 위해 일반적으로 1~2 분 정도를 OTP 생성 간격으로 둔다.
(3) 이벤트 동기화 : 서버와 클라이언트가 카운트 값을 동일하게 증가시켜 가며, 해당 카운트 값을 입력값으로 OTP를 생성해 인증하는 방식이다.
(4) Challenge-Response : 서버에서 난수 생성 등을 통해 임의의 수를 생성하고 클라이언트에 그 값을 전송하면, 클라이언트가 그 값으로 OTP를 생성해 응답한 값으로 인증하는 방식이다.

19. 그림은 무선 AP(Access Point)를 설정한 결과 화면의 일부이다. ㉠~㉢에 대한 설명으로 옳지 않은 것은?



① ㉠의 'home'은 관리자가 변경할 수 없다.
② ㉡을 '사용함'으로 설정하였기 때문에, 클라이언트의 무선 네트워크 연결 목록에서 'home'을 볼 수 있다.
③ 무선 네트워크 연결 목록에서 'home'을 볼 수 없게 하여 접속시도를 줄이려면, ㉡을 '사용하지 않음'으로 설정을 변경한다.
④ ㉢ 을 'WPA2PSK'로 설정하였기 때문에, '암호화 방법' 으로 AES를 사용할 수 있다.
정답 체크 :
(1) SSID는 다른 AP와 겹치지 않는다면 관리자가 언제든지 변경이 가능하다.
오답 체크 :
(2) SSID 알림을 사용하면 클라이언트가 SSID를 볼 수 있다.
(3) SSID 알림을 사용하지 않으면 클라이언트가 SSID를 볼 수 없다. 클라이언트가 직접 타이핑해야 접속이 가능하다.
(4) WPA2PSK는 암호화 방식으로 AES(강력한 대칭키)를 사용한다.

20. 그림은 C 언어 소스코드의 일부이다. 이 소스코드 ㉠~㉣에서 오버플로우 취약점을 가진 행은?

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BUFSIZE 10
int main(int argc, char **argv)
{
   char *dest = NULL; ---------- ㉠
   dest = (char *)malloc(BUFSIZE); ------ ㉡
   strcpy(dest, argv[1]); ---------- ㉢
   free(dest);
   return 0; ---------- ㉣
}
```

① ㉠
② ㉡

③ ㉢
④ ㉣
정답 체크 :
(3)
(ㄷ) strcpy(dest, argv[1]); // dest에 10바이트가 할당되어 있는데 입력으로 들어오는 argv[1]의 크기가 정해져 있지 않다. 만약, 공격자가 10바이트 이상을 입력하게 되면 dest의 버퍼를 초과하게 되어(strcpy는 argv[1]을 dest로 복사함) 스택에 저장된 복귀 주소를 공격자가 원하는 복귀 주소로 수정할 수 있다.
Tip! : 시큐어 코딩(secure coding)이란 안전한 소프트웨어 개발을 위해 소스 코드 등에 존재할 수 있는 잠재적인 보안 취약점을 제거하는 것이다. 즉, strcpy와 같은 함수를 사용하지 않는다.