1 Overview about TLS and IPsec

TLS ha come progenitore il protocollo SSL da un certo momento in poi cambiò nome a seguito di importanti aggiornamenti al protocollo. Nelle varie versioni di TLS hanno aggiunto notevoli funzionalità aggiuntive tra cui il supporto al protocollo di trasporto dati UDP, l'utilizzo (per poi esser abbandonati in TLV v.1.3) di algoritmi di encrypt come MD5 e SHA-1 ed infine, in TLS v.1.3 (la versione ad oggi in uso) di utilizzare i protcolli AEAD:protocolli che garantiscono confidenzialità e integrità dei dati. Dato che TLS utilizza TCP (o nella versione DTLS UDP), si assegna ad ogni processo una socket per rendere sicuri i dati di livello applicativo è considerato un protocollo che opera tra il livello di trasporto e il livello applicativo. Il suo nome, quindi, è fuorviante *Transport Layer Security*. Come vedremo in seguito questa possibilità di aprire una socket per ogni applicazione risulta problematica. Infine, TLS protegge solamente il payload di TCP e non l'intero pacchetto poiché per come è stato progettato, se così non fosse, non si saprebbe dove e a chi inviare i pacchetti. Da questo aspetto se ne conclude che l'header del pacchetto TCP può essere modificato e quindi si è soggetti ad attacchi di tipo TCP spoofing, MITM, CCA etc...
Un altro protocollo, sviluppato di pari passi a TLS, è IPsec. Esso viene considerato una versione molto più crittograficamente sicura e, data la problematica di TLS nell'utilizzare socket diverse per le varie applicazione, si pone tra il livello di trasporto e di rete. Infatti, "monta"sopra IP e, grazie a questa proprietà, è possibile rendere crittograficamente sicuro l'intero pacchetto TCP/UDP/Altro incapsulando il pacchetto crittografato IP in un altro paccheto così da nascondere l'intero contenuto di quello che si voleva mandare.
Da qui il concetto di **traffic flow confidentiality** che rappresenta un nuovo requisito di sicurezza:la straficazione del pacchetto deve aggiungere sicurezza al protocollo.
Una differenza tra TLS e IPsec è che nella loro progettazione hanno sviluppato il set-up dell'ambiente e il modo in cui si trasferiscono i dati in maniera differente. Questi due aspetti, infatti, rappresenzanto due concetti cardini nello sviluppo di un protocollo di sicurezza.
Quindi, TLS decise di unire il set-up dell'ambiente e il trasferimento di dati delegano il primo aspetto venne realizzato tramite una fase di handshake in cui si negoziano gli algoritmi e, tramite crittografia asimmetrica, si comunicano tali chiavi, necessarie a garantire integrità e confidenzialità, la seconda fase, invece viene detta record phase;mentre in IPsec, si decise di disaccoppiare questi due aspetti: la prima fase di set-up è delegata ad un protocollo automatico detto IKE (Internet Key Exchange);la seconda fase, di trasferimento dati delegata al protocollo utilizzato a supporto di IPsec.

2 TLS Protocol Stack

TLS, nella record phase, utilizza un protocollo detto TLS Protocol Stack che è composto da diverse componenti dette:TLS Record Protocol responsabile del

trasferimento dei dati, Handshake Protocol responsabile dello scambio delle informazioni necessarie alla sicurezza della comunicazione, Alert Protocol responsabile della definizione dei messaggi di warning/alert, il Change Cipher Suite responsabile dell'inizializzazione del cipher. Il Protocol Stack si interpone tra HTTP e TCP.

3 TLS Record Protocol

Il TLS Record Protocol è responsabile del trasferimento dei dati dell'utente e, in base agli obiettivi proposti da TLS, a questi dati bisogna aggiungere integrità e confidenzialità. In aggiunte, per questioni tecniche, si decise di aggiungere anche una funzione di compressione dei dati. In totale, quindi vi sono queste funzionalità: compressione, HMAC (per l'integrità) e Encryption. Quello che si potrebbe pensare è che non vi è alcuna differenza tra l'ordine in cui vengono effettuate queste operazioni, tuttavia non è così:ne è un esempio l'attacco CRIME in cui si frutta la compressione e poi l'encryption per decifrare l'intero dato che abbiamo ricevuto. Inoltre, un'altra operazione totalmente scorretta è quella di effettuare prima l'integrità con HMAC e poi l'encryption.

4 Message Authentication Code