PROTOCOLLO HTTPS

Il protocollo di base funziona encapsulando HTTP in un layer di sicurezza TLS. Oltre a ciò però estende le funzioni di TLS per supportare il VIRTUAL HOSTING. Il protocollo inoltre integra meccanismi per proteggersi dai DOWNGRADE ATTACK (HTTPS → HTTP).

VIRTUAL HOSTING

Il protocollo HTTP permette di effettuare il MULTIPLEXING di più SITI WEB sullo stesso SERVER. Tramite il protocollo HTTP infatti è possibile raggiungere più siti WEB presenti sullo stesso server (in ascolto sulla porta 80) grazie ad un HEADER obbligatorio che definisce il VIRTUAL HOST richiesto dal client. Questo permette inoltre di associare vari virtual siti web allo stesso indirizzo IP.

In realtà nelle architetture moderne le applicazioni vengono DISTRIBUITE su più macchine, ponendo all'ingresso della rete un REVERSE PROXY che accetta e distribuisce le richieste.
Questo è un problema se si utilizza HTTPS perché l'hostname, usato per fare reverse proxy, è nell'header che viene CIFRATO. Poiché siti web diversi possono usare CERTIFICATI DIVERSI, il reverse proxy potrebbe non essere in grado di leggere l'header (perché non ha il certificato)
Per risolvere questo problema viene sfruttata la possibilità di aggiungere delle INFORMAZIONI FACOLTATIVE (dette ESTENSIONI) all'interno del CLIENT HELLO (pacchetto inviato dall'handshake TLS). Viene quindi inserito, IN CHIARO, il VIRTUAL HOSTNAME del sito a cui si ci vuole connettere, per consentire quindi al reverse proxy di funzionare. Il nome dell'host viene quindi spedito sia nell'header che in chiaro nell'handshake e ciò è un difetto di sicurezza NECESSARIO a far funzionare il tutto.

È in corso di discussione una modifica allo standard TLS 1.3 per poter gestire anche la cifratura dell'hostname permettendo comunque il ~~TLS~~ reverse proxying.
L'inserimento dell'hostname nell'handshake TLS rompe la separazione totale dei compiti tra TLS e HTTPS.

SSL STRIPPING

Normalmente se uno user-agent fa una richiesta HTTP il server risponde con un REINDIRIZZAMENTO verso il protocollo HTTPS, quindi effettua un UPGRADE del protocollo.

L'attacco SSL STRIPPING, effettuato tramite MAN IN THE MIDDLE, consiste proprio nello sfruttare questa prima connessione NON SICURA in HTTP per intercettare la connessione e dirottarla a proprio volta una connessione HTTPS con il server. L'unico modo per evitare questo attacco è dire al client di usare sempre e solo la connessione HTTPS fin da subito.

Alcuni approcci per affrontare questo rischio sono:

I) Agire sul client, magari con un'estensione che faccia l'utilizzo di HTTPS

II) HTTP STRING TRANSPORT SECURITY (HSTS): il server aggiunge un header alla prima risposta HTTP per comunicare al client di usare sempre e solo HTTPS in futuro. Questo protocollo riprende quindi l'approccio TOFU!

Esistono dei servizi che "collezionano" elenchi di web servers per controllare chi utilizza HSTS ed inserire questi elenchi all'interno dei browser, per evitare il rischio dell'approccio TOFU per alcuni siti particolarmente sensibili.