# HTTP

HTTP : Hypertext Transfer Protocol 은 ‘분산된 하이퍼미디어 정보 분산 애플리케이션 distributed hypermedia information distribution application’으로 불린다. HTTP는 누군가 인터넷에서 브라우징할 때 사용되는 애플리케이션이다. HTTP는 요청/응답 모델을 사용한다.

# 일반적인 HTTP 통신 분석

일반적인 HTTP 통신은 요청/응답 통신 방식을 사용한다. 클라이언트는 HTTP 서버 요청을 만들고, 서버는 상태 코드로 응답한다.

HTTP 통신은 80번 포트 연결 확립을 위해 TCP 3방향 핸드셰이킹으로 시작한다. 물론 HTTP 통신은 다른 포트를 사용할 수 있다.

TCP 연결이 성공적으로 확립된 후 클라이언트는 ‘/’를 위한 HTTP GET 요청을 생성한다. 서버는 상태 코드 Status Code 200 OK로 응답하고 메인 페이지의 내용을 클라이언트로 전송한다.

# HTTP 패킷 구조 분석

HTTP 패킷은 가변 길이이다.

# HTTP 메소드

http 메소드는 HTTP 명령어로도 불리며, 메소드는 HTTP 패킷의 목적을 정의한다.

* GET : URI(Uniform Resource Indicator) 필드에 의해 정의된 검색 정보(Retrieves information)
* HEAD : 원하는 URI와 관련된 메타데이터 검색
* POST : 데이터를 HTTP 서버로 전송
* OPTIONS : 리소스와 관련된 옵션을 결정
* PUT : 데이터를 HTTP 서버로 전송
* DELETE : URI에 의해 정의된 리소스 삭제
* TRACE : 클라이언트는 서버가 클라이언트로부터 무엇을 수신했는지 볼 수 있게 원격 루프백을 적용한다. 이는 크로스사이트 추적 취약점에 대비하기 위해 많은 회사가 비활성화했기 때문에 드물게 확인할 수 있다.
* CONNECT : 프록시 디바이스에 연결

# 호스트

호스트 헤더 필드는 모든 HTTP/1.1 요청 메시지에서 요구된다. 호스트 필드는 요청되는 리소스의 인터넷 호스트와 포트 변호를 식별한다. 포트 변호가 정의되지 않으면 서비스에 대한 디폴트 포트 번호가 정의된다(예를 들면 HTTP 서비스는 80번 포트)，

# 요청 수식자

HTTP 요청과 응답은 요청에 대한 세부 사항을 제공하기 위해 요청 수식자 Request Modifier를 사용한다.

* Accept : 허용 가능한 컨텐츠 유형
* Accept-Charset : 허용 가능한 문자 집합
* Accept-Encoding : 허용 가능한 인코딩
* Accept-Language : 허용 가능한 언어
* Accept-Ranges : 서버는 범위 요청을 허용할 수 있음
* Authorization : HTTP 인증을 위한 인증서
* Cache-Control : 지시문 캐시
* Connection : 사용자 에이전트에 의해 선호하는 연결 유형
* Cookie : HTTP 쿠키
* Content-Length : 요청 바디의 길이(바이트)
* Content-Type : 본문의 MIME 유형(POST와 PUT 요청에 사용되는)
* Date : 날짜와 시간 메시지 전송
* Expect : 클라이언트에 의해 예상된 서버 동작을 규정
* If-Match : 클라이언트가 제공한 정보와 일치하면 동작을 수행
* If-Modified-Since : 캐시 데이터의 날짜/시간을 제공. 현재라면 304 Not Modified
* If-Range : 부족한 정보 범위를 요청
* If-Unmodified-Since : 특정 날짜/시간 이후에 수정되지 않는 경우에만 전송
* Max-Forwards : 프록시나 게이트웨이를 통한 전달 번호를 제한
* Proxy-Authorization : 프록시 연결에 대한 인증서
* Range : 개체의 일부분만 요청
* Referrer : 이전 웹사이트 주소를 현재로 연결
* TE : 전송 인코딩 허용
* User-Agent : 사용자 에이전트, 일반적으로 브라우저나 운영체제
* Via : 프록시 통과