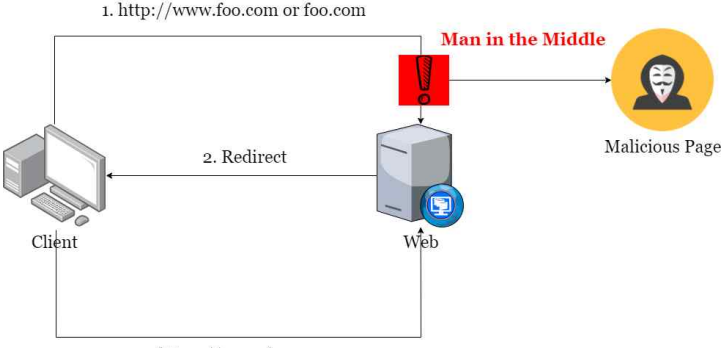
Web은 Network에서 How?

Web은 뒤에서 어떻게 동작하는가?
HSTS, HTTP에 관해 알아보자

HSTS

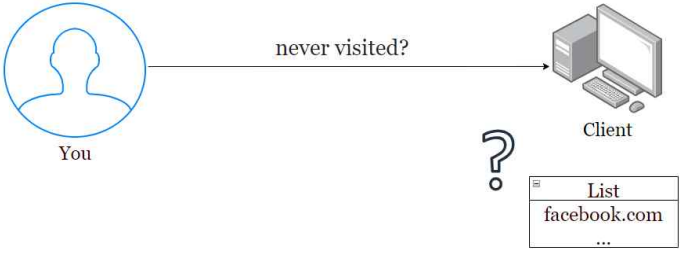
HTTP 대신 HTTPS만을 사용해 통신해야 한다고
Web Site가 Browser에 알리는 보안 기능

HSTS



안전한 원본 페이지 X가 아닌 => 악의적인 다른 페이지로
Redirect되는 Man in the middle Attack 위험

HSTS preload



Chrome 보안 팁 => HSTS preload list [https://hstspreload.org/]
첫 번째 방문에도 Strict Transport Security사 자동으로 활성화되는 Domain 목록
Firefox, Safari, Opera, Edge - Chrome의 HSTS preload list 통합

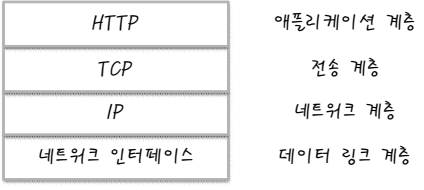
HSTS preload

```
< HTTP/2 200
< server: NWS
< date: Sat, 26 Feb 2022 13:59:31 GMT
< content-type: text/html; charset=UTF-8
< set-cookie: PM_CK_loc=9d49771401a0034347470e6c71926b14bc8c100cba7c5e4021189a7f
8fedcc43; Expires=Sun, 27 Feb 2022 13:59:31 GMT; Path=/; HttpOnly
< cache-control: no-cache, no-store, must-revalidate
< pragma: no-cache
< p3p: CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV I
NT DEM STA PRE"
< x-frame-options: DENY
< x-xss-protection: 1; mode=block
< strict-transport-security: max-age=63072000; includeSubdomains
< referrer-policy: unsafe-url
```

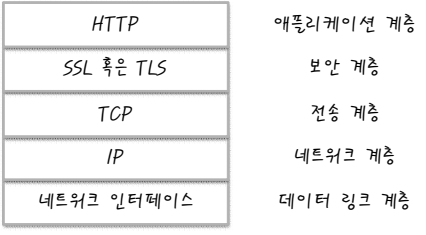
Curl -v http://www.naver.com 입력시 HTTP Header
strict-transport-security 헤더 옵션이 걸려있음

HTTPS

HTTP 프로토콜 + 대칭 / 비대칭 인증서 기반 암호 집합

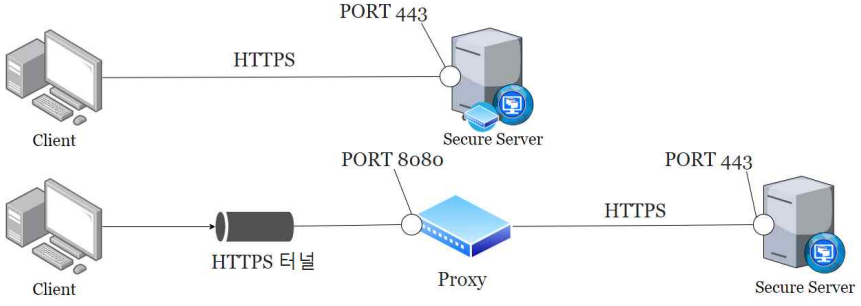


(A) HTTP



(A) HTTP

SSL



1) Client : server에 443 포트로 연결 & Binary 포맷으로 된 몇몇 SSL 보안 매개변수를 교환하면서, HandShake를 하고, 암호화된 HTTP 명령을 보냄

SSL or TLS

SSL 3.0을 참고로 해 RFC 2246으로 표준화한 것 => TLS

응용 프로그램 자체 구현 가능
(EAP, IPsec 등 – OS 등에 밀접 관련)

전송 계층 상에서 Client, Server에 대한 인증, 데이터 암호화 수행

SSL Handshake

Protocol 버전 번호 교환

양쪽이 알고 있는 암호 선택

양쪽의 신원을 인증

채널을 암호화하기 위한 임시 세션 키 생성

SSL Handshake 과정

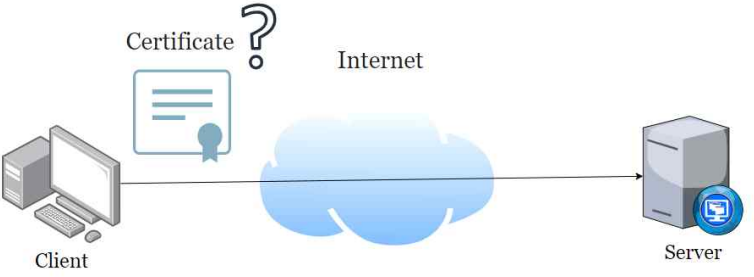
(1) 클라이언트가 암호 후보들을 보내고 인증서 요구

(2) 서버는 선택된 암호화 인증서 보냄

(3) 클라이언트가 비밀정보를 보냄 & 클라이언트, 서버는 키를 만듦

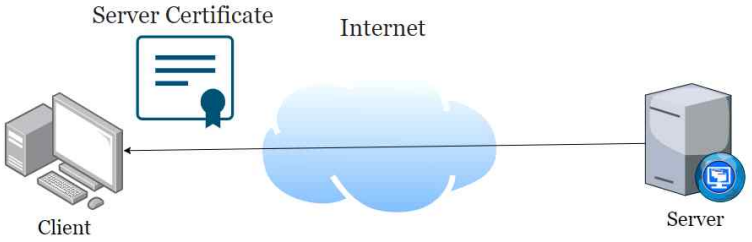
(4) 클라이언트와 서버는 서로에게 암호화를 시작한다고 말해줌

SSL Handshake



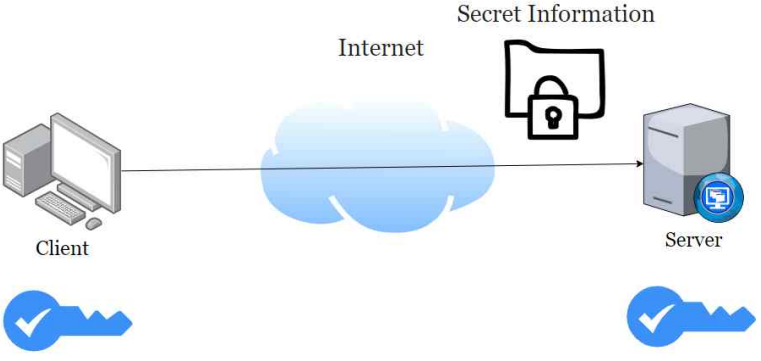
(1) 클라이언트가 암호 후보들을 보내고 인증서 요구

SSL Handshake



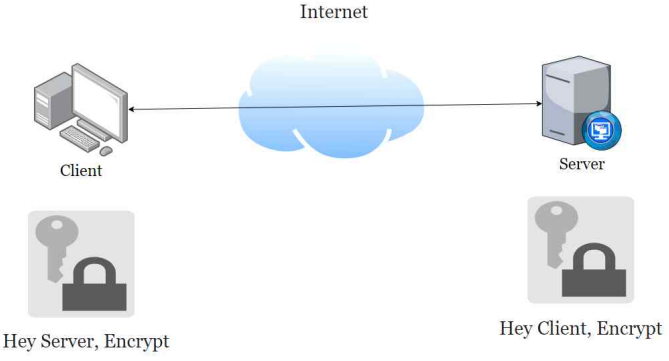
(2) 서버는 선택된 암호화 인증서 보냄

SSL Handshake



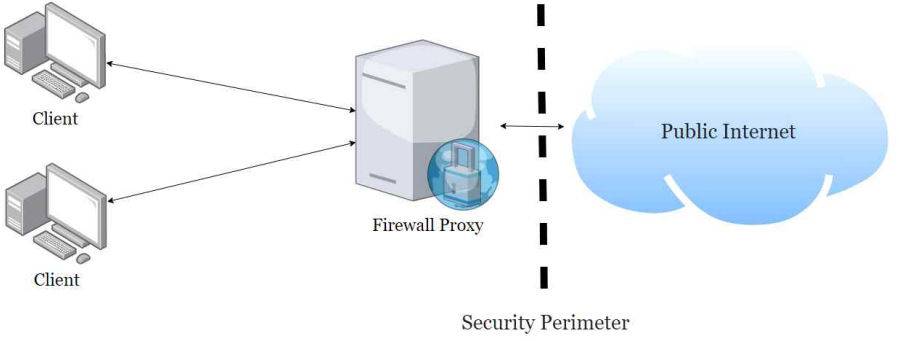
(3) 클라이언트가 비밀정보를 보냄 & 클라이언트, 서버는 키를 만듦

SSL Handshake



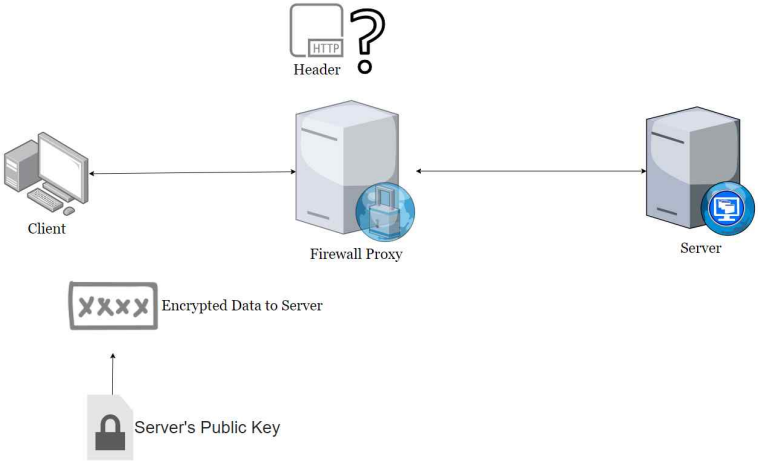
(4) 클라이언트와 서버는 서로에게 암호화를 시작한다고 말해줌

Proxy & Security Traffic Tunneling

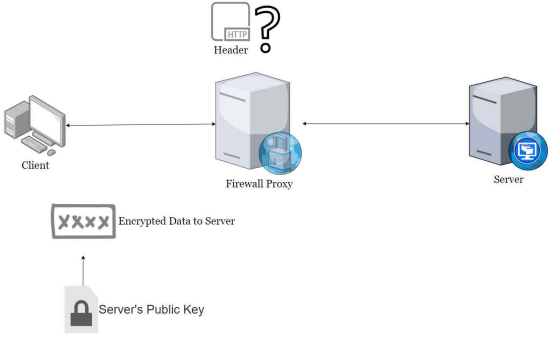


Proxy – 방화벽 라우터가 HTTP 트래픽의 교환을 허락한 유일한 장치, 바이러스 검사 & 기타 콘텐츠 제어

Proxy & Security Traffic Tunneling



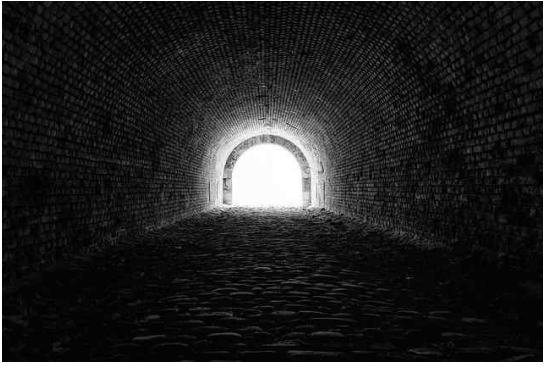
Proxy & Security Traffic Tunneling



Proxy가 Header를 읽을 수 없다면?

Proxy는 요청을 어디로 보내야 하는지 알 수 없게됨

????



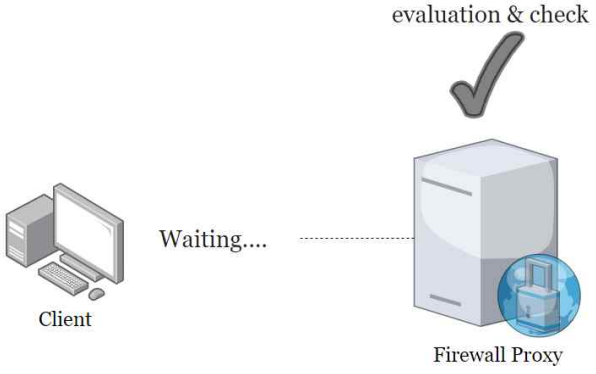
Solution: HTTPS SSL Tunneling Protocol

클라이언트는 Proxy에게 연결하고자 하는 안전한 호스트와 포트를 말해준다
(암호화가 시작되기 전의 평문으로!)

HTTP -> CONNECT 메소드 : 평문으로 된 종단 정보를 전송하기 위해 사용됨

Connect? : Proxy에게 희망하는 Host, Port 번호로 연결 요구 => 완료되면, 터널 생성
(클라이언트와 서버 사이에 데이터가 직접적으로 오갈 수 있음)

Solution: HTTPS SSL Tunneling Protocol



Client는 Proxy로부터 응답 대기 & Proxy는 확인 작업

Solution: HTTPS SSL Tunneling Protocol

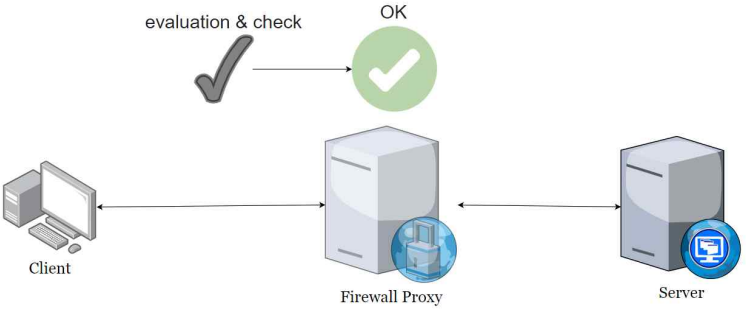
```
CONNECT server.example.com:80 HTTP/1.1
Host: server.example.com:80
Proxy-Authorization: basic aGVsbG86d29ybGQ=
```

"원 서버의 Host:Port <HTTP 버전 문자열> CRLF"

0개 이상의 HTTP 요청 헤더줄들이 이어지고, 빈 줄 하나가 옴

if handshake ok, SSL 데이터 전송이 시작됨

Solution: HTTPS SSL Tunneling Protocol



HTTP/1.1 200 Connection Established

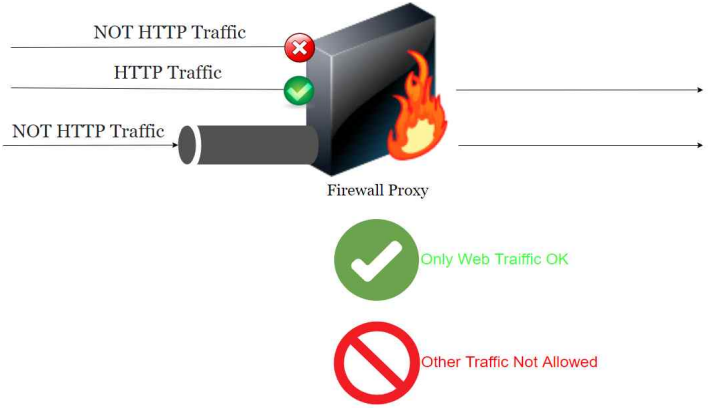
Proxy는 목적지 서버로 연결하고 성공하면 응답을 Client에게 보냄

Tunneling

HTTP Tunneling – 제한된 네트워크 연결 조건에서 네트워크 링크를 만드는데 사용

웹 터널 – HTTP Protocol을 지원하지 않는 Application에
HTTP Application을 사용해 접근하는 방법 제시

Tunneling



왜? HTTP 커넥션 안에 HTTP가 아닌 트래픽을 없기 위해서

Tunneling Protocol

https://en.wikipedia.org/wiki/Tunneling_protocol

https://developer.mozilla.org/en-US/docs/Web/HTTP/Proxy_servers_and_tunneling

참고

HTTPS를 강제하는 HSTS 사용하기

Chrome Network Tools

Chrome Network Features Reference

SSL vs TLS

HTTP Tunneling

Book: HTTP 완벽 가이드