# Garantiscono l'interscambio comunicativo all'interno del modello client-server. Ecco come funzionano i protocolli HTTP e HTTPS e quali sono le differenze tra i due

Sono le fondamenta su cui si regge l'intera comunicazione e trasmissione di dati all'interno del **World Wide Web**. I **protocolli HTTP** e **HTTPS**, creati a cavallo tra il 1989 e il 1994 da, rispettivamente, Tim Berners-Lee e Netscape Communications, svolgono il ruolo di mediatore all'interno del **modello client-server** e permettono lo scambio di informazioni tra due nodi della Rete gestendo sessioni di comunicazione, richieste e quant'altro connesso a questo processo.

## Il protocollo HTTP

Nato in concomitanza con la nascita del web moderno, il protocollo HTTP (*HyperText Transfer Protocol*, “protocollo di trasmissione di documenti ipertestuali”) funziona come un protocollo di richiesta-risposta all'interno dell'architettura *client*-server. Un esempio classico di ciò è quello di un **web browser,** che svolge la funzione di *client* (cliente), sul nostro computer mentre **un'applicazione** (un software, un documento digitale, ecc.) **disponibile su un altro computer** connesso alla rete e ospitante una risorsa web, svolge la funzione di server. In un caso come questo, il server risponde all'interrogazione in arrivo dal *client* fornendo la risorsa digitale richiesta come ad esempio una pagina HTML.

All'interno del modello ISO/OSI di reti di calcolatori, il protocollo HTTP appartiene al settimo e ultimo livello, quello delle applicazioni. Progettato all'interno dell'Internet Protocol Suite, il suo funzionamento presuppone l'utilizzo di protocollo appartenente al livello di trasporto che sia sicuro e affidabile. Solitamente è utilizzato il protocollo TCP (*Transmission Control Protocol*), anche se in alcuni casi possono essere utilizzati protocolli meno affidabili come lo User Datagram Protocol.

Le risorse HTTP sono localizzate e identificate nel web grazie al sistema dell'**Uniform Resource Identifiers** (URI) e più specificatamente con l'**Uniform Resource Locators** (URL), ovvero il comune “indirizzo web”.

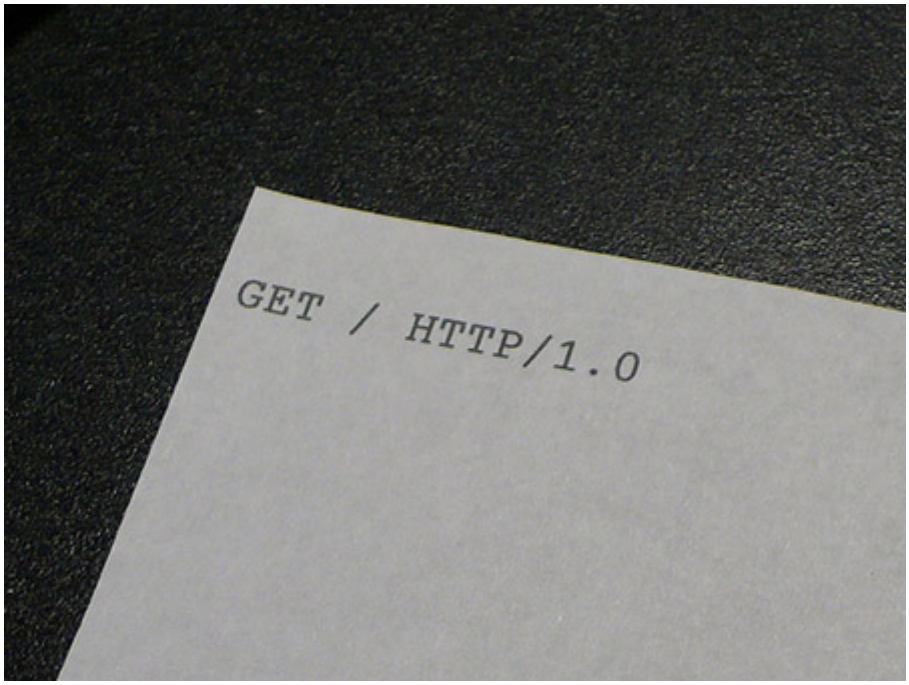
## Come funziona una sessione HTTP

Quando *client* e server si mettono in contatto l'uno con l'altro si stabilisce una **sessione HTTP**. Si trattadi una sequenza di richieste di rete tra i due nodi, grazie alla quale l'uno (il *client*) chiede e ottiene informazioni dall'altro (il server). Una sessione ha inizio quando il *client* stabilisce una connessione TCP con una particolare porta del server (tipicamente la **porta 80**) inviando una richiesta di informazioni o risorse.

Il server risponde con una **linea di stato** (del tipo: “HTTP1.1 200 OK”) e un messaggio contenente le informazioni richieste o, eventualmente, un messaggio di errore.

## Metodi di richiesta del protocollo HTTP

Affinché lo scambio di informazioni vada a buon fine, il *client* deve inviare le proprie richieste seguendo una *grammatica* specifica e articolata. All'interno di questo schema troviamo alcune parole chiave dette **verbi** che identificano l'azione richiesta dal *client* al server.




Tra i verbi più comuni troviamo:

- **GET** (“prendere”): rappresenta la richiesta di una specifica risorsa ospitata sul server. Con questo comando è possibile recuperare il dato ma non compiere altre operazioni (come ad esempio cancellare o modificare il dato sul server)

- **HEAD** (“testa”): stessa funzionalità di GET ma senza ottenere l'intero corpo della risposta limitandosi alle sole informazioni riassuntive presenti nell’intestazione. Utile nel caso si debbano recuperare delle *metainformazioni*scritte nell'*header* (“intestazione”) di una risorsa web senza essere costretti a scaricare l’intero documento

- **POST** (“spedire”): richiede al server di accettare l'entità allegata al messaggio inviato come un nuovo subordinato della risorsa web identificata dall'URI. Può essere, ad esempio, una nota da apporre ad una risorsa esistente, un messaggio per una newsletter o un forum, ecc

- **DELETE** (“cancellare”): richiesta di cancellazione o eliminazione di una specifica risorsa dal server

## Il protocollo HTTPS

Il **protocollo HTTPS (HTTP *Secure***, “HTTP Sicuro”**)** è un protocollo di comunicazione realizzato nel 1994 per il web browser Netscape – e successivamente adottato da tutti gli altri – per rendere più sicuro ed affidabile lo scambio di informazioni tra due nodi del web. Tecnicamente non si tratta di un protocollo vero e proprio ma dell'applicazione del protocollo SSL/TLS in congiunzione al protocollo HTTP con lo scopo di prevenire attacchi del tipo *man-in-the-middle* (letteralmente “**uomo nel mezzo**”, tipologia di attacco hacker nella quale una terza entità si inserisce nella linea di comunicazione tra *client* e server ed intercetta tutti gli scambi informativi tra i due nodi).

Per far ciò, il protocollo HTTPS garantisce l'identificazione del sito web che si sta visitando e del server web che lo ospita. Inoltre, fornisce la **crittografia bidirezionale della comunicazione**, proteggendo l'utente dal pericolo di essere intercettato o di visitare siti manomessi. Storicamente questo protocollo è stato utilizzato per proteggere transazioni economiche e finanziarie telematiche, servizi di posta elettronica e per proteggere lo scambio di informazioni sensibili all'interno di grandi *corporation*. Tra la fine del decennio passato e l'inizio del decennio attuale, l'HTTPS ha trovato applicazione sempre più ampia, arrivando a certificare l'autenticità di siti web di qualsiasi genere.

Ad "assicurare" i browser web sull'autenticità di un sito e del server che lo ospita sono le cosiddette **autorità di certificazione** (VeriSign, Microsoft, ecc.) che forniscono certificati digitali che attestano, appunto, l'autenticità del portale web che si sta visitando.

## Le differenze tra HTTP e HTTPS

La maggiore differenza tra i due protocolli sta nella crittografia. Mentre con l'HTTP tutte le comunicazione avvengono in "chiaro", quindi "leggibili" da chiunque riesca ad intercettare il flusso di scambio dati, l'HTTPS crea un canale di comunicazione sicuro al di sopra di una rete di comunicazione non sicura utilizzando, per l'appunto, la crittografia. Per far questo anzitutto procede ad accertare l'autenticità del proprio interlocutore attraverso la valutazione della certificazione digitale fornita dal server, dopodiché utilizza questo stesso certificato come chiave di criptazione che utilizzerà poi per crittografare e decrittografare tutti i dati scambiati con esso. Cambia, inoltre, la porta standard utilizzata per mettersi in contatto e comunicare con il server: nel caso dello HTTP è la porta 80, nel caso dello HTTPS è la porta 443.

## La falla Heartbleed

Non più di un mese fa, un gruppo di hacker è riuscito a "bucare" il protocollo HTTPS, ritenuto sino ad allora praticamente inviolabile. Sfruttando una "deficienza strutturale" del sistema OpenSSL, i pirati informatici sono riusciti a mettere in atto attacchi di tipo man-in-the-middle anche su server e siti web protetti dal protocollo HTTPS. La falla, divenuta celebre con il nome Heartbleed, ha messo in pericolo i dati personali di milioni di utenti.