해킹 입무자를 위한 TCP/IP 이론과 보안

노트북: 네트워크
만든 날짜: 2020-04-07 오전 12:34 **수정한 날짜:** 2020-04-29 오전 12:21
작성자: yoonjeong_choi@tmax.co.kr

[IP주소, 서브넷 마스크, 기본 게이트웨이의 의미]

- 이더넷 어댑터 : PC에서 사용하는 LAN 카드
  - LAN 카드
    - NIC(Network Interface Controller)
    - LAN 카드가 다수 인 PC에서는 cmd창의 "ipconfig" 명령어에 대해 "이더넷 어댑터 로컬 영역 연결2"와 같은 내용이 추가로 생긴다
  - 이더넷
    - LAN 영역에서 사용하는 통신 기술 중 하나
    - 현재 LAN 영역에서의 사실상 표준 방식
  - 이더넷 방식의 LAN 영역에서 사용하는 NIC 장치

- IP 주소(IPv4)
  - 인터넷 공간에서 호스트가 사용하는 고유한 논리 식별자
  - 32비트로 이루어진 체계
    - 0.0.0.0 ~ 255.255.255.255 (255 = pow(2.8-1))
    - Note : IPv6은 128비트 체계
  - 인터넷 공간에서 자기 PC를 유일하게 구별하기 위한 식별자
  - 출발지 IP 주소에 해당
    - (why?) 자기 PC에서 사용하는 IP 주소는 데이터를 송신하는 주체
    - 즉, 자기 PC에서 사용하는 IP 주소는 출발지 IP 주소라고 간주
  - 클래스 : 32비트 체계에서 첫번째 자리 수로 구분
    - A 클래스 : 1~126
      - ex) 8.8.8.8
    - B 클래스 : 128 ~ 191
      - ex) 168.126.63.1
    - C 클래스 : 192~223
  - 특수 IP 127.0.0.01
    - 어떤 클래스에도 속하지 않는 주소
    - 자기가 사용하는 LAN 카드 자신을 의미
    - 루프백 주소(Loopback address)

- 서브넷 마스크
  - IP 주소와 쌍으로 사용하는 개념
  - 서브넷 마스크 표기법 : IP 주소를 서브넷 마스크를 이용하여 표기하는 방식
  - IP 주소를 네트워크 ID와 호스트 ID를 분리하는 개념
  - 서브넷 마스크 예시
    - 10.10.10.10 255.0.0.0
      - 네트워크 ID : 10
      - 호스트 ID : 10.10.10
    - 172.16.10.10 255.255.0.0
      - 네트워크 ID : 172.16
      - 호스트 ID : 10.10
    - 192.168.10.10 255.255.255.0
      - 네트워크 ID : 198.168.10
      - 호스트 ID : 10
  - IP 주소와 서브넷 마스크의 대응 관계는 IP 주소의 체계를 의미
    - IP 주소 체계는 전화번호 체계에서 유래
    - 네트워크 ID는 국번, 호스트 ID는 국번에서 사용하는 일련 번호에 해당하는 개념
    - 전화번호에서 국번과 번호를 대시(-)로 구분하는 것처럼 IP 주소에서는 서브넷 마스크를 통해 구분

- 게이트웨이(라우터)
  - 게이트웨이와 라우터는 같은 의미
    - 게이트웨이 : 소프트웨어 측면을 강조할 때 사용
    - 라우터 : 하드웨어 측면을 강조할 때 사용
  - 인터넷으로 접속하기 위한 일종의 관문 역할을 수행하는 장비
  - 서로 다른 LAN 영역의 호스트 사이를 연결해주는 기능(라우팅)을 수행
  - 예시 : 무선 공유기

- LAN(Local Area Network) 영역
  - 네트워크 ID를 공유하는 장치들의 집합체를 이루는 공간
  - 동일한 네트워크 ID를 공유하는 장치들의 집합적 공간
  - i.e 동일한 게이트웨이 주소를 사용하는 장치들의 집합적 공간

- 네트워크 ID
  - (NOTE) PC와 게이트웨이에 설정한 네트워크 ID는 동일
  - 전화번호 체계 예시
    - 동일한 국번은 동일한 음성 교환기를 의미
    - 서로 다른 국번은 서로 다른 음성 교환기를 의미

        - 같은 네트워크 ID를 사용하는 호스트는 서로 동일한 LAN 영역에 있음을 의미
        - 다른 네트워크 ID를 사용하는 호스트는 서로 다른 LAN 영역에서 있음을 의미
    - 무수한 LAN 영역에서 자기 LAN 영역을 구분하기 위한 식별자
    - i.e 해당 LAN 영역을 식별하는 고유 IP 주소 대역

- 호스트 ID
    - 동일한 LAN 영역에 속한 호스트 각각을 구분하기 위한 식별자

- 라우팅
    - 라우터가 수행하는 기능
    - 서로 다른 네트워크 ID를 사용하는 LAN 영역 사이를 연결해 주는 기능
    - 전화번호 체계 예시
        - 국번이 다른 경우 외부 음성 교환기로 회선을 연결해주는 동작
    - i.e 출발지와 목적지가 서로 다른 LAN 영역인 경우 출발지 호스트와 목적지 호스트를 연결해주는 기능
    - 같은 네트워크 ID를 사용하는 i.e 동일 LAN 영역에서의 출발지 와 목적지를 연결하는 기능은 "스위칭"

[DHCP 서비스, DNS 서비스, 물리적 주소의 의미]

- DHCP(Dynamic Host Configuration Protocol)
    - PC 사용자에게 IP 주소, 서브넷 마스크, 게이트웨이 IP 주소 등을 자동으로 할당해주는 기능을 제공하는 서버
        - 사용할 IP 주소를 서버에 미리 등록
        - IP 주소를 유동 IP 방식으로 사용
    - 배경
        - IP 주소는 사용자가 IP 주소, 서브넷 마스크, 게이트웨이 IP 주소 등을 직접 입력해야 한다
        - i.e 사용자가 IP 주소의 기본 체계를 알고 있어야 한다
        - 대규모의 LAN 영역에서 장치의 IP 주소 등을 일일이 입력하는 것은 번거롭다
    - 전화 번호 체계 예시
        - 음성 교환기에서 전화번호를 자동으로 할당하는 방식
        - 즉, 음성 교환기는 라우팅과 DHCP 기능을 모두 수행
    - 예시 : 무선 공유기 (라우팅, DHCP 서버 기능 모두 수행)

- DNS(Domain Name Server) 서버
  - 도메인 네임과 IP 주소의 대응 관계를 일종의 데이터베이스 형태로 저장하여 사용하는 기능 제공
    - 도메인 네임 : 웹 서버에 접속하기 위해 웹 브라우저의 주소창(URL)에 입력하는 주소
  - 전화번호 체계 예시
    - 전화번호부에서 성명과 전화번호에 대응
    - 성명 <-> 도메인 네임
    - 전화번호 <-> IP 주소

- MAC(Media Access Control) 주소
  - 물리적 주소 의미
  - 이더넷 어댑터 로컬 영역 연결 부분과 관련
    - i.e 동일 LAN 영역에서의 연결 부분과 관련
  - 특정 LAN 영역에서 다수의 호스트가 허브/스위치 같은 집선 장치에 물린 상태
    - 집선 장치 : 선을 연결해주는 장비
    - 내부 통신 : LAN 영역에서 스위치같은 집선 장치에 물린 호스트 사이에서 일어나는 통신
    - 내부 통신은 LAN 영역 내에서만 발생 i.e 라우팅과 무관
  - 특정 LAN 영역에서 호스트들 간 구분할 수 있는 주소
    - i.e LAN 영역에서 내부 통신을 수행하기 위해 필요한 주소
  - 스위칭(Switching)
    - 모종의 테이블에 맥 주소를 저장하고 검색하는 과정
    - 스위치 장비가 맥 주소에 기반하여 호스트 사이에 내부 통신을 구현해주는 기능
    - 포워딩 : 맥 주소 인식을 통해 오직 목적지 맥 주소가 있는 해당 포트로만 데이터를 전송하는 것
  - LAN 카드(NIC)에 새겨진 주소(=> 물리적 주소)
    - i.e LAN 카드 한 장 구입 <-> 1개의 맥 주소 구입
    - 이더넷 기술이 LAN 영역의 사실상 표준
      - 맥 주소는 사실상 이더넷 방식
      - 이더넷 통신을 위해 이더넷 방식에 따른 맥 주소가 필요
  - 맥 주소는 16진수 표기
    - 예시 : 00-24-1D-DF-8C-47
      - OUI : 00-24-1D
      - 일련번호 : DF-8C-47
    - 맥 주소는 UPI와 일련번호로 구분
      - IP 주소를 네트워크 ID와 호스트 ID로 구분하는 것과 비슷한 방식
      - (BUT!) OUI와 일련번호는 24비트 단위로 고정적
      - i.e OUI - 일련번호 형식
    - OUI(Organiztionally Unique Identifier)

        - 맥 주소를 생성하는 기업의 고유한 식별자
        - 네트워크 ID가 해당 LAN의 고유한 식별자인 것과 비슷한 개념
        - IEEE에서 관리 => 맥 주소 중복을 방지

- 메트로 이더넷(Metro Ethernet)
    - 두 개의 서로 다른 LAN 영역을 한 개의 동일한 LAN 영역으로 통합하는 방식
    - 현업에서 라우터가 사라짐

[ARP 캐시 테이블의 이해]

- ARP(Address Resolution Protocol)
    - IP 주소와 맥 주소 사이를 연결해주는 기능

- ARP 캐시 테이블
    - IP 주소와 맥 주소의 대응 관계를 저장한 테이블
    - 도메인 네임과 IP 주소의 대응 관계를 관리하는 DNS 서비스 기능과 유사

- ARP 요청과 응답 과정
    - 운영체제는 출발지 서브넷 마스크를 기준으로 출발지와 목적지의 네트워크 ID를 비교해 목적지가 스위칭 통신의 대상(동일 LAN)인지 / 라우팅 통신의 대상(다른 LAN)인지 판단
    - 운영체제는 출발지의 ARP 캐시 테이블에 접근해 목적지 맥 주소를 검색
        - 목적지가 스위칭 통신의 대상이면 실제 목적지 IP 주소에 해당하는 맥 주소를 테이블에서 검색
        - 목적지가 라우팅 통신의 대상이면 라우터 IP 주소에 해당하는 맥주소를 테이블에서 검색
    - Case 1 : ARP 캐시 테이블에 목적지 맥 주소가 있는 경우
        - 운영체제는 곧바로 해당 맥주소를 참조해 목적지까지 유니캐스트 방식으로 전송
    - Case 2 : ARP 캐시 테이블에 목적지 맥 주소가 없는 경우
        - 자신이 속한 LAN 영역 전체를 대상으로 ARP 브로드캐스트 질의 전송
        - 동일 LAN 영역에 속한 목적지 호스트는 ARP 유니캐스트 응답을 출발지 호스트로 전송
        - 운영체제는 ARP 유니캐스트 응답으로 획득한 목적지 맥 주소를 ARP 캐시 테이블에 반영

        - 운영체제는 사용자가 전송하고자 하는 실제 데이터를 테이블에 기반해 목적지 호스트로 유니캐스트 방식에 따라 전송

- ARP 영역
    - ARP 요청과 응답이 일어나는 공간
    - ARP 동작은 동일한 네트워크 ID를 공유하는 호스트 i.e 같은 LAN 영역에 속하는 호스트를 대상으로 맥 주소를 구하는 기능
    - ARP 영역 자체가 LAN 영역 자체를 의미

- Type 1 :LAN 영역이 다른 출발지와 목적지 호스트
    - 출발지 호스트가 ARP 캐시 테이블을 생성하여 출발지와 목적지의 IP 주소와 맥 주소를 저장
        - 목적지의 IP 주소는 출발지 호스트와 연결된 게이트웨이에 연결
    - 게이트웨이에게 해당 정보를 주고 목적지에 데이터를 전달하는 것은 해당 게이트웨이에서 수행

- Type 2 :LAN 영역이 같은 출발지와 목적지 호스트
    - 출발지 호스트가 ARP 캐시 테이블을 생성하여 출발지와 목적지의 IP 주소와 맥 주소를 저장
    - 해당 주소가 테이블에 업데이트되고 해당 주소로 데이터 전송

- 명령어 ping 8.8.8.8 예시(출발지와 목적지의 LAN 영역이 다른 경우)
    - 출발지 및 목적지 정보
        - 출발지 IP : 183.100.206.166 / 255.255.255.0
        - 기본 게이트 웨이 : 183.100.206.1
        - 목적지 IP : 8.8.8.8 /255.255.255.0
    - 출발지 호스트는 자신의 ARP 캐시 테이블에 출발지와 목적지의 IP 주소 및 맥 주소를 저장
    - 출발지 네트워크 ID와 목적지 네트워크 ID를 비교
        - 서로 다른 것은 목적지와 출발지가 다른 LAN 영역을 의미
        - 운영체제는 목적지 IP 주소를 라우터의 IP 주소로 변경
            - 목적지 IP : 183.100.206.1
    - 기본 게이트웨이에 해당하는 목적지 맥 주소를 알아야 한다
        - (why?) 출발지 호스트와 동일한 LAN 영역에 위치한 게이트웨이까지 스위칭 통신을 할 수 있기 때문이다
    - 출발지 호스트에서는 목적지 맥 주소를 모른다
        - (=>)출발지 호스트는 자기와 동일한 네트워크 ID를 사용하는(같은 LAN 영역의) 모든 호스트 대상으로 목적지

IP(183.100.206.1)에 대응하는 맥 주소를 구하기 위해 ARP 질의를 브로드캐스트 방식으로 전송
        - 브로드캐스트(broadcast) : 자신과 동일한 네트워크 ID를 사용하는 모든 호스트에게 데이터 전송
    - 출발지 호스트와 같은 LAN 영역의 모든 호스트는 ARP 질의를 받고, 게이트웨이가 해당 호스트의 맥 주소를 요청한다는 사실 인지
        - 자신의 맥 주소를 게이트웨이이게 유니캐스트 방식으로 전송
        - 유니캐스트(unicast) : 특정한 호스트에게 데이터를 전송
        - 출발지 호스트의 ARP 캐시 테이블에서 목적지 맥 주소 업데이트
    - 출발지 호스트의 ARP 캐시 테이블에 목적지 맥 주소가 올라오면 운영체제에서 테이블을 참조하여 "유니캐스트 방식"으로 게이트웨이에 전송
    - 이후, 게이트웨이가 IP 주소에 기반한 라우팅 통신을 통해 전송

- 명령어 ping 183.100.206.166 예시(출발지와 목적지의 LAN 영역이 같은 경우)
    - 출발지 및 목적지 정보
        - 출발지 IP : 183.100.206.166 / 255.255.255.0
        - 기본 게이트 웨이 : 183.100.206.1
        - 목적지 IP : 183.100.206.1 /255.255.255.0
    - 출발지 호스트는 자신의 ARP 캐시 테이블에 출발지와 목적지의 IP 주소 및 맥 주소를 저장
    - 출발지 네트워크 ID와 목적지 네트워크 ID를 비교
        - 서로 같은 것은 목적지와 출발지가 같은 LAN 영역을 의미
    - 운영체제는 ARP 캐시 테이블에서 목적지 IP 주소에 대응하는 맥 주소의 존재를 검색
        - 테이블에 해당 맥 주소가 없다면,자기가 속한 LAN 영역 전체를 대상으로 목적지에 대응하는 맥 주소를 구하기 위해 ARP 질의를 브로드캐스트 방식으로 전송
    - 목적지 IP 주소를 사용하는 게이트웨이가 자신에 대한 맥 주소를 유니캐스트 방식으로 ARP 응답을 전송
    - 출발지 호스트는 자신의 ARP 캐시 테이블에 목적지 호스트로부터 응답받은 내용을 반영
    - 출발지 호스트의 ARP 캐시 테이블에 목적지 맥 주소가 올라오면 운영체제에서 테이블을 참조하여 "유니캐스트 방식"으로 게이트웨이에 전송

- LAN 영역의 정의

    - 동일한 "네트워크 ID"를 공유하는 공간
    - 맥 주소에 기반해 "스위칭" 방식으로 내부 통신을 수행하는 공간
    - 단일한 "ARP 영역"을 생성하는 공간

[DNS 캐시 테이블의 이해]

- DNS 캐시 테이블
    - 도메인 네임과 IP 주소의 대응 관계를 저장한 테이블
    - 명령창에서 ipconfig/flushdns, ipconfig/displaydns 명령어를 통해 DNS 캐시 테이블을 삭제, 출력 가능

- 목적지 주소로 입력한 도메인 네임을 IP 주소로 바꾸는 과정
    - DNS 캐시 테이블에서 해당 도메인 네임에 대응하는 IP 주소 검색
    - Case 1 :테이블에 해당 도메인 네임이 검색된다면, 대응하는 IP 주소로 바로 이동
    - Case 2 :테이블에 해당 도메임 네임이 없다면, 운영체제는 로컬 DNS 서버의 IP주소로 도메인 네임에 대한 질의 요청
        - 운영체제가 로컬 DNS 서버로부터 도메인 네임에 대한 IP 주소를 응답받으면 해당 응답 내용을 DNS 캐시 테이블에 반영
        - ARP 요청을 받아 목적지 맥 주소를 ARP 캐시 테이블에 반영하는 것과 비슷

- ARP 동작 vs DNS 동작
    - ARP 동작
        - 출발지와 목적지 네트워크 ID 비교
        - ARP 캐시 테이블 검색
        - ARP 요청과 응답 수행
        - ARP 캐시 테이블에 목적지 맥 주소 반영
    - DNS 동작
        - hosts 파일 검색
        - DNS 캐시 테이블 검색
        - DNS 요청과 응답 수행
        - DNS 캐시 테이블에 목적지 IP 주소 반영
    - 공통점
        - ARP 캐시 테이블은 IP 주소와 맥 주소의 대응 관계를 저장하는 일종의 데이터 베이스

        - DNS 캐시 테이블은 도메인 네임과 IP 주소의 대응 관계를 저장하는 일종의 데이터베이스

- 스푸핑(Spoofing) 공격
    - 데이터베이스에서 부적절한 대응 관계를 유발하여 사용자가 원하는 목적지로 못 가게 하는 공격
    - ARP 스푸핑 공격
        - ARP 캐시 테이블에 저장한 대응 관계 조작
    - DNS 스푸핑 공격
        - 파밍(farming) 공격
        - DNS 캐시 테이블에 저장한 대응 관계 조작
        - hosts 파일 변조를 통해서도 가능 => hosts 파일을 우선적으로 참조하기 때문에 치명적
    - DHCP 스푸핑 공역
        - (NOTE) DHCP 서버는 클라이언트에게 DNS 서버 IP 주소를 할당
        - 공격자가 가짜 DNS 서버 IP 주소를 하당하면 클라이언트는 해당 가짜 DNS 서버를 이용할 수 밖에 없다

[UDP 방식과 TCP 방식]

- UDP(User Datagram Protocol) 방식
    - 송신 호스트에서 송신 데이터가 생기면 곧바로 수신 호스트에게 전송을 수행
    - 수신 호스트의 수신 여부 고려 X

- TCP : 3단계 연결 설정(3-Way Handshaking)
    - TCP 방식에서 데이터 전송 전에 수행하는 일련의 과정
    - Step 1 : 수신 가능 여부 묻기
        - 송신 호스트에서 수신 호스트로 SYN 동기화 신호 전송
    - Step 2 : 요청 수락
        - SYN 신호를 받은 수신 호스트가 ACK&SYN 신호 전송
        - SYN은 수식 측에서 역으로 송신 측에게 동기화를 요청한다는 의미
        - 송신 호스트가 일정 시간 내로 ACK&SYN 신호를 받지 못하면 다시 SYN 신호를 전송(goto Step1)
    - Step 3 : 데이터 전송 준비 완료
        - ACK&SYN 신호를 받은 송신 호스트는 ACK 신호를 전송
        - ACK는 수신 측의 동기화 요청 신호인 SYN에 대한 응답
    - Step 1~3 동기화 과정을 거친 뒤 송신 호스트와 수신 호스트는 실제 데이터를 전송

- TCP : 3/4단계 연결 종료(3/4-Way Terminating)
  - TCP 방식에서 데이터 전송 후 수행하는 과정
  - Step 1 : 연결 종료 여부 묻기
    - 데이터 전송을 끝낸 송신 호스트는 수신 호스트로 FIN 신호 전송
  - Step 2-1 : 데이터 처리가 완료된 경우
    - FIN 신호를 받은 수신 호스트는 ACK&FIN 신호 전송
  - Step 2-2 : 데이터 처리 중인 경우
    - FIN 신호를 받았을 때, 수신 측에서 아직 데이터 처리 중인 경우 ACK 신호를 먼저 전송
    - 데이터 처리를 모두 끝낸 후 FIN 신호 전송
  - Step 3 : 연결 종료 완료
    - ACK와 FIN 신호를 받은 송신 호스트는 ACK 신호를 전송
    - ACK는 수신 측의 동기화 요청 신호인 SYN에 대한 응답

- TCP(Transmission Control Protocol)
  - 3단계 연결 설정을 통해 송신자와 수신자 사이의 연결을 확립
  - 데이터 전송 과정
    - 송신 측에서 데이터를 전송한 뒤 수신 측으로부터 ACK 신호를 받을 때까지 대기
    - 수신 측에서는 데이터를 성공적으로 수신 받으면 ACK 신호를 송신 측에게 전송
    - 일정 시간 내로 ACK 신호가 없다면 전송 중 오류 발생으로 판단하여 데이터 재전송
  - 3/4단계 연결 종료를 통해 송신자와 수신자 사이의 연결을 끊음

- TCP vs UDP
  - TCP 방식에서 3단계 연결 설정과 3/4단계 연결 종료 수행 가능한 이유
    - TCP는 버퍼링(Buffering) 방식
    - i.e UDP 방식과 TCP 방식은 버퍼링 유무에 따라 구분 가능
  - UDP 방식
    - 버퍼링 기능이 없어서 일방적 전송만 수행
    - 시간에 민감한 환경에서 사용
    - 신뢰성은 떨어지지만 빠른 전송 가능
    - 예시 : DNS, DHCP, SNMP
  - TCP 방식
    - 버퍼링 기능을 통해 송신자와 수신자 사이에 일련의 상호작용(연결/종료) 가능
    - 안정적인 전송을 요구하는 환경에서 사용

        - 전송 중 오류는 제어 가능한 반면 버퍼링 과정으로 인한 처리 지연 발생
        - 예시 : FTP, SSH, TELNET, SMTP, HTTP

- 소켓(Socket)
    - 포트 번호(Port Number)와 IP 주소를 통칭
        - 포트 번호는 가상적인 주소(전송 계층)
        - cf) 맥 주소는 물리적 주소(데이터 링크 계층)
    - 운영체제가 논리적인 방식에 따라 서로 떨어진 두 대의 호스트를 연결해주는 인터페이스 의미
    - 소켓 생성 : 운영체제가 통신에 필요한 내부 자원을 할당한다는 의미

[데이터 전송 단위]

- 편지 봉투와 편지지 예시
    - 편지지에 내용을 적는다
    - 편지 봉투에는 보내는 사람과 받는 사람의 주소를 적는다
    - 편지지는 실제 정보를 담는 부분
    - 편지 봉투는 주소 정보를 담는 부분

- 운영체제가 전송하는 데이터 구성 : 페이로드|헤더1|헤더2|헤더3...

- 페이로드(payload)
    - 편지지의 내용인 데이터에 해당
    - 상대방에게 전송하고자 하는 실제 정보가 담긴 공간

- 헤더(header)
    - 편지 봉투에 해당
    - 출발지 주소와 목적지 주소가 담긴 공간

- 메시지(message)
    - 페이로드만으로 이루어진 데이터 전송 단위
    - 편지지만 있는 상태

- 데이터그램(datagram)/세그먼트(segment) - 전송 계층
  - 페이로드 앞에 붙는 첫 번째 헤더
  - 핵심 정보 : 포트 번호
    - 출발지 포트 번호
    - 목적지 포트 번호
  - 데이터 구성 : UDP/TCP 페이로드|데이터그램/세그먼트 헤더
  - 페이로드의 속성이 UDP 속성이면 데이터그램 헤더 / TCP 속성이면 세그먼트 헤더라고 부른다
  - 단편화(Fragmentation)
    - 데이터의 분할
    - 생성한 페이로드 영역을 여러 개 조각낸 뒤 전송하는 기법
    - 전송의 효율성과 데이터의 기밀성 등을 위해 사용
    - TCP 방식에만 존재
    - 단편화가 없는 UDP 속성에서는 데이터그램 / 단편화가 있는 TCP 속성에서는 세그먼트라고 한다
  - 버퍼링과 단편화 유무에 따라 UDP와 TCP 방식을 구분

- 포트 번호(Port Number)
  - 데이터그램 헤더의 핵심
  - 상대방(송신) 운영체제에서는 데이터그램 헤더에 담긴 포트 번호를 통해 페이로드의 내용이 어떤 종류의 서비스에 해당하는지 판단
  - 출발지 및 목적지 포트 번호
    - 출발지 포트 : 1024번 이후의 포트 번호
    - 목적지 포트 : 서비스(FTP,SSH,HTTP..)에 해당하는 포트 번호
  - 예시 : 웹 사이트 접속 (by HTTP)
    - 출발지/목적지 운영체제에서 해당 네트워크의 적절한 포트 번호 할당
    - 출발지 : 임의의 1024번 이후의 포트 번호 할당
    - 목적지 : HTTP에 해당하는 80번 포트 번호 할당

- 패킷 헤더(Packet Header) - 네트워크 계층
  - 데이터그램 앞에 붙은 두 번째 헤더
    - UDP 페이로드|데이터그램|패킷
  - 핵심 정보 : IP 주소
    - 출발지 IP 주소
    - 목적지 IP 주소
  - 라우팅 장비가 라우팅 기능을 수행할 때 참조하는 부분
    - 라우터 운영체제가 패킷 헤더를 참조하여 라우팅 기능을 수행

- 프레임 헤더(Frame Header) - 데이터 링크 계층
  - UDP 페이로드|데이터그램|패킷|프레임
  - 핵심 정보 : 맥 주소
    - 출발지 맥 주소
    - 목적지 맥 주소
  - 스위치 장비가 스위칭 기능을 수행할 때 참조하는 부분
  - (BUT) 프레임 헤더는 스위칭 통신 영역/라우팅 통신 영역에 있을 때 담는 정보가 다르다
    - 프레임 헤더가 LAN 영역에 있을 때는 맥 주소
    - 프레임 헤더가 WAN 영역에 있을 때는 다른 정보(See CH.14)
    - LAN/WAN 영역에서의 데이터그램 헤더와 패킷 헤더의 주소는 변하지 않는다 i.e 프레임 헤더 앞 부분의 헤더들에 대한 영향 X
  - 패킷 헤더에서 출발지와 목적지의 네트워크 ID가 다르면 프레임 헤더의 목적지 맥 주소는 라우터의 맥 주소를 의미

- Summary
  - 비트(Bit)
    - 프레임 단위를 생성한 데이터
    - 페이로드~프레임까지의 정보
  - 메시지 → 페이로드|데이터그램|패킷|프레임 → 비트
  - 인캡슐레이션(Encapsulation)
    - 생성한 페이로드 앞에 일련의 헤더를 붙이는 과정
    - 운영체제가 메시지 → 데이터그램 → 패킷 → 프레임 → 비트로 변환하여 데이터를 송신을 하는 과정
  - 디캡슐레이션(Decapsulation)
    - 비트로부터 일련의 헤더를 떼는 과정
    - 운영체제가 비트 → 프레임 → 패킷 → 데이터그램 → 페이로드로 변환하여 데이터를 수신하는 과정
  - 인캡슐레이션 과정은 OOP에서 사용하는 정보 은폐 과정
    - 라우터는 오직 패킷 해더만 읽을 수 있고, 데이터그램 헤더는 읽을 수 없다
    - 스위치는 오직 프레임 헤더만 읽을 수 있고, 패킷 헤더는 읽을 수 없다

[TCP/IP 방식의 계층적 구조]

- 프토토콜(Protocol)
  - 호스트와 호스트 사이에서 사용하는 일종의 언어같은 개념
  - 송신자와 수신자 사이에 동일한 프로토콜을 설정해야만 통신이 가능
  - 예시 : IP, TCP, UDP, DHCP, DNS

- TCP/IP 프로토콜 5 계층
  - 응용/프로세스 계층
  - 전송 계층(TCP)
  - 네트워크/인터넷 계층(IP)
  - 데이터 링크 계층
  - 물리 계층
  - ※ 4 계층으로 구분할 경우, 데이터 링크 + 물리 계층을 네트워크 인터페이스/네트워크 접근 계층이라고 부른다

- 계층(Layer)
  - 비음성 통신에서 데이터를 전송하기 위한 일련의 과정/단계/절차
  - 예시 : TCP/IP 프로토콜 5 계층, OSI 7 계층
    - 역사적인 이유로 TCP/IP 프로토콜이 사실상 표준이 되었다
  - 송신자 운영체제는 응용 계층에서 시작해서 물리 계층을 순차적으로 통과하면서 데이터를 전송(인캡슐레이션)
  - 수신자 운영체제는 물리 계층에서 시작해서 응용 계층을 순차적으로 통과하면서 데이터를 수신(디캡슐레이션)
  - 응용 계층에서 물리 계층까지 하위 계층으로 내려갈수록 속성이 구체적
  - 물리계층에서 응용 계층으로 올라갈수록 속성이 논리적이고 추상적
  - 데이터의 송신 과정
    - 상위 계층에서 하위 계층으로 변환하는 과정
    - 논리적 속성이 물리적 속성으로 변환하는 과정
    - 일련의 부가 정보를 추가하는 과정
    - 응용 계층에서 완성한 메시지는 각 계층을 통과하면서 헤더들이 결합
  - 데이터의 수신 과정
    - 하위 계층에서 상위 계층으로 변환하는 과정
    - 물리적 속성이 논리적 속성으로 변환하는 과정
    - 물리 계층에서 수신한 데이터는 각 계층을 통과하면서 헤더들이 분리

[TCP/IP 방식의 응용 계층]

- 페이로드를 생성해주는 계층
  - 운영체제는 수신자에게 전송할 데이터를 생성 및 전송
  - 페이로드 생성 프로토콜이 존재
    - ex) DNS, HTTP

- 포트 번호
  - 응용 계층은 실제 정보를 저장하는 페이로드 생성 기능을 수행
  - 응용 계층에 속하는 프로토콜을 고유한 식별자 번호(포트 번호)로 인식
  - 응용 계층에서는 포트 번호를 통해 무수한 프로토콜을 구분
  - 해당 프로토콜에서 발생한 정보가 흐르는 가상의 통로
    - 해당 포트 번호에서는 특정 데이터 유형을 처리
    - 25번(SMTP) : 전자 우편과 관련된 내용
    - 53번(DNS) : 도에인 네임에 대한 질의와 응답 내용

- 포트 번호 유형
  - 잘 알려진 포트 번호(Well-Known Port Number)
    - 주로 서버 측에서 사용
    - 번호 : 0 ~ 1023
  - 등록 포트 번호
    - 주로 클라이언트 측에서 사용
    - 번호 : 1024 ~ 49151
  - 사설 또는 동적 포트 번
    - 주로 클라이언트 측에서 사용
    - 번호 : 49152 ~ 65535

- 포트 스캔(Port Scan)
  - 원격지 호스트를 대상으로 어떤 포트 번호를 사용 중인지 확인하는 기법
  - 예시
    - 서버 측 운영체제제에서는 특정 서비스를 구동하면서 외부로부터 접속을 받기 위해 특정 서비스에 해당하는 특정 포트를 개방
    - 해당 서비스를 요청하는 클아이언트는 해당 서버로 DNS를 요청하지 않고도 Nmap(포트 스캔 도구)이라는 포트 스캐너를 이용하여 해당 포트 번호의 활성 상태 여부를 확인 가능

[TCP/IP 방식의 전송 계층]

- 전송 계층
  - 응용 계층에서 생성한 페이로드에 포트 번호 정보가 담긴 헤더를 붙이는 계층

    - 전송 계층의 프로토콜은 단 2개이다
        - UDP 프로토콜
        - TCP 프로토콜

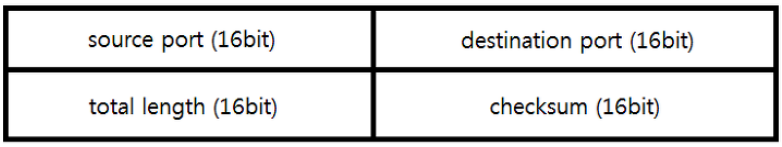
- UDP 방식의 전송 계층
    - 응용 계층에서 생성한 페이로드에 출발지/목적지 포트 번호가 담긴 헤더를 붙이고 네으워크 계층으로 전송 과정을 넘긴다
    - 512 바이트 미만의 페이로드를 대상으로 전송 과정에만 초점을 두고 개발
    - 실시간 요구하는 환경에 적합한 구조

- TCP 방식의 전송 계층 : 3단계 연결 설정
    - UDP 방식과 다르게 전송 전에 3단계 연결 설정을 수행(버퍼링)
    - Step 1 : 송신 호스트의 연결 요청
        - 응용 계층에서 생성한 페이로드를 응용 계층 버퍼에 임시 저장하고, 전송 계층에서 SYN 신호를 담은 세그먼트 1개를 생성
        - 해당 SYN 세그먼트는 네트워크-데이터 링크 계층을 차례대로 통과하면서 헤더가 붙여지고, 물리 계층에서 비트 단위로 변환해 목적지(수신지)로 전송
        - SYN 신호 담긴 세그먼트 헤더 | IP 헤더 패킷 | 이너넷 프레임 헤더
    - Step 2 : 수신 호스트의 연결 수락
        - 해당 SYN 신호를 전송 계층으로 끌어올리고 전송 계층에서 ACK&SYN 신호를 담은 세그먼트를 1개를 생성
        - 해당 ACK&SYN 세그먼트는 송신지로 전송
        - ACK&SYN 신호가 담긴 세그먼트 | IP 헤더 패킷 | 이너넷 프레임 헤더
    - Step 3 : 송신 호스트의 데이터 전송 준비
        - 해당 ACK&SYN 신호를 전송 계층으로 끌어올리고 전송 계층에서 ACK 신호를 담은 세그먼트 1개를 생성
        - 해당 ACK 세그먼트는 수신지로 전송
        - ACK 신호가 담긴 세그먼트 | IP 헤더 패킷 | 이너넷 프레임 헤더

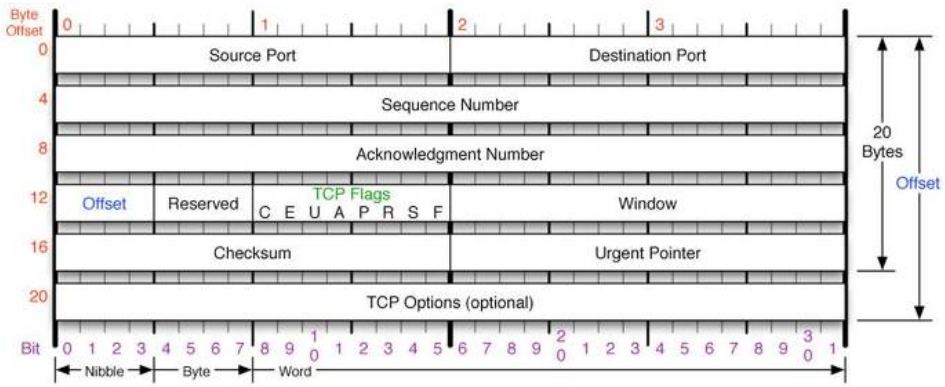
- TCP 방식의 전송 계층 : 데이터 전송
    - 3단계 연결 설정을 통해 송신-수신 연결이 확립되면, 운영체제는 응용 계층 버퍼에 저장 했던 TCP 페이로드를 전송 계층으로 넘긴다
    - 응용 계층에서 받은 TCP 페이로드를 대상으로 단편화 수행
        - 단편화 : 전송 효율성과 데이터 기밀 성을 위해 페이로드를 여러 개로 분할하는 기법

    - 단편화 후 조각난 페이로드마다 출발지/목적지 포트 번호가 담긴 헤더를 붙여 다수의 세그먼트 생성
        - cf) UDP 방식은 1개의 페이로드에 대해 1개의 세그먼트만 생성

- 전송(transmission)은 어느 계층에서나 시작 할 수 있다
    - 무조건 응용 계층에서 시작하지는 않는다
    - Example : 전송 전 3단계 연결 설정은 전송 계층에서 시작
    - 해당 계층에서 전송이 시작되면, 송신지 또한 해당 계층에서 전송이 끝난다

- UDP 헤더
    - 헤더의 크기는 8바이트(64비트)
    - 

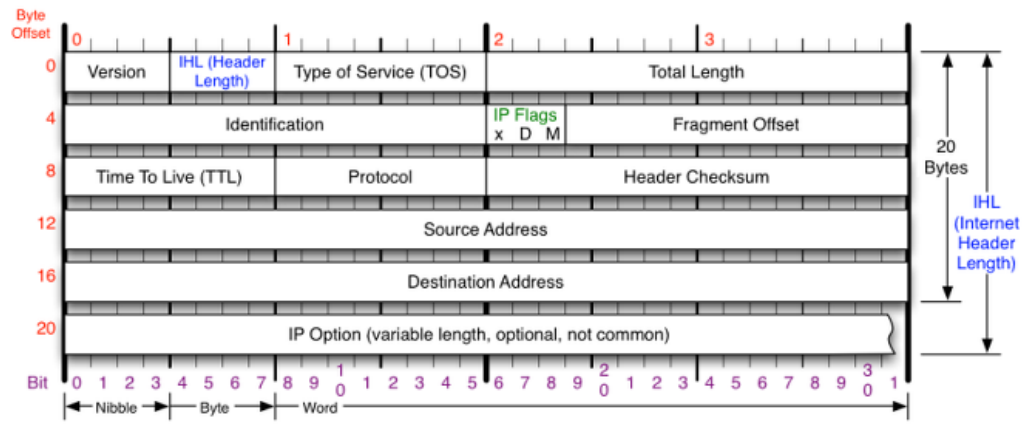
    - 출발지/목적지 포트
        - 각각 16비트로 구성
        - 응용 계층에 속하는 프로토콜의 종류가 65536(2^16)인 이유
    - 길이(Length)
        - 데이터그램(UPD 페이로드 + UDP 헤더)의 크기 정보
    - 오류 검사(Checksum)
        - 기본적으로는 비활성 상태
        - 데이터가 전송 중에 손상되지 않고 원본과 동일한지 여부를 확인하는 기능 제공
        - incorrect일 경우 송신지에게 패킷의 재전송을 요구

- TCP 헤더
    - 헤더의 크기는 가변적
        - 일반적으로는 20바이트(160 비트)
        - TCP 추가(TCP options) 항목을 이용하여 21바이트 이상으로 사용 가능
    - 

- 출발지/목적지 포트
  - 각각 16비트로 구성
  - UDP 헤더와 동일
- 일련 번호(Sequence Number)
  - 32비트로 구성
  - 단편화에 의해 분할된 세그먼트들의 고유한 ordered 번호
  - 불안정한 네트워크로 인한 패킷의 분실, 지연 등으로 세그먼트 순서가 어긋나게 도착하여도 일련 번호를 이용하여 데이터를 올바르게 재배열 가능
  - 크기가 32비트이기 때문에 총 2^32 개수의 세그먼트 송신 가능
- 확인 번호(Acknowledgement Number)
  - 32비트로 구성
  - 수신하기를 기대하는 다음 일련 번호
  - 다음 세그먼트를 수신할 준비가 되었다는 사실을 알린다 i.e 상대방이 보낸 세그먼트를 잘 받았다는 것을 알려주는 번호
  - 모든 데이터가 수신되었다는 것을 나타내는 묵시적인 확인 메시지 역할
- 일련 번호 & 확인 번호
  - (전송 전) 3단계 연결 설정 및 (전송 후) 3단계 연결 종료 등에 관련
  - 송신자와 수신자 사이에 주고받는 세그먼트의 연속성을 보장하기 위하여 일련번호와 확인번호 이용
  - 수신 측에서 단편화한 세그먼트 단위를 순서대로 재조립할 때도 중요한 정보
- 오프셋(Offset)
  - 3비트로 구성
  - 헤더 길이에 대한 정보
  - TCP 헤더는 가변적이기 때문에 헤더의 크기를 담는 항목이 필요
  - 일반적으로 20
- 플래그(Flag)
  - 6비트로 구성
  - (전송 전) 3단계 연결 설정 및 (전송 후) 3단계 연결 종료 등에서 사용하는 SYN, FIN 신호 같은 일종의 제어 정보 저장
  - Example : ACK&SYN 신호는 ACK 플래그와 SYN 플래그를 동시 설정
  - TCP 방식이 수행하는 일련의 정보 저장
  - 총 8(2^3)개의 플래그
    - CWR(Congestion Window Reduced) : 혼잡 윈도우 크기 감소 i.e 송신자에게 전송 데이터를 줄여서 보내라는 의미
    - ECN(Explicit Congestion Notification) : 혼잡 발생

        - URG(Urgent) : 긴급 데이터
        - ACK(Acknowledgement) : 확인 응답
        - PSH(Push) : TCP 페이로드를 포함한다는 신호
        - RST(Reset) : 상대방과의 연결을 강제로 종료한다는 신호
        - SYN(Synchronize) : 상대방과 동기화를 확립하기 위한 개시 신호
        - FIN(Finish) : 상대방과 동기화 해체를 위한 종료 신호
    - 윈도우(Window)
        - 16비트로 구성
        - 흐름 제어 기능과 관련
        - 플래그(Flag)와 밀접한 관계가 존재
        - Example : 혼잡과 부하로 인해 데이터를 부분적으로 수신했다면, 다음으로 받아야 할 정보를 윈도우에 저장하고, 플래그 항목에 ACK&CWR 플래그로 응답
        - 혼잡 윈도우(Congestion Window) : 송신자가 수진자의 확인 응답에 따라 전송할 데이터의 양 조절
        - 슬라이딩 윈도우(Sliding Window) : 송신자가 전송할 수 있는 정보의 양

[TCP/IP 방식의 네트워크 계층]



- 전송 계층에서 전송된 각각의 세그먼트 앞에 IP 주소를 주요한 정보로 하는 헤더를 추가하여 패킷 생성

- IP 헤더 크기는 TCP 헤더처럼 가변적이다
    - 일반적으로는 20바이트(160 비트)
    - IP 추가(IP Options) 항목을 이용하여 21바이트 이상으로 사용 가능

- 버전(Version)
  - IPv4 주소는 4가 들어가고, IPv6 주소는 6이 들어간다

- 헤더 길이(Header Length)
  - IP 헤더의 크기 정보
  - 일반적으로 20

- 서비스 종류(ToS, Type of Service)
  - 해당 패킷의 전송 우선 순위를 저장
  - 회선이 혼잡할 경우 우선 순위에 따라 해당 패킷을 우선적으로 전송

- 전체 길이(Total Length)
  - IP 헤더를 포함한 패킷 전체(헤더+데이터)의 길이 정보
  - 전체 길이 - 헤더 길이 : 남은 크기

- ID & IP 플래그 & 프래그먼트 오프셋
  - MTU에 따른 패킷 분할 정보를 담는 항목
  - MTU(Maximum Transmission Unit)
    - 최대 전송 단위
    - 각 프로토콜에서 정한 패킷 크기의 최대 범위
  - MTU를 초과한 패킷이 발생하면, ID/IP플래그/프래그먼트오프셋 항목들이 필요

- 패딩(Padding)
  - MTU를 초과한 패킷을 분할 할때 필요한 과정
  - 해당 패킷의 크기를 MTU의 배수로 만들기 위해 더미 데이터를 채우는 과정

- ID(Identification)
  - 각 패킷을 식별하는 번호
  - 패킷을 재조합 할때 사용

- IP 플래그(IP Flags)
  - 패킷이 단편화되었는지 아닌지 단서를 제공하는 역할
  - 2개의 비트를 이용하여 패킷의 분할 유무 표시

        - 원래는 3개의 비트로 구성되지만 실제로는 2개의 비트만 사용
        - D 비트와 M 비트로 구성
        - cf) TCP 헤더의 플래그는 제어 신호를 설정하는데 사용
    - D(Do not fragment)비트
        - 패킷이 분할되어 있는지 알려주는 비트
        - 0 : 분할되어 있음
        - 1 : 분할되어있지 않음
    - M(More fragments) 비트
        - 분할된 패킷이 더 있는지 알려주는 비트
        - NULL : 패킷 분할이 이루어지지 않은 경우
        - 0 : 마지막 패킷을 의미
        - 1 : 마지막 패킷이 아니고, 분할된 패킷이 더 존재한다는 의미

- 프래그먼트 오프셋(Fragment Offset)
    - 패킷 재조립시 분할된 패킷 간의 순서에 대한 정보
    - 전체 데이터에서 분할된 패킷의 상대 위치를 표현
    - Example
        - 6000 바이트 패킷, 1500 바이트 MTU
        - 각 분할된 패킷의 프래그먼트 오프셋은 0/1500/3000/4500 이다
    - 티얼드롭(TearDrop) 공격
        - IP 헤더에서 사용하는 프래그먼트 오프셋에 담긴 정보가 애매하면 수신측에서 분할 패킷을 재조립할 때 비정상적으로 처리
        - 패킷 단위에서 재조립이 일어나는 특징을 악용한 공격
    - 본크 보인크(Bonk Boink) 공격
        - TCP 헤더에서 사용한는 일련번호에 담긴 정보가 애매하면 수신측에서 단편화 세그먼트를 재조립할 때 비상적으로 처리
        - 세그먼트 단위에서 재조립이 일어나는 특징을 악용한 공격

- 생존 기간(TTL, Time To Live)
    - 라우팅 루프가 일어난 구간에서 패킷을 폐기하기 위한 용도로 사용
    - 해당 패킷이 통과할 수 있는 라우터의 개수 정보를 담는 항목

- 프로토콜(Protocol)
    - 상위 계층(전송 계층)에 속한 프로토콜 번호 저장
    - UDP 페이로드는 17, TCP 페이로드는 6으로 설정

    - 프로토콜 항목을 보고, 해당 패킷을 데이터그램/세그먼트로 간주

- 헤더 오류 검사(Header Checksum)
    - IP 패킷 헤더의 오류 발생 여루블 확인하기 위한 필드
    - 기본적으로는 비활성 상태

- 사설 IP 주소
    - LAN 전용 주소 or 내부 IP 주소
    - IP 주소의 고갈을 완화하는 방법 중 하나
    - LAN 영역에서 전용으로 사용 할 수 있는 주소이지만, NAT 기법을 이용하여 실제 외부 인터넷에 접속 가능
    - NAT(Network Address Translation)
        - 출발지 사설 IP 주소를 출발지 IP 주소로 바꾸는 기법
        - 보통 라우터 같은 장비에서 이용
    - PAT(Port Address Translation)
        - NAT 기법에 포트 번호 주소를 연동하여 사용하는 방법
        - 이론적으로 1개의 공인 IP 주소에 65536개의 사설 IP 주소 연결 가능
    - 보안 측면에서는 외부에서 직접 접근이 불가능하다는 장점 존재
        - NAT 기법을 통해 외부로는 나갈 수 있으나, 외부에서 실제 내부 IP 주소를 이용하는 PC로 직접적인 접근은 불가능
        - 외부에서 접근하기 위해서는 포트 포워딩을 설정해야 한다
    - 포트 포워딩(Port Forwarding)
        - PAT 기법을 응용하여 내부로 접근할 수 있도록 설정하는 기법
        - 설정을 통해 외부에서 공인 IP 주소를 입력해도 사설 IP 주소로 접속 가능

- ICMP(Internet Control Message Protocol)
    - 화면 출력 메시지에 기반해 오류 통보 기능과 질의/응답 기능 등을 수행하기 위한 프로토콜
        - 오류 통보 기능 : 전송 중 일어날 수 있는 목적지 도달 불가/발신지 억제/시간 초과/매개변수의 문제 등을 사용자 화면에 출력하기 위한 기능
        - 사용자 입장에서는 질의/응답 기능이 더 중요
        - IP 등장 이후 전송 작업을 화면에 출력하기 위한 용도로 등장
        - ex) ping 명령어는 ICMP에 기반
    - 네트워크 계층에서 페이로드 생성

        - 호스트가 ICMP 질의 요청 시 운영체제는 네트워크 계층에 기반해 쓰레기 값이 채워진 페이로드 생성
        - cf) FTP나 SSH는 응용 계층에서 사용자의 실제 정보를 담은 페이로드 생성
    - ICMP 헤더
        - 타입, 코드, 오류 검사, ICMP 페이로드
        - 타입 : 해당 ICMP가 요청인지 응답인지 구분하기 위한 정보
    - ICMP의 인캡슐레이션
        - ICMP 페이로드 | ICMP 헤더 | IP 헤더
    - ping 명령어
        - 출발지와 목적지 사이의 통신 여부를 점검하는 용도
        - 출발지 호스트에서 임의의 쓰레기 값으로 이루어진 데이터를 생성하여 목적지로 송신
        - 목적지 호스트가 동작 중이면 응답이 출발지로 온다
    - tracert 명령어
        - 출발지와 목적지 사이의 라우팅 과정을 확인하기 위한 용도
        - IP 헤더의 TTL 속성을 이용하여 구현
            - TTL 값을 0 에서부터 순차적으로 늘려가면서 목적지 호스트의 ICMP 응답이 돌아 올때까지 반복
        - 윈도우 계열은 ICMP 방식을, 유닉스/리눅스 계열은 UDP 방식을 이용
        - 동적으로 동작하는 라우팅 알고리즘때문에 경로 추적 시 경로가 유일하지 않음
    - ICMP 스머핑(ICMP Smurfing) 공격
        - 브로드캐스트 IP 주소의 속성을 악용한 공격

- ARP, RARP 프로토콜
    - 네트워크 계층과 데이터 링크 계층 사이에서 동작하는 프로토콜
    - IP 주소와 MAC 주소를 대응시켜주는 기능
    - 통상 네트워크 계층으로 분류

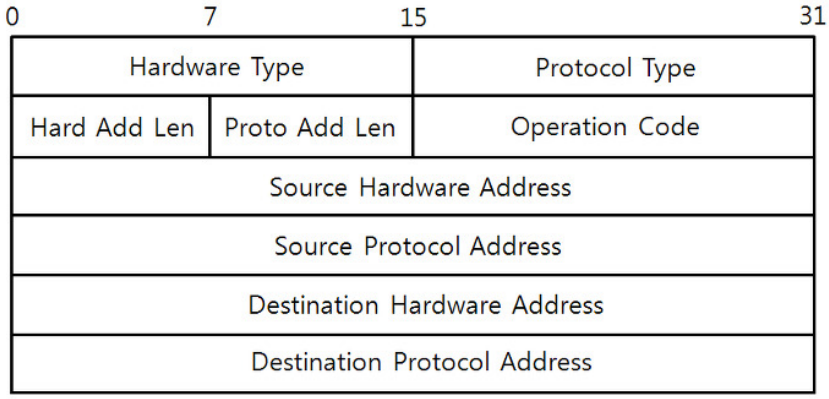
[TCP/IP 방식의 데이터 링크 계층]

- 데이터 링크 계층에 속하는 프로토콜은 LAN/WAN 영역에서 사용하는 프로토콜에 해당

- LAN 영역

    - 근거리 통신망
        - 같은 네트워크 IP를 가진 호스트들의 집합
    - 이더넷, 토큰 링, FDDI(Fiber Distrubuted Interface) 방식

- WAN 영역
    - LAN 영역을 벗어나 라우터 간의 통신이 있는 영역
    - HDLC(High-Level Data Link Control), PPP(Point-to-Point Protocol), 프레임 릴레이(frame relay), ATM 방식
    - LAN 영역과 달리 표준 프로토콜이 없어 상대방 라우터와 연동 시 프로토콜 설정에 주의 필요

- ARP(Address Resolution Protocol)
    - LAN 영역에서 사용하는 MAC 주소와 네트워크 계층에서 사용하는 IP 주소의 연결을 위해 설계
    - ARP 헤더는 네트워크 계층에서 생성되어 데이터 링크 계층으로 넘어간다
        - 페이로드 영역이 없고 헤더로만 이루어진 구조
        - ARP 헤더 | 이더넷 프레임 헤더
    - 

    - 연산 코드(Operation Code)
        - ARP 요청 및 응답과 관련된 정보
        - ARP 캐시 테이블(출발지/목적지의 IP/MAC 주소)를 채울 때 사용
    - ARP 브로드캐스트 요청
        - ARP 헤더의 목적지 MAC 주소 항목은 00-00-00-00-00-00 처럼 채워진다 i.e 정확한 목적지 MAC 주소를 모른다는 의미
        - ARP 헤더의 연산 코드 항목은 1(ARP 요청 의미)
        - ARP 헤더 앞에 붙은 프레임 헤더의 목적지 맥 주소는 아직 모르기 때문에 ff:ff:ff:ff:ff:ff로 설정 i.e 브로드 캐스트 의미
        - 자신이 속한 LAN 영역의 모든 호스트 대상으로 질의
    - ARP 유니캐스트 응답

        - ARP 브로드캐스트 요청을 받은 해당 라우터는 자신의 맥 주소를 저장한 ARP 헤더를 만들어 전송
    - 에이징(Aging)
        - ARP 캐시 테이블에서 일정 시간 동안 사용되지 않은 MAC 주소를 삭제하는 것
        - DNS 캐시 테이블에서도 발생

[TCP/IP 방식의 물리 계층]

- 물리 계층은 하드웨어 속성과 관련이 깊다
    - 기계적/전기적/기능적/절차적 기능 등을 정의
    - 통신에 필요한 신호 방식 or 전송 대역폭 등을 규정
    - 전산보다는 전기/전자 분야에 해당

- 물리 계층에 해당하는 장비
    - UTP 회선
    - 처리에 적합한 비트 신호를 전송에 적합한 비트 신호로 변환해주는 장치
        - DCE(Data Circuit Terminating Equipment)
        - 허브 : LAN 영역에서 사용하는 대표적인 집선 장치

- 네트워크 3대 장비 : 허브, 스위치, 라우터
    - 물리적으로 각 장비들이 장비들의 포트에 연결
        - 허브 - 스위치 - 라우터 형식으로 연결
    - 허브
        - 물리 계층에서 비트 단위를 처리하는 장치
        - 비트 단위로 들어온 신호를 수신한 포트를 제외한 다른 포트로 비트 단위를 플러딩(flooding)
        - 디캡슐레이션 과정이 없어 다른 장비들보다 이론상 가장 빠르다
    - 스위치
        - 데이터 링크 계층에서 프레임 단위를 처리하는 장치
        - 비트를 프레임으로 디캡슐레이션
        - 스위치 운영체제는 프레임 헤더에 담긴 목적지 맥 주소가 자신의 스위칭 테이블에 있는지 검색
            - 스위칭 테이블은 (포트 번호, 맥 주소) 테이블 구조
        - 스위칭 테이블에 있다면, 프레임을 비트로 변환하여 특정 포트로만 비트 신호를 포워딩(forwarding)하고, 목적지

맥 주소가 없는 포트로는 신호가 넘어가지 못하게 블로킹(blocking) 처리
        - 스위칭 테이블에 목적지 맥 주소가 없거나 ARP 요청 프레임이라면, 허브처럼 플러딩(flooding) 방식으로 동작
        - 스위칭 동작은 포워딩/블로킹/플러딩으로 구성
    - 라우터
        - 네트워크 계층에서 패킷 단위를 처리하는 장치
        - 비트 단위로 들어온 신호를 프레임으로 변환하여 자신의 이더넷 인터페이스(라우터의 LAN 카드)에 새겨진 맥 주소와 프레임 헤더의 목적지 맥 주소 비교
        - 라우터의 맥 주소와 프레임 헤더의 맥 주소가 일치 하는 경우 프레임을 패킷으로 변환
        - 라우터 운영체제는 패킷 헤더의 목적지 IP 주소가 자신의 라우팅 테이블에 있는지 검색
            - 라우팅 테이블은 (목적지 네트워크 ID, 경유지 인터페이스) 테이블 구조
        - 라우팅 테이블에 있다면, 해당 경유지 인터페이스로 포워딩 하기 위해 패킷을 프레임으로 변환하고, 프레임 헤더에 맥 주소가 아닌 PPP 정보를 삽입
        i.e 이더넷 헤더가 아닌 PPP 헤더로 변경
        - 라우팅 테이블에 목적지 네트워크 ID가 없다면, 라우팅 불가로 판단하여 해당 패킷을 폐기하고 송진자에게 ICMP 방식을 이용하여 해당 사실 통보
            - 스위치에서의 플러딩 동작과 다른 방식

[LAN/WAN 영역의 개념적 이해]

- 데이터 링크 계층과 물리 계층은 LAN/WAN 영역의 기술을 흡수하기 위해 마련한 계층

- 네트워크 ID, 호스트 ID, ARP/RARP
    - 네트워크 ID
        - 무수한 LAN 영역 중 특정 LAN 영역을 구분하기 위한 식별자
        - 라우터가 라우팅을 수행하기 위한 주소 체계
    - 호스트 ID
        - 동일한 LAN 영역에서 속한 무수한 호스트 중 특정 호스트를 구분하기 위한 식별자
        - 동일한 LAN 영역에서 호스트 사이의 통신은 맥 주소에 기반한 스위칭 통신
        - 호스트 ID와 맥 주소 사이의 연결 고리 필요
    - ARP/RARP

        - 호스트 ID와 맥 주소를 연결하기 위해 네트워크 계층과 데이터 링크 계층 사이에서 동작하는 프로토콜
        - ARP는 IP 주소에 기반해 맥 주소를 구하는 기능 수행
        - RARP는 맥 주소에 기반해 IP 주소를 구하는 기능 수행 (현재는 DHCP 기능이 RAPR 역할 수행)

- 데이터 링크 계층의 종류
    - LAN 영역 전반을 관리하는 IEEE에서 세부적으로 데이터 링크 계층의 종류를 구분
    - 논리 회어(LLC) 부계층
        - 네트워크 계층과 데이터 링크 계층의 중간 매체
        - NIC(Network Interface Controller) 드라이버 등을 구현하기 위한 계층
    - 매체 접근 제어(MAC) 부계층
        - 프레임 전송 단위를 생성하는 계층

- 물리 계층의 핵심 장치
    - 비트 단위를 처리하는 장치
    - 회선과 허브

- 데이터 링크 계층의 핵심 장치
    - 프레임 단위를 처리
    - LAN 카드와 스위치

- 데이터 링크 계층의 핵심 장치 1 : LAN 카드
    - MAC 주소가 새겨져 있다
        - IP 주소처럼 사용자가 임의로 변경할 수 있는 속성이 아니다
    - 회선을 타고 비트 신호가 오면 LAN 카드에서는 비트를 프레임으로 변환
        - 프레임 헤더의 목적지 맥 주소와 자신의 맥 주소 비교
        - 목적지와 자신의 맥 주소가 동일하면 네트워크 계층으로 보냄
        - 목적지와 자신의 맥 주소가 다르면 자신에게 오는 데이터가 아니라고 판단하여 폐기
    - 무작위 모드(Promiscuous mode)
        - LAN 카드의 맥 주소와 프레임 헤더의 목적지 맥 주소가 다르더라도 LAN 카드가 해당 프레임을 수신하는 동작
        - 허브로 LAN 영역을 구성하는 경우 허브가 플러딩 동작을 하는 장비여서 LAN 카드 자신과 다른 맥 주소가 나타날 수 있다
          => 무작위 모드 설정 필요

        - 해당 설정을 한 허브 환경에서는 출발지/목적지 호스트가 서로 주고받은 패킷을 스니핑할 수 있는 상태
          (why?) 자신의 맥 주소에 해당하지 않는 프레임도 수신하기 때문

- 데이터 링크 계층의 핵심 장치 2 : 스위치
    - 스위칭 테이블은 스위치의 각각의 포트(물리적인 접속 부위)에 대한 맥 주소로 구성
    - 러닝(learning)
        - 출발지 맥 주소를 스위칭 테이블에 반영하는 동작
        - 특정 호스트에서 비트를 전송하면, 스위칭 테이블은 해당 비트를 프레임으로 변환하고, 프레임 헤더의 출발지 맥 주소를 스위칭 테이블에 반영
        - 플러딩 방식으로 맥 주소 질의를 한 뒤 응답을 수신받으면 해당 맥 주소를 스위칭 테이블에 반영
    - 플러딩(flooding)
        - 스위치에서 MAC 주소 질의를 위해 송신 호스트와 연결된 포트를 제외한 남은 포트에 비트를 송신
        - 요청을 수신 받은 호스트들은 목적지 맥 주소와 자신의 LAN 카드 맥 주소를 비교하여 다르면 폐기, 같으면 수신
    - 에이징(Aging)
        - 일정 시간 통신이 없을 때 테이블에 있는 해당 MAC(IP) 주소를 삭제하는 동작
        - 스위칭 테이블에서 에이징이 발생하면 플러딩 발생
        - ARP 캐시 테이블에서 에이징이 발생하면 브로드캐스팅 발생

- 허브와 스위치를 서로 연동하는 방식은 부적절한 구성
    - 허브 : 비트 단위를 플러딩으로 처리
    - 스위치 : 프레임 단위를 포워딩으로 처리
    - Situation : 5개의 포트로 구성된 허브의 1~4 포트는 호스트, 5번 포트는 스위치 1번 포트에 물려 있는 상황
    - 허브의 1~4 포트는 플러딩에 따라 통신
    - 스위치가 물린 허브의 5번 포트로도 플러딩이 전해진다
        - 허브에 통신이 일어날 때 마다 스위칭 테이블에 러닝 발생 i.e 스위칭 테이블에 데이터가 계속 추가
        - 스위칭 테이블 용량이 채워진 상태에서 스위칭 테이블에 없는 목적지 맥 주소가 요청되면, 프럴딩 방식으로밖에 전송 할 수 없음
          => 포워딩으로 동작하던 스위치가 플러딩으로 동작하는 허브로 전락!
    - 맥 플러딩(MAC Flooding)
        - 스위치가 플러딩으로만 동작하도록 하는 공격 유형

- VLAN(Virtual LAN)
  - 스위치 장비의 또 다른 기능
  - 1개의 물리적인 LAN 영역을 여러 개의 논리적인 LAN 영역으로 분할하는 기법
  - 1개의 LAN 영역을 대상으로 이루어진 1개의 ARP 영역을 여러 개의 ARP 영역으로 분할하는 기법
  - ARP 요청과 응답의 본질은 운영체제에서 맥 주소를 확인하고 해당 맥 주소를 적재하는 작업
    - 라우터가 2^16~2^24(LAN 영역의 등급 별 호스트 ID 개수)의 ARP 요청/응답을 수행하면 과부하로 인해 라우팅이 불가능
    - VLAN는 통해 대규모의 ARP 영역을 여러 개의 ARP 영역으로 분할하여 각각의 ARP 영역의 크기를 줄이는 기능
  - VLAN 기법을 이용하면 호스트 ID 개수는 줄어든다
    - ARP 영역의 개수는 늘어나지만 해당 ARP 영역의 크기는 감소
    - 규모가 줄어든 만큼 ARP 요청과 응답의 빈도가 줄어든다
  - 물리적으로는 같은 공간에 있지만 논리적으로는 각기 다른 LAN 영역으로 분리
  - LAN 통신은 내부 통신이기 때문에, 논리적으로 분리된 다른 LAN 영역 간 통신이 불가능
    - 외부 LAN 영역으로부터 침해를 구조적으로 차단
  - 인터 VLAN 라우팅, L3 스위칭
    - 서로 다른 VLAN 사이를 라우팅으로 연결하는 라우팅

- 스위치 장비의 VLAN 기능 구성 순서
  - 단일 IP 주소 대역을 서브넷 대역으로 분리하고 스위치 장비의 포트에 기반해 각각의 서브넷 대역 설정
  - 서브넷(subnet)
    - 1개의 IP 주소 대역을 2의 배수 단위로 나누는 기법
    - 서브넷 마스크를 통해 네트워크 IP 주소와 호스트 IP 주소의 범위를 알 수 있다
  - 프리픽스 표기법
    - 서브넷 마스크의 1로 표기된 비트의 개수를 이용해 서브넷 상태를 제공하는 표기
    - 예시 : 255.255.0.0 <-> 16
    - n개의 대역으로 나눠지면 프리픽스는 24 + log_2(n)가 된다
  - 대역을 나누면, 프리픽스/서브넷 마스크/네트워크 IP/브로드캐스트 IP 에 변화가 생긴다
  - 스위치 장비에서 각각의 대역에 대해서 스위치 장비의 포트를 나눈다
    - 40개의 호스트와 2개의 영역이 있을 때, 첫번째 영역은 1~20 포트에 연결하고 VLAN 10을 설정하고, 두번째 영

역은 21~40 포트에 연결하고 VLAN 20으로 설정

- WAN(Wide area network)
  - 서로 다른 2개의 LAN 영역 사이에서 발생
    - LAN 영역을 벗어나 라우터 간의 통신이 있는 영역
    - IP 주소에 기반해 내부 통신을 외부와 연결하는 구간
    - 서로 다른 네트워크 ID/ARP 영역을 연결하는 구간
  - 라우팅
    - WAN 영역에서 수행하는 일련의 기능
    - 목적지까지 도달하는 무수한 (라우터 간의) 경로 중 최상의 경로를 선택하는 기능
    - 라우터가 동적인 방식에 따라 라우팅 테이블에 기반해 라우팅을 수행하기 위해서는 라우팅 알고리즘 필요

- 라우팅 알고리즘
  - 거리 계산 알고리즘에 기반해 출발지에서 목적지까지 가장 빠르게 도달할 수 있는 경로를 계산하는 알고리즘
  - 라우팅 알고리즘의 종류
    - 기반 알고리즘과 경로 척도(메트릭)을 통해 구분
    - RIP, BGP, OSPF, ISIS, EIGRP
  - RIP, BGP 방식
    - 벨만-포드 알고리즘(Bellman-Ford Alogirthm)에 기반
    - RIP 방식은 출발지에서 목적지까지 경유하는 라우터의 개수를 기준으로 계산
      i.e 홉 카운트(hop count)
    - BGP 방식은 경로 속성이라는 복합적인 메트릭을 이용해 계산
  - OSPF, ISIS 방식
    - 다익스트라 알고리즘(Dijkstra Alogrithm)에 기반
    - 두 방식은 RIP 방식과 달리 경로 척도가 출발지에서 목적지까지 주어진 대역폭(전송 속도)에 기반하여 계산한 코스트 이용
    - OSPF 방식은 TCP/IP 방식에 적합하도록 설계된 라우팅 알고리즘
    - ISIS 방식은 OSI 방식에 적합하도록 설계한 라우팅 알고리즘
  - EIGRP
    - RIP와 OSPF 방식을 혼합한 라우팅 알고리즘
    - 출발지에서 목적지까지 주어진 대역폭(전송 속도)와 해당 구간에서 실시간 발생하는 지연까지 고려하여 계산한 코스트 이용
    - 최상의 경로와 차선의 경로를 동시에 계산하여 복구 능력 시간을 빠르게 수행

- WAN 영역의 물리 계층 기반 구분
  - 전용 회선 방식
    - 출발지와 목적지 사이에서 회선을 항상 연결 상태로 유지하는 방식
  - 회선 교환 방식
    - 출발지와 목적지 사이에서 통신이 일어날 때만 회선을 연결하는 방식
  - 가상 회선 방식
    - 전용 회선 방식처럼 출발지와 목적지 사이에서 회선을 언제나 연결 상태로 유지하는 방식
    - (BUT!!) 출발지와 목적지 사이에 연결된 회선은 논리적 회선을 이용

[TCP/IP 네트워크 공격 유형]

- TCP/IP 방식의 속성을 이용한 고전적인 공격 유형

- 스캐닝(Scanning) 공격
  - 본격적인 공격에 앞서 수행하는 개념
  - 예시 : 포트 스캔
    - 원격지 호스트를 대상으로 어떤 포트 번호를 사용 중인지 확인하는 기법
    - TCP 헤더의 플래그 항목을 이용해 수행

- 엔맵(Nmap)
  - 가장 많이 사용하는 포트 스캔 도구
  - 엔맵을 이용한 포트 스캔 기법
    - TCP Full Open 스캔
    - TCP Half Open 스캔
    - TCIP FIN 스캔
    - TCP X-max 스캔

- TCP Full Open 스캔
  - 공격자가 해당 스캔을 이용하면 전송 계층에서 특정 포트에 대한 SYN 플래그를 생성해 공격 대상자에게 전송
  - 해당 포트가 사용 중이면 공격 대상자는 ACK&SYN 신호로 응답하고 공격자는 해당 포트가 사용 중임을 확인 가능
  - 해당 포트가 미사용 중이면 공격 대상자는 ACK&RST 신호로 응답하고 공격자는 해당 포트가 미사용 중임을 확인 가능

    - TCP 3단계 연결을 완성하기 때문에 결과는 정확
    - (BUT!!) TCP 3단계 연결을 완성하기 때문에 공격 대상자 측에 포트 스캔 기록이 남을 가능성 존재

- TCP Half Open 스캔
    - TCP Full Open 스캔이 포트 스캔 기록을 남길 가능성을 해결하기 위해 등장한 기법
    - 공격자가 해당 스캔을 이용하면 전송 계층에서 특정 포트에 대한 SYN 플래그를 생성해 공격 대상자에게 전송(like TCP Full Open 스캔)
    - (BUT!!) 해당 포트가 사용 중이면 공격 대상자는 ACK&SYN 신호로 응답하고 공격자는 RST 신호로 응답하여 3단계 연결을 끊음
    - TCP Full Open 스캔과 다르게 어떤 경우에도 3단계 연결을 하지 않음으로써 공격 대상자 측에서 포트 스캔을 알 수 없다
    - 스텔스 포트 스캔(Stealth Port Scan)
        - 공격 대상자에게 포트 스캔을 남기지 않게 수행하는 포트 스캔 기법
        - 예시 : Half Open, FIN, X-mas 스캔
    - 방화벽 입장에서 외부에서 내부로 들어오는 SYN 플래그를 차단하면 TCP 3단계 연결 수행 불가
        - 포트 스캔 수행 불가
        - 공격자는 정상적인 TCP 3 단계 연결 순서를 어긋하네 설정하여 방화벽 우회 가능

- TCP FIN 스캔
    - 방화벽이 외부 SYN 플래그를 차단하여도 FIN 플래그는 허용
    - 공격 대상자가 공격자와 3단계 연결을 설정한 적 없는 상태에서 FIN 신호를 수신받으면 어떤 응답도 회신 불가
        - 공격자는 공격 대상자의 응답이 없으면, 해당 포트를 사용 중이라고 판단
        - 공격 대상자의 해당 포트가 미사용 중이면 공격자는 ACK&RST 응답 수신

- TCP X-mas 스캔
    - TCP FIN 스캔 기법을 응용한 기법
    - 공격자는 전송 계층에서 URG&PSH&FIN 플래그를 생성하여 공격 대상자에게 전송
        - 원래는 모든 플래그를 동시 이용
        - 엔맵은 URG&PSH&FIN 플래그만 사용
    - 방화벽은 외부 SYN 플래그만 차단하기 때문에 공격자의 플래그는 공격 대상자에게 해당 신호 전송 가능

    - TCP FIN 스캔과 마찬가지로 공격 대상자의 응답 여부를 통해 해당 포트의 사용 여부 확인

- TCP Null 스캔
    - TCP X-mas 기법과 정반대의 개념
        - TCP X-mas 스캡은 모든 플래그를 동시에 설정해 방화벽을 우회하는 기법
    - TCP Null 스캔 기법은 어떤 플래그도 설정하지 않고 방화벽을 우회하는 기법

- 스니핑 공격
    - 패킷 분석이라고도 불림
    - 스니퍼를 이용하여 네트워크상의 데이터를 도청하는 행위
        - 스니퍼 : 컴퓨터 네트워크상에 흘러 다니는 트래픽을 엿듣는 도청장치
    - 스니핑이 가능한 이유는 근본적으로 TCP/IP 방식에는 암호화 기능이 없기 때문
        - TCP/IP 방식에 따라 생성한 데이터는 암호 설정이 없는 평문
        - 제3자가 페이로드 내용 확인 가능
    - 이런 공격을 대응하기 위해 VPN(Virtual Private Network) 기법 등장

- 스푸핑 공격
    - 출발지 주소 등을 은폐하거나 변경하는 기법
        - 엄밀하게는 주소 체계 전반을 대상으로 수행하는 공격
    - ARP 스푸핑
        - ARP 캐시 중독 공격이라고도 불림 i.e ARP 캐시 테이블에서 사용하는 주소 체계 조작
        - 관련 계층 : 데이터 링크 계층
        - 공격 내용 : 목적지 맥 주소 조작
    - IP 스푸핑
        - 관련 계층 : 네트워크 계층
        - 공격 내용 : 출발지 IP 주소 조작
    - DNS 스푸핑
        - DNS 캐시 중독 공격이라고도 불림 i.e DNS 캐시 테이블에서 사용하는 주소 체계 조작
        - 관련 계층 : 응용 계층
        - 공격 내용 : 목적지 IP 주소 조작

- 플러딩 공격

    - 출발지 IP 주소를 수시로 변경하여 상대방에게 불필요한 데이터를 계속 전송해 인위적인 부하를 유발하는 기법
    - 일반적으로 IP 스푸핑 기법과 결합해 수행
        - 출발지 IP 주소를 수시로 변경하기 때문
    - DDoS(Distributed Denial of Service) 공격이라고 불림

[TCP/IP 방식의 계층별 취약점에 기반한 공격 유형]

- 물리 계층
    - 물리 계층의 전송 단위는 비트
        - 비트는 전기 신호를 의미
    - 회선 태핑(tapping) 공격
        - 전기 신호를 직접 자신에게 끌어오는 방식
        - 광섬유는 어떤 유형의 신호도 외부로 방출할 수 없게 설계되어 해당 공격 방어
    - 템페스트(tempest) 공격
        - 전송 매체에 흐르는 전기 신호를 검출해 데이터를 해석하는 기법
        - 유선보다는 무선 분야에서 광범위하게 발생
        - Example : 무선 AP 신호를 탐지하는 워드라이빙(war driving) 공격
    - 물리 계층은 기계적/전기적/기능적/절차적 기능을 수행하기 때문에 전기전자 분야 등과 상당히 밀접한 관계

- 데이터 링크 계층
    - 데이터 링크 계층의 전송 단위는 프레임
        - 대표적인 장치는 LAN 카드/스위치/무선 AP
    - 대표적인 공격
        - 맥 플러딩(MAC flooding) 공격
        - ARP 스푸핑(spoofing) 공격
        - VLAN 홉핑(hopping) 공격
    - 맥 플러딩(MAC flooding) 공격
        - 스위치는 포워딩&블로킹으로 동작
        - 플러딩만 동작하는 허브와 스위치가 연동되면, 불필요한 맥 주소들이 스위칭 테이블에 채워지면서 해당 스위치는 플러딩으로 동작
        - 이러한 인위적인 상황을 발생시켜 스위치를 허브처럼 동작하게 하는 기법
        - 스위치 재밍(Switch Jamming) 공격 : 공격자는 자신의 LAN 카드를 무작위 모드로 변환한 뒤 가짜 맥 주소를 무수히 발생시키는 공격

        - 방어 방법 : 스위치 포트마다 사용자의 맥 주소를 "정적"으로 설정함으로써 방어
    - ARP 스푸핑(spoofing) 공격
        - ARP 캐시 테이블에 저장한 대응 관계를 조작하는 공격
        - ARP 캐시 테이블에서 실제 라우터의 맥 주소 대신 공격자의 맥 주소로 변경함으로써 공격 대상자가 전송하는 모든 데이터를 감지
        - 공격자는 공격 대상자와 라우터 사이의 통신을 중계
        - 방어 방법 : 운영체제에서 IP 주소와 맥 주소를 "정적"으로 설정함으로써 방어
        - 응용 계층에서 수행하는 DNS/SSL/쿠키 스푸핑이나 각종 스니핑 공격을 수행하기 위한 기반 공격
          => LAN 보안에서 가장 큰 비중 차지
    - VLAN 홉핑(hopping) 공격
        - 자신과 다른 VLAN 영역으로 넘어가는 기법
        - 이더넷 프레임 헤더 형식
            - 목적지 주소|출발지 주소|VLAN 식별자 정보|타입
        - 홉핑 공격 시 이더넷 프레임 헤더 형식
            - 목적지 주소|출발지 주소|VLAN 식별자 정보1|VLAN 식별자 정보2|타입
        - VLAN 식별자 두 개를 헤더에 붙임을써 스위치 운영체제를 속인다
    - DHCP 고갈 공격
        - DHCP 서버는 LAN 영역에 존재
        - 가짜 맥 주소를 브로드캐스트 방식으로 생성하여 DHCP 서버의 맥 주소에 해당하는 IP 주소가 계속 공격자에게 할당
        - DHCP 서버가 확보한 IP 주소가 소진되면서 IP 할당 불가능
        - DHCP 서버를 대상으로 수행하는 일종의 플러딩 공격에 해당
    - DHCP 스푸핑 공격
        - DHCP 서버의 IP 주소 고갈 상태를 이용하여 공격자가 DHCP 서버로 위장 가능
        - 공격자가 사용자들에게 조작된 게이트웨이/IP주소 할당 가능
        - 사용자들의 모든 데이터가 공격자에게 흘러 들어오게 한다

- 네트워크 계층
    - 네트워크 계층의 전송 단위는 패킷
        - 대표적인 장치는 라우터
    - 대표적인 공격
        - 랜드(LAND) 공격
        - ICMP 플러딩 공격

        - ICMP 스머핑 공격
    - 랜드(LAND) 공격
        - IP 스푸핑 공격을 변형한 기법
        - 출발지와 목적지 IP 주소를 동일하게 설정하여 공격 대상자에게 인위적인 과부하를 유발하는 네트워크 계층의 플러딩 공격
        - 출발지와 목적지 IP 주소를 동일하기 때문에 ICMP 요청 패킷 등을 전송하면 공격 대상자는 응답패킷을 자신에게 보내는 무한 루프 발생
        - 방어 방법: 방화벽 등에서 출발지와 목적지 IP 주소가 동일한 패킷을 차단
    - ICMP 플러딩 공격
        - 죽음의 핑 공격이라고도 불림
        - 보통 ICMP 페이로드 크기를 65000 byte 이상으로 설정하고 IP 스푸핑 공격을 적용해 출발지 IP 주소를 매 순간 변경하여 전송
        - 수신 측은 매번 분할 패킷을 재조립하여 ICMP 응답 패킷을 전송하기 때문에 과부하 발생
    - ICMP 스머핑 공격
        - ICMP 플러딩 공격의 변형 기법
        - 공격 대상자의 IP 주소를 출발지 IP 주소로 설장하고 목적지 IP 주소를 브로드캐스트 IP 주소로 설정
        - 공격 대장자는 공격자가 보낸 패킷을 자신과 동일한 대역에 있는 호스트에게 전송하고 응답 패킷을 수신받으면서 과부하 발생
        - 방어 방법 : 브로드캐스트 IP 주소의 비활성화
    - 티얼 드롭(Teardrop)
        - 패킷 분할 속성을 악용한 기법
        - 플러딩 공격의 일종
        - 분할 패킷의 순서 정보를 조작함으로써 운영체제가 존재하지 않는 프래그먼트 오프셋 정보를 검색하면서 과부하 발생
        - 방어 방법 : 프래그먼트 오프셋 정보가 일정 시간 이상 불일치 한 경우 전체 패킷을 폐기

- 전송 계층
    - 전송 계층의 전송 단위는 세그먼트/데이터그램
    - 대표적인 공격
        - TCP SYN 공격
        - 본크-보인크(Bonk-Boink) 공격
    - TCP SYN 공격
        - TCP 3단계 연결 속성을 악용한 공격
        - 네트워크 계층의 ICMP 플러딩 공격과 유사
        - 매 순간 공격 대상자에게 SYN 플래그를 전송함으로써 공격 대상자의 과부하 상태 유발

        - 최근에는 악성 코드를 이용한 좀비 시스템으로 하여금 공격자가 설장한 목적지에 SYN 플래그를 전송하게 함으로써 출발지 IP 주소 기반 차단이 불가능
        - 방어 방법 : 최근 방화벽은 임계치 설정을 통해 TCP SYN 공격 차단
    - 본크-보인크(Bonk-Boink) 공격
        - 네트워크 계층의 티얼드롭 공격과 유사
        - TCP 헤더 중 일련번호 항목을 조작해 수신 측에서 정상적인 재조립이 불가능하게 함으로써 과부하를 유발
        - 방어 방법 : 수신 받은 일련번호가 불일치하면 세그먼트 전체를 폐기

- 응용 계층
    - 응용 계층에는 무수한 공격 존재
    - 웹 분야에서의 플러딩 공격
        - HTTP 페이로드는 헤더와 바디로 구성
        - 헤더 : 서버의 바디 처리 방식 등에 대한 제어 정보
        - 바디 : HTML 코드나 사용자의 계정 정보
        - 수신자 서버는 헤더를 먼저 수신하고, 헤더에 설정한 정보에 따라 바디 처리
        - 헤더와 바디는 캐리지 값(\r\n\r\n)으로 구분 i.e 캐리지 값은 헤더와 바디의 경계를 표시하는 구분자
        - 헤더의 기능을 조작하면 다양한 형태의 플러딩 공격 가능
    - HTTP GET 플러딩 공격
        - F5 리로드 공격이라고도 불림
        - 특정 사이트에 F5 키를 누르면 웹 브러우저는 GET 지시자를 이용해 웹 서버에게 지속적으로 기본 페이지 요청
        - 해당 요청을 반복적으로 수행하면 웹 서버 측에 과부하 발생
        - 방어 방법 : 방화벽에서의 동시 접속 제한 설정 외에는 적절한 방법이 없음
    - HTTP GET 캐시 제어 플러딩 공격 or CC 공격
        - 일반 헤더의 캐시 설정 부분을 조작해 캐싱 서버가 아닌 서버에게 직접 처리를 요청하여 서버의 과부하 유발
        - 방어 방법 : 방화벽에서의 동시 접속 제한 설정을 통해 방어
    - 슬로우 HTTP 헤더 공격 or 슬로우리스 공격
        - 서버는 헤더 정보를 완전히 수실할 때까지 연결을 유지
        - 헤더와 바디의 구분자를 애매하게 설정함으로써 서버의 연결 유지로 인한 과부하를 유발하는 기법
        - 방어 방법 : 방화벽의 연결 타임아웃 설정, 응용 계층 기반의 방화변을 통해 조작 헤더 유입 차단
    - 슬로우 HTTP 포스트 공격 or 러디(Rudy) 공격
        - 서버로 대량의 데이터를 전송할 때 장시간 동안 분할 전송하는 기법

        - 웹 서버 측에서는 슬로우리스 공격처럼 해당 데이터를 완전히 수신하기 위해 연결을 유지하면서 과부하 발생
        - 방어 방법 : 방화벽의 타임아웃 설정

[보안 알고리즘]

- 사이버 보안의 구성
    - 전제 : 송신자와 수신자는 서로 만날 수 없다
    - 기밀성 : 서로 주고받은 정보에 대한 비밀성 보장
        - 사이버 보안의 기본이자 중심
    - 무결성 : 서로 주고받은 실제 정보에 대한 정확성 보장
    - 가용성 : 정당한 사용자가 필요할 때마다 즉각적으로 정보에 접근하여 사용
    - 인증 : 송진자와 수신자 사이의 확신성을 보장
    - 부인 봉쇄 : 수신자가 정보를 받았는데 송신자가 이를 부인하는 일 등을 방지

- 기밀성 보안 알고리즘
    - 기밀성(Confidentiality)
        - 송신자와 수신자가 주고받는 데이터를 대상으로 비밀성을 보장하는 개념
        - 송신자와 수신자가 평문이 아닌 암호문을 통해 상호 간의 비밀 통신을 보장 하는 개념
    - 평문(Plaintext)
        - 누구나 읽을 수 있거나 접근할 수 있는 정보
    - 암호문(Cyphertext)
        - 누구나 읽을 수 없거나 접근할 수 없는 정보
    - 열쇠(Key)
        - 사이버 암호 체계에서의 암호 해독문
    - 송신자와 수신자가 사용하는 열쇠 방식에 따른 구조
        - 대칭 암호 구조(Symmetric Key Algorithm)
            - 송신자와 수신자가 사용하는 열쇠가 동일
            - 비밀 열쇠(secrete key) : 동일하게 사용하는 열쇠
        - 비대칭 암호 구조(Asymmetric Key Algorithm)
            - 송신자와 수신자가 사용하는 열쇠가 다름
            - 공개 열쇠(public key) : 송신자가 사용하는 열쇠
            - 개인 열쇠(private key) : 수신자가 사용하는 열쇠
    - 암호화(Encryption)
        - 평문을 암호문으로 변경하는 개념
        - 송신자가 주체

    - 복호화(Decryption)
        - 암호문을 평문으로 다시 변경하는 개념
        - 수신자가 주체
    - 비밀 열쇠의 배분 문제
        - 대칭 암호 구조에서 송신자와 수신자가 암호 통신 전 어떻기 비밀 열쇠를 교환할 것인지에 대한 문제
        - 송신자가 수신자에게 직접 비밀 열쇠를 전송하면 제3자가 탈취할 수 있는 위험성 존재
        - DH 알고리즘을 이용하여 해결

- DH 알고리즘(Diffie-Hellman-Public key Exchange Alogorithm)
    - 이산 대수 속성에 따라 공개 열쇠와 개인 열쇠라는 2개의 열쇠 개념 도입
    - DH 알고리즘 순서
        - 송신자와 수신자는 각각 공개 열쇠와 개인 열쇠 생성
        - 송신자와 수신자는 서로 공개 열쇠 교환
        - 각자의 개인 열쇠와 상대방에게 받은 공개 열쇠를 끼워서 비밀 열쇠 생성
    - 공개 열쇠는 비밀 열쇠의 부분이기 때문에 외부에 공개 되어도 상관 없음
    - 비밀 열쇠를 이용하는 대표적인 대칭 암호 구조
        - DES(Data Encryption Standard) 방식
        - AES(Advancde Encryption Standard) 방식
    - DES/AES 방식을 사용하기 전에는 DH 알고리즘에 따라 서로 공개 열쇠를 주고받아 비밀 열쇠를 생성하는 과정 필요
    - 문제점
        - 공개 열쇠와 개인 열쇠는 오직 비밀 열쇠를 생성하는 데만 사용
        - 공개/개인 열쇠 자체를 직접 복호화/암호화 불가능

- RSA 알고리즘
    - DH 알고리즘과 달리 공개 열쇠와 개인 열쇠를 직접 암호화/복호화하는데 사용하는 비대칭 암호 구조
    - RSA 알고리즘 순서
        - 송신자가 수신자에게 공개 열쇠를 요청
        - 수신자는 송신자에게 자신의 공개 열쇠 전송
        - 송신자는 수신자의 공개 열쇠를 이용하여 평문을 암호화하여 수신자에게 전송
        - 수신자는 수신받은 암호문을 자신의 개인 열쇠를 이용하여 복호화
    - 문제점 - 공개 열쇠의 신뢰 문제
        - 수신자가 보낸 공개 열쇠가 진짜인지 가짜인지 보장 못함
        - 해당 문제는 PKI 구조로 해결

- PKI(Public Key Infrastructure) 구조
  - 비대칭 구조에 기반한 암호 방식을 광범위하게 활용하기 위한 기술적/조직적/법률적 트리 형태의 기반 시설
  - 공인 인증 기관, 등록 기관, 디렉토리 서비스 서버 등으로 구성
  - 사용자는 자신의 공개 열쇠를 공인 인증 기관에 등록하는 방식
  - 공인 인증 기관에서는 각 사용자의 공개 열쇠를 DB에 등록하고, 고유한 일련번호를 공개 열쇠에 부여(공인 인증서)
    - 송신자가 수신자에게 공개 열쇠를 받으면, 공인 인증 기관에 해당 공개 열쇠를 의뢰
    - 공개 열쇠의 일련번호가 인증 기관에서 발행한 일련번호와 일치하면 송신자는 해당 공개 열쇠 신뢰 가능

- 하이브리드 암호 방식
  - 비대칭 암호 구조는 처리 속도의 문제 존재
  - 대칭 암호 구조와 비대칭 암호 구조를 혼합하는 방식
  - SSH VPN 기법
    - 계정/비밀번호처럼 가벼운 인증 정보는 RSA 알고리즘 등을 사용
    - 실제 본문처럼 무거운 정보는 AES 알고리즘 등을 사용
  - SSL/TSL VPN 기법
    - 웹 보안 구혈할 때 주로 사용
    - 송신자는 임의의 비밀 열쇠를 생성하여 전송 데이터를 암호화한 뒤 수신자에게 전송
    - 수신자는 복호화를 위해 자신의 공개 열쇠를 송신자에게 전송
    - 송신자는 수신자의 공개 열쇠를 이용하여 임의의 비밀 열쇠를 암호화(called 전자 봉투)
    - 송신자는 해당 전자 봉투를 수신자에게 전송
    - 수신자는 자신의 개인 열쇠를 이용하여 전자 봉투를 복호화 => 송신자의 비밀 열쇠 획득
    - 송신자의 비밀 열쇠를 이용하여 암호문을 복호화
    - 세션 키(session key)
      - 송신자가 난수를 이용하여 임의로 생성한 비밀 열쇠
      - 송신자와 수신자 사이에서 일회성 열쇠로 사용하기 위한 용도

- 무결성 보안 알고리즘
  - 무결성(Integrity)
    - 송신자와 수신자가 주고받는 데이터를 대상으로 정확성을 보장하는 개념

    - 알고리즘 종류
        - 요약 함수(Hash Function)
        - 전자 서명(Digital Signature)

- 요약 함수(Hash Function)
    - 대칭 암호 구조와 비대칭 암호 구조 모두에서 무결성을 구현하기 위해 사용하는 알고리즘
    - 가변적인 길이의 원본을 고정적인 길이의 요약본으로 처리하는 일종의 메시지 무결성 코드
        - 원본 : 요약 함수 처리 이전 상태
        - 요약본 : 요약 함수 처리 이후 상태
    - 특징 1 - 일방향성/역상저항성
        - 요약본을 다시 원본으로 복원 불가능
    - 특징 2 - 충돌 저항성
        - 충돌 : 서로 다른 2개의 원본에서 같은 요약본이 나온 상황
        - 요약본에 대한 원본은 항상 unique
    - 요약 함수는 역상저항성 및 출동 저항성을 통해 데이터 정확성을 검증
    - 요약함수를 통한 무결성 검증 과정
        - 송신자는 원본 데이터에 원본에 대한 요약본을 첨부하여 수신자에게 전송
        - 수신자는 같은 방식의 요약 함수를 통해 수신받은 원본을 요약본으로 만든다
        - 수신자가 만든 요약본과 첨부받은 송신자의 요약본을 비교하여 원본의 정확성을 판단
    - Example
        - 운영체제의 비밀번호 저장시 사용
    - HMAC(Keyed-Hashing for Message Authentication)
        - 요약 함수를 이용해 원본에 비밀 열쇠를 추가하여 요약본을 생성
        - 무결성을 구현하기 위한 요약 함수를 인증 기능까지 확장하여 사용하는 기법
        - 비밀 열쇠를 사용하는 환경에서 구현

- 전자 서명(Digital Signature)
    - 비대칭 암호 구조에서만 사용 가능
    - 전자 서명 구현시 RSA 알고리즘 사용 가능
        - 이 경우, 공개/개인 열쇠의 용도는 기밀성에서 사용할 때와 정반대
        - 전자 서명의 암호화 : 송신자의 개인 열쇠
        - 전자 서명의 복호화 : 수신자의 공개 열쇠
    - 예시 상황
        - 송신자가 전자 우편을 작성하고 자필 서명을 했다고 가정

        - 이때, 전자 우편은 본문 내용과 자필 서명 부분으로 구분
        - 서명의 이유는 작성자가 본인임을 상대방에게 알리기 위한 부가적인 정보
    - 열쇠의 사용처
        - 수신자의 공개 열쇠 : 전자 서명 부분의 복호화
        - 수신자의 개인 열쇠 : 전자 서명 부분의 암호화
        - 송신자의 공개 열쇠 : 본문의 암호화
        - 송신자의 개인 열쇠 : 본문의 복호화
    - 전자 서명 순서
        - 송신자는 수신자의 공개 열쇠를 이용해 본문 내용을 암호화하고, 자필 서명은 자신의 개인 열쇠로 암호화
        - 수신자는 전자 우편의 본문 내용을 자신의 개인 열쇠로 복호화하고, 자필 서명 부분을 송신자의 공개 열쇠로 복호화
        - 해당 과정을 통해 복호화가 성공하면 본문 내용과 송신자에 대한 인증이 동시에 해결
          (why?) 공개 열쇠와 개인 열쇠는 한 쌍이기 때문

[VPN 개념]

- VPN(Virtual Private Network)
    - 기밀성
        - 서로 주고 받은 실제 정보에 대한 비밀성을 보장하는 개념
    - VPN은 기밀성을 구현하기 위해 수행하는 일련의 암호화 기법
    - TCP/IP 방식에서의 데이터 전송
        - 통신을 구현하기 위해 설계되어 있어서 모든 페이로드는 헤더와 평문 구조
        - 스니핑(중간자 개입 공격)에 매우 취약
        - 스니핑 공격과 같은 기밀성 위협 요소에 대응하기 위해서는 VPN 기법이 반드시 필요
    - VPN 기법은 TCP/IP 방식의 데이터 전송 단위를 대상으로 암호 알고리즘을 적용한 범위에 따라 구분
        - 응용 계층 기반의 VPN 기법
        - 네트워크 계층 기반의 VPN 기법
        - 데이터 링크 계층 기반의 VPN 기법

- 응용 계층 기반의 VPN 기법
    - 페이로드 영역만 암호화
        - 암호화 페이로드 | 평문 헤더1 | 평문 헤더2 | 평문 헤더3

        - 해당 데이터의 페이로드는 읽을 수 없음
        - 모든 헤더에 대한 정보는 읽기 가능
    - Example
        - SSH VPN 기법
        - SSL/TLS VPN 기법(전송 계층으로 분류하는 경우도 있음)
        - PGP VPN 기법

- SSH VPN 기법
    - DH 알고리즘(대칭)과 RSA 알고리즘(비대칭) 등을 사용
    - 계정/비밀번호 등에 대해서는 RSA 알고리즘 이용
    - 인증 이후의 작업 내용은 AES 알고리즘 이용

- SSL/TLS VPN 기법
    - 송신자의 비밀 열쇠(수신자가 복호화 할 때 사용, 세션 키)를 수신자의 공개 열쇠로 암호화하여 수신자에게 전송하는 방식
    - 응용 계층과 전송 계층 사이에서 동작
    - 핸드셰이크 프로토콜 | 암호 변경 사양 프로토콜 | 경고 프로토콜 | 레코드 프로토콜 구조
        - 핸드셰이크 프로토콜 : DES 알고리즘 등에 기반한 세션 키 생성
        - 레코드 프로토콜 : 세켠 키를 상대방(수신자)의 공개키로 암호화
        - 암호 변경 사양 프로토콜 : 상호 암호 통신을 수행하기 위해 필요한 일련의 보안 알고리즘을 사전에 협의하기 위한 보안 매개 변수 생성
        - 경고 프로토콜 : 오류 발생 시 상대방(수신자)에게 오류 통보 기능 수행
    - 동작 과정
        - 초기 협상 단계 : 클라이언트와 서버 사이에 클라이언트 헬로/서버 헬로 신호 교환
        - 서버 인증 단계 : 서버에서 공개 열쇠를 클라이언트에게 전송
        - 클라이언트 인증 단계 : 클라이언트 헨드셰이크 프로토콜에서 생성한 임시 비밀 열쇠(세션)을 서버의 공개 열쇠로 암호화하여 전송 AND 다음 단계부터 사용할 보안 매개 변수를 서버로 전송
        - 종료 단계 : 일련의 통신 이후 TCP 방식에 따라 순차적으로 연결 종료

- 네트워크 계층 기반의 VPN 기법
    - 페이로드 뿐만 아니라 세그먼트/데이터그램 헤더와 패킷 헤더를 암호화

    - 대표적인 기법 : IPSec VPN 기법

- SA(Security Association)
    - 일반 IP 패킷과 다르게 SA 패킷은 상호 간 수행할 일련의 보안 정책 집합체 i.e 보안 매개 변수가 존재
        - 보안 매개 변수 : 상호 암호 통신을 수행하는 데 필요한 일련의 보안 알고리즘 정보를 사전에 협의하기 위한 내용
    - SA 패킷을 전송 하기 위해서는 IKE/ISAKMP 프로토콜 필요
        - SA 패킷이 통조림이면, IKE/ISAKMP는 통조림을 이동하게 해주는 컨베이어 벨트
        - IKE(Internet Key Exchange)
        - ISAKMP(Internet Security Association and Key Management Protocol)

- IPSec VPN
    - 송신자와 수신자는 IPSec VPN 통신 수행 전 IKE(ISAKMP) 프로토콜을 이용하여 SA 패킷을 주고받으며, 보안 설정에 대한 일련의 내용(보안 매개 변수)를 교환
    - IKE 1단계 : 메인 모드(Main Mode)
        - 송신자와 수신자가 SA 패킷을 통해 상호 간에 인증하는 과정 의미
    - IKE 2단계 : 퀵 모드(Quick Mode)
        - 보안 매개 변수를 주고받은 다음 IPSec VPN 종류 등을 결정하기 위한 협상 과정 의미
    - IP Sec VPN 종류
        - AH 방식
        - ESP 방식
    - AH(Authentication Header) 방식
        - 무결성과 인증 기능만 부여
    - ESP(Encapsulating Security Payload)
        - 무결성과 인증 및 기밀성 기능 부여
        - IPSec VPN 기법을 구성 할때 암호문 패킷 기반의 ESP 방식을 주로 이용
    - IPSec VPN 기법은 터널 구간의 차이와 ESP 헤더 삽입 위치의 차이에 따라 구분
        - 터널 구간 : 암호문이 통과하는 영역
        - 전송 모드 / 터널 모드

- IPSec VPN 종류 1 :전송 모드
    - 암호화와 복호화를 송신자 호스트와 수신자 호스트에서 수행
        - i.e 송신자/수신자 호스트 사이에 터널 구간을 형성
        - 해당 LAN 영역에 설치한 VPN 장비가 아닌, 실제 사용자 호스트 사이에서 암호화/복호화 수행

=> LAN 영역의 공격자로부터 스니핑 공격을 가장 최소화
    - 암호문 페이로드 | 암호문 TCP/UDP 헤더 | ESP 헤더 | 평문 IP 헤더
        - TCP/UDP 헤더와 IP 헤더 사이에 ESP 헤더가 들어감
        - TCP/UDP 헤더와 페이로드는 암호화되어 있어 스니핑 공격이 있어도 ESP 헤더까지만 노출
        - 라우터가 라우팅을 수행하기 위해 IP 헤더는 평문을 유지
    - 장점
        - 스니핑 공격의 최소화
    - 단점
        - 사용자가 IPSec VPN 기법을 설정해야 하는 어려움 존재
        - 단점의 해결 방안으로 터널 모드

- IPSec VPN 종류 2 :터널 모드
    - 전송 모드의 단점(사용자의 번거로움)을 해결하는 방법
        - 실제 IPSec VPN에서 사용하는 모드
    - 송신 측 VPN 장비와 수신측 VPN 장비에서 암호화와 복호화 진행
        - 사용자는 VPN 이론을 몰라도 암호 통신 수행 가능
        - (why?) 터널 구간이 송신/수신 VPN 장비 사이에 있기 때문
        - (but!) 동일한 LAN 영역에서 일어나는 스니핑 공격에는 취약
    - 암호문 페이로드 | 암호문 TCP/UDP 헤더 | 암호문 IP 헤더 | ESP 헤더 | 평문 IP 헤더
        - ESP 헤더 뒤에 나오는 IP 헤더, TCP/UDP 헤더, 페이로드는 암호화되어 있어 스니핑 공격이 있어도 노출 X
        - 실제 IP 주소는 암호화되어 있어 송신자&수신자의 실제 IP 은폐 가능
        - 외부 라우터의 원할한 라우팅 기능을 위해 평문 IP 헤더가 존재
            - 해당 평문 IP 헤더는 송신/수신 VPN 장치의 IP 주소
            - 이 두개는 실제 송신/수신 호스트의 IP와 다름

[보안 장비에 대한 이해]

- 방화벽(Firewall)
    - 외부망과 내부망 사이에서 미리 설정한 규칙에 따라 특정한 패킷을 차단하거나 허용하는 소프트웨어 설정 or 하드웨어 장비

    - 보안 정책상 예방 통제를 구현하기 위한 대표적인 장치
        - 예방 통제 : 악성 코드 등을 사전에 차단함으로써 내부 전산 자원을 보호하겠다는 개념
    - 방화벽의 구분
        - ACL(Access Control List) : 네트워크, 전송 계층 기반
        - ALG(Application Level Gateway) : 응용 계층 기반

- ACL(Access Control List)
    - 패킷 필터링(Packet Filtering) 방식
    - 프록시 방화벽(웹)이라고도 불림
    - 출발지 IP주소에 기반한 표준 ACL 방식과 출발지/목적지 IP 포트 번호에 기반한 확장 ACL 방식으로 구분
    - Example : SQL 삽입 공격 등을 차단하는 웹 방화벽

- 상태 기반 감시(Stateful Inspection)
    - 방화벽의 기능
    - 내부에서 외부로 나갔다 되돌아오는 리턴 패킷 여부 추적
    - 외부로 나갔다가 다시 내부로 돌아오는 ICMP 요청 및 응답에 대해서는 차단을 하지 않기 위한 기능
    - TCP의 3단계 연결 설정 또한 외부<->내부 통신이기에 해당 기능이 없다면 연결 설정이 막힘
    - 상태 흐름 테이블
        - 상태 기반 감시 기능을 위해 내부에서 외부로 나간 패킷 등을 기록하는 테이블
        - 상태 흐름 테이블에 없는 패킷은 차단하고 존재하는 패킷은 허용함으로써 리턴 패킷 관리
        - 타임아웃 기능을 이용하여 내부->외부 패킷에 대해서 응답이 없으면 외부에서 들어온 해당 패킷은 차단하는 방식을 이용

- ACL 방식의 차별된 기능
    - 동적 ACL 기법
        - 원격 접속자가 해당 라우터에 텔넷 접속을 통해 인증을 성공하면 내부 자원에 접근을 허용하는 기법
    - 반사 ACL 기법
        - ACL 방식으로 구현한 일종의 상태 기반 감시 기능
        - 외부로부터 들어오는 접속은 차단
        - 내부 사용자에 의한 응답으로 돌아오는 접속은 허용
    - 시간 기반 ACL 기법
        - 일정한 시간 동안만 접속을 허용하는 기법

- IPTables
  - 리눅스의 기본 방화벽
  - 3&4 계층 기반의 방화벽으로 상태 기반 감시 기능 뿐만 아니라 NAT 기능 제공
  - 체인(Chain)
    - 인풋 체인 : 방화벽을 목적지로 설정해 이동하는 경로
    - 아웃풋 체인 : 방화벽을 출발지로 설정해 이동하는 경로
    - 포워드 체인 : 방화벽을 통과하거나 경유해 이동하는 경로

- 침입 탐지 장비 + 침입 방지 장비
  - 바이러스 백신이 수행하는 탐지와 차단 기능을 분리해서 구현한 장치
  - 침입 탐지 장비(Intrusion detection system)
    - 일정한 탐지 규칙에 따라 기존의 공격 유형을 탐지하면 정보를 안전한 공간으로 전환 and 관리자에게 해당 내용 통보 and 공격자에게 경고
    - 방화벽과 달리 접근 권한 제어/인증 기능이 없는 소프트웨어/하드웨어 장비
    - (Note) 바이러스 백신의 구동 원리 : 기존의 악성 코드 견본을 수집한 뒤 자체 데이터베이스에 저장하여 악성 코드 여부 판단
    - 바이러스 백신처럼 동작하고, 악성 코드 유형을 저장한 엔진의 주기적 업데이트가 필수
  - 침입 방지 탐지(Intrusion protection system)
    - 오탐, 미탐 같은 탐지 오류 증상을 주의
    - 오탐(False Positive) : 정상적인 유형을 악의적인 유형으로 오판
    - 미탐(False Negative) : 악의적인 유형을 정상적인 유형으로 오판
    - 일시적인 접속 폭주를 DDoS 공격으로 판단하는 것은 오탐
    - 새로운 악성 코드가 통과하는 것은 미탐

- 악성 코드 탐지 방법
  - 서명 기반 탐지 기법
    - 악성 코드를 탐지하는 가장 일반적인 방법
    - 악성 코드 유형을 사전에 등록하여 탐지 수행
    - 정확한 탐지 가능
    - 새로운 악성 코드에 대해서는 탐지를 못하는 약점
  - 정책 기반 탐지 기법
  - 이상 기반 탐지 기법
  - 유인 기반 탐지 기법