HTTP

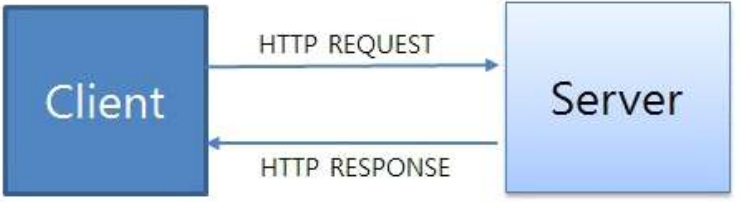
클라이언트와 서버 간 통신을 위한 통신 규칙

HTTP는 인터넷에서 데이터를 주고받기 위한 프로토콜 중 하나로 TCP/IP를 이용

데이터를 암호화하지 않고 평문으로 전송

보안 기능이 없기 때문에 데이터가 노출될 수 있음

HTTP 동작원리



HTTP 동작원리

- 사용자가 브라우저를 통해서 어떠한 요청(request)을 하면 서버에서는 해당 요청사항에 맞는 결과를 찾아서 사용자에게 응답(response)하는 형태로 동작

- HTTP 프로토콜은 주로 웹 브라우저와 웹 서버 간의 통신에 사용되며 TCP/IP 통신 위에서 동작

- 기본 포트는 80번.

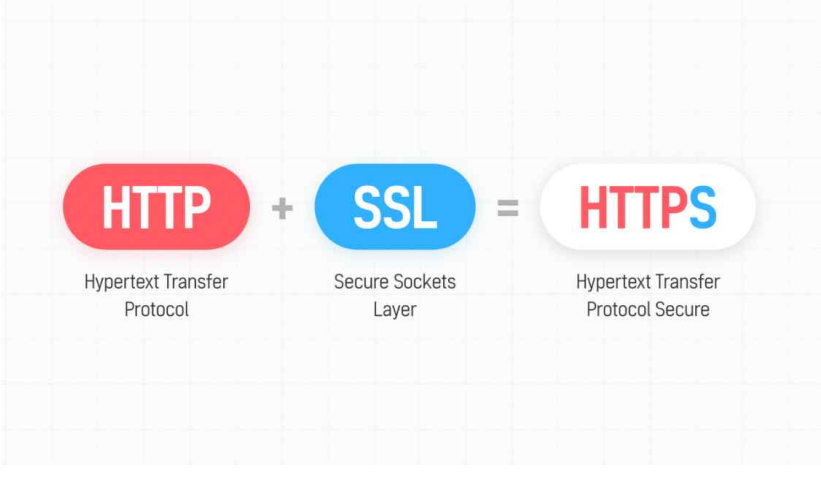
Status Code (상태 코드)

- 1XX (조건부 응답) : 요청을 받았으며 작업을 계속한다.
- 2XX (성공) : 클라이언트가 요청한 동작을 수신하여 이해했고 승낙했으며 성공적으로 처리했음을 가리킨다.
- 3XX (리다이렉션 완료) : 클라이언트는 요청을 마치기 위해 추가 동작을 취해야 한다.
- 4XX (요청 오류) : 클라이언트에 오류가 있음을 나타낸다.
- 5XX (서버 오류) : 서버가 유효한 요청을 명백하게 수행하지 못했음을 나타낸다.

HTTP Request Methods

- GET : 리소스 조회
- POST: 요청 데이터 처리, 주로 등록에 사용
- PUT : 리소스를 대체(덮어쓰기), 해당 리소스가 없으면 생성
- DELETE : 리소스 삭제

HTTPS


HTTP
+
SSL
=
HTTPS
Hypertext Transfer Protocol
Secure Sockets Layer
Hypertext Transfer Protocol Secure

HTTPS

- HTTP에서 보안 레이어가 추가된 프로토콜
- HTTPS에서는 브라우저와 서버가 데이터를 전송하기 전에 암호화된 연결을 사용
- TCP 위에 SSL/TLS 층을 추가하여 암호화, 인증 그리고 무결성 보장

SSL/TLS


HTTP
TCP
IP
네트워크 인터페이스
HTTP


HTTP
SSL or TLS
TCP
IP
네트워크 인터페이스
HTTPS

SSL/TLS

- 전송 계층에서 보안을 제공하는 포로토콜로 보안 소켓 계층(SSL/TLS)이라고 한다
- SSL은 1.0부터 시작해서 TLS로 명칭이 변경되었으나 보통 이를 합쳐 SSL/TLS로 많이 부른다

- 클라이언트와 서버가 통신을 할 때 보안 소켓 계층(SSL/TLS)을 통해 제3자가 간섭하지 못 하도록 막는 역할

- 보안 소켓 계층(SSL/TLS)은 보안 세션을 기반으로 데이터를 암호화하며 보안 세션이 만들어질 때 인증 메커니즘, 키 교환 암호화 알고리즘, 해싱 알고리즘이 사용된다.

SSL 프로세싱

- SSL 인증서 관련 프로세스의 대표적인 두 가지 암호화 방식 대칭키와 공개키 방식

대칭키 암호화 방식

- 대칭키 암호화 방식

- 암호화를 하는 키와 복호화를 하는 키가 동일한 방식
- 하나의 암호화키(key)로 평문을 암호화하고, 다시 암호문을 원래의 평문으로 복호화하는 방식
- 키를 분실하거나 키가 유출되면 암호문을 누군가가 복호화 할 수 있다는 단점

공개키 암호화 방식


공개키 Encryption
Data
Encrypted Data
개인키 Decryption


개인키 Encryption
Data
Encrypted Data
공개키 Decryption

공개키 암호화방식

- 대칭키 암호화 방식의 문제를 해결하고자 나온 방식
- 공개키, 개인키 두 개의 키를 한 쌍으로 각각 암호화와 복호화에 사용하는 방식으로 이를 RSA 알고리즘이라 함
- 보통 공개키로 암호화한 것을 개인키로 복호화
- 개인키를 먼저 생성하고 공개키를 파생하여 만듬
- 대칭키 방식에 비해 안전하지만 계산 과정이 복잡하고 연산 도중 컴퓨터의 자원이 많이 사용됨

다른 프로토콜의 종류

- FTP (File Transfer Protocol):.
- SMTP (Simple Mail Transfer Protocol):
- POP3 (Post Office Protocol 3):
- IMAP (Internet Message Access Protocol):
- DNS (Domain Name System):
- SSH (Secure Shell):

다른 프로토콜의 종류

- FTP (File Transfer Protocol):
- FTP는 파일 전송을 위한 프로토콜로, 파일을 서버와 클라이언트 사이에서 전송하는 데 사용됩니다. 주로 파일 공유나 웹 호스팅과 같이 파일 전송이 필요한 경우에 사용됩니다.

- SMTP (Simple Mail Transfer Protocol):
- SMTP는 전자 메일을 전송하는 데 사용되는 프로토콜입니다. 이메일 클라이언트가 메일 서버로부터 메일을 전송할 때 사용됩니다.

- POP3 (Post Office Protocol 3):
- POP3는 전자 메일 서버로부터 이메일을 다운로드하는 데 사용되는 프로토콜입니다. 이메일 클라이언트가 서버로부터 메일을 받아오는 데 사용됩니다.

- IMAP (Internet Message Access Protocol):
- IMAP는 전자 메일 서버에 저장된 이메일에 직접 접근하여 관리하는 데 사용되는 프로토콜입니다. 메일 서버와 클라이언트 사이의 이메일 동기화에 사용됩니다.

- DNS (Domain Name System):
- DNS는 도메인 이름을 IP 주소로 변환하는 데 사용되는 프로토콜입니다. 웹 브라우저가 도메인 이름을 입력하면 DNS를 통해 해당 도메인의 IP 주소를 찾아 웹 서버와 통신합니다.

- SSH (Secure Shell):
- SSH는 원격 컴퓨터에 안전하게 접속하기 위한 프로토콜입니다. 원격 접속, 파일 전송, 암호화된 터널링 등에 사용됩니다.

HTTPS를 사용하는 방법

- HTTPS를 사용하는 방법은 다음과 같은 단계로 이루어집니다:

- 1. SSL/TLS 인증서 구입:
- - SSL/TLS 인증서를 인증기관에서 구입합니다. 인증서는 도메인 이름과 회사 또는 개인의 정보를 포함합니다.

- 2. 인증서 설치:
- - 웹 서버에 구입한 인증서를 설치합니다. 보통 웹 서버 소프트웨어에서 인증서를 관리하는 기능이 제공됩니다.

- 3. 서버 설정 수정:
- - 웹 서버의 설정을 수정하여 HTTPS를 사용하도록 설정합니다. 이를 위해 보통 웹 서버 소프트웨어의 설정 파일을 수정합니다.

- 4. 포트 설정:
- - HTTPS는 기본적으로 443 포트를 사용합니다. 따라서 웹 서버의 포트 설정을 443으로 변경하여 HTTPS를 사용하도록 설정합니다.