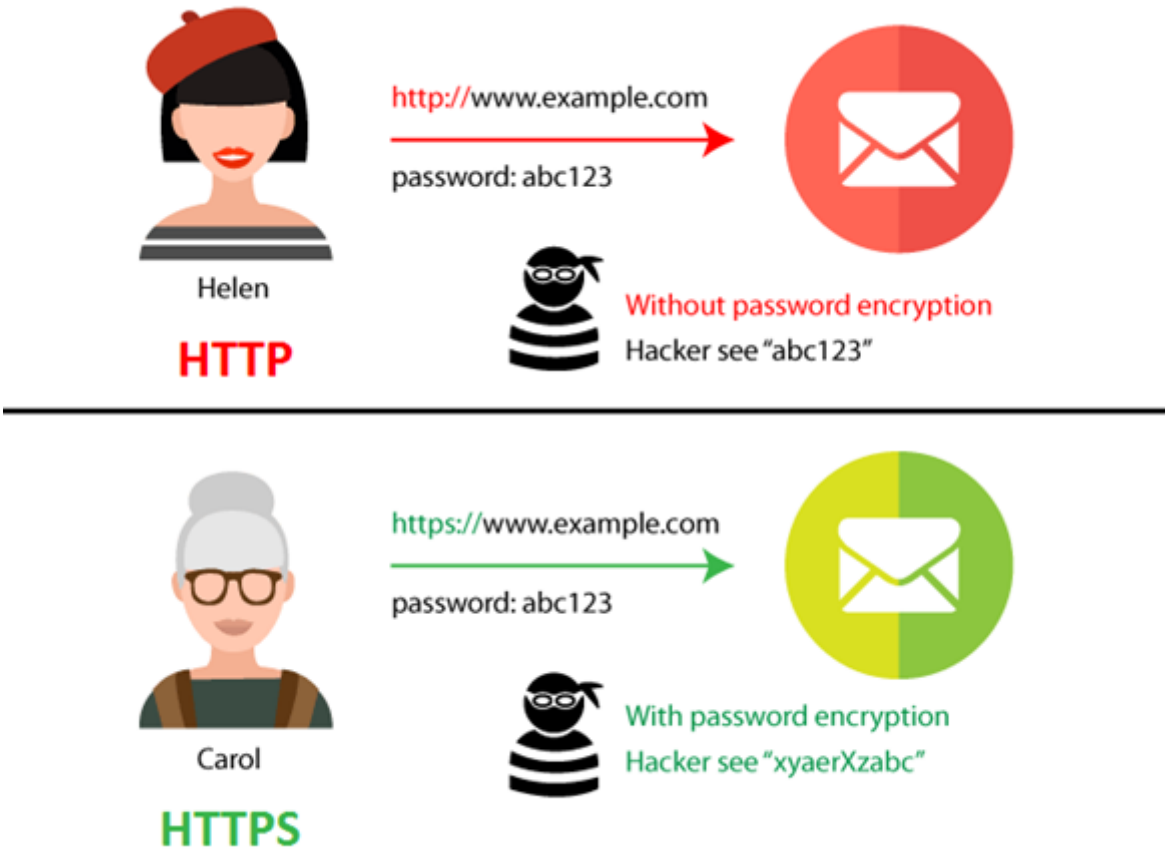
HTTP & HTTPS



HTTP(HyperText Transfer Protocol)

OSI 7계층 중 7 계층인 응용 계층(Application Layer) 에 해당하는 프로토콜

Stateless 한 특성을 가짐

▼ **Stateless ?**

서버가 클라이언트의 상태를 보존하지 않는다는 것

장점 : 서버 확장성이 높음

단점 : 클라이언트가 추가 데이터를 전송해야함

Method, Path, Version, Header, Body 등으로 구성됨

Request 와 Response 로 통신을 하는 비연결성(Connectionless) 프로토콜

▼ **Connectionless?**

클라이언트의 요청에 맞는 응답을 보낸 후 연결을 끊는 특징

평문 텍스트, 즉 암호화되지 않는 텍스트를 전송하는 프로토콜로, 중간자 공격에 취약함

변조, 위장, 도청에 취약함

HTTPS (HyperText Transfer Protocol over Secure Socket Layer)

쉽게 말해, HTTP의 단점인 보안성을 보완한 프로토콜

인터넷 상에서 정보를 암호화하는 TLS/SSL 프로토콜을 사용함

텍스트를 암호화함

TLS (Transport Layer Security) : 전송 계층 보안

가장 최신 기술로, 더 강력한 버전의 SSL

그러나 SSL 이 더 일반적으로 사용되는 용어이므로, 여전히 보안 인증서는 SSL 이라고 불림

SSL (Secur Socket Layer)

Netscape 사에서 웹서버와 웹브라우저 간의 보안을 위해 만들어짐

공개키/개인키 대칭키 기반으로 사용함

SSL 흐름

1. 사이트에서 인증기관 (CA: Certificate Authority) 에 인증 요청
2. 인증기관에서 검증 후에 사이트의 공개키와 정보를 인증기관의 개인키로 암호화, 인증기관의 공개키는 브라우저에 제공
3. 사용자가 사이트로 접속 요청 시 사이트는 인증서 전송

4. 사용자는 브라우저에 내장된 공개키로 인증서를 복호화 → 사이트의 공개키로 대칭키를 암호화하여 전송
5. 사이트는 전송받은 암호화된 대칭키를 사이트의 개인키로 복호화 → 사용자, 사이트 같은 대칭키 획득
6. 전송 시 해당 대칭키로 암호화하여 전송

HTTPS 통신 흐름

1. 애플리케이션 서버 (A) 를 만드는 기업은 HTTPS 적용을 위해 공개키와 개인키를 만듦
2. 신뢰할 수 있는 CA 기업을 선택하고, 그 기업에게 공개키 관리를 부탁하며 계약을 함

> **CA: Certificate Authority.** 공개키를 저장해주는 신뢰성이 검증된 민간기업

3. 계약 완료된 CA 기업은 해당 기업의 이름, A 서버 공개키, 공개키 암호화 방법을 담은 인증서를 만들고, 해당 인증서를 CA 기업의 개인키로 암호화해서 A 서버에게 제공함
4. A 서버는 암호화된 인증서를 갖게 되었음. 이제 A 서버는 A 서버의 공개키로 암호화된 HTTPS 요청이 아닌 요청이 올 경우 이 암호화된 인증서를 클라이언트에게 넘김
5. 클라이언트 입장에서, 예를 들어 A 서버로 index.html 파일을 달라고 요청. 그러면 HTTPS 요청이 아니므로 CA 기업이 A 서버의 정보를 CA 기업의 개인키로 암호화한 인증서를 받게됨
6. CA 기업의 공개키는 브라우저가 이미 알고있음 (세계적으로 신뢰 가능한 기업으로 등록되어 있으므로, 브라우저가 인증서를 탐색하여 해독이 가능함
7. 브라우저는 이를 해독한 뒤 A 서버의 공개키를 얻게 되었음
8. 이제 A 서버와 통신할 때는 A 서버의 공개키로 암호화해서 요청을 보냄

> HTTPS 도 무조건 안전한 것은 아님. (신뢰받는 CA 기업이 아닌, 자체 인증서를 발급한 경우 등)
> 이 경우, HTTPS 이나 브라우저에서 `주의 요함`, `안전하지 않은 사이트` 와 같은 알림으로 주의를 받게됨

HTTPS 보안 외의 장점

1. **검색엔진 최적화(SEO, Serarch Engine Optimization)에 있어서도 큰 혜택을 볼 수 있음**

   구글에서는 HTTPS 를 사용하는 웹사이트에 대해서 검색 순위 결과에 약간의 가산점을 주겠다고 발표하였음

2. **가속화된 모바일 페이지(AMP, Accelerated Mobile Pages) 를 만들 수 있음**

   AMP 를 만들기 위해서는 HTTPS 프로토콜을 사용해야만함

⇒ 모바일 친화적인 웹사이트를 만드는 것과 모바일 검색순위 및 지역에 SEO 를 증가시키는 것이 점점 더 중요해 지고 있으므로, HTTPS 를 사용하는 것이 필수라고 볼 수 있음
