SICUREZZA NELLE RETI a. a. 2012/13 – Riassunto IPsec

by Salvatore Fresta

IPsec (**Internet Protocol Security**) è una suite di protocolli progettata per rendere sicure le comunicazioni IP grazie alla possibilità di **trasferire pacchetti crittati** e **autenticare** i nodi. È supportato nativamente in IPv6 mentre è un facoltativo in IPv4.

Con IPsec è possibile sapere se due nodi si possono connettere oppure no, è possibile autenticare l'origine dei dati (**impossibile fare spoofing**), garantire l'**integrità dei campi immutabili** (ad esempio il TTL non può essere immutabile), garantire **confidenzialità** dei dati (dati cifrati) ed infine garantire **protezione da replay** (reintrodurre in rete pacchetti già inviati in precedenza) grazie ad una tecnica *sliding window* con contatore.

IPsec **è formato da diversi protocolli**, quali:

- **Authentication Header** (AH): usato per l'integrità, l'autenticazione e la protezione da replay.
- **Encapsulating Security Payload** (ESP): usato per la confidenzialità.

Entrambi usano una Security Association (SA) per lo scambio di chiavi, **identificata** dal campo *Security Parameter Index*.

Dunque **ad ogni connessione IPsec** è abbinata una SA (i nodi che vogliono comunicare si mettono d'accordo sul tipo di algoritmi crittografici che useranno, quali sono le chiavi coinvolte ecc..).

Le SA sono diverse per AH e ESP (quindi ce ne sono due). Esse sono stabilite **staticamente** (grazie all'amministratore di sistema che configura la rete) o **dinamicamente** (protocolli che stabiliscono lo scambio di credenziali e parametri di sicurezza).

Come detto in precedenza, IPsec offre protezione da attacchi di replay. In particolar modo, AH adotta la tecnica **sliding window.** In breve la tecnica consiste nell'accettare in un determinato **intervallo di tempo solo** i pacchetti **contenuti** in una certa **finestra temporale** (ad esempio per questo intervallo di tempo ricevo solo i pacchetti numerati dal 2 al 10, quindi vanno scartati quelli inferiori di 2 e superiori a 10). La tecnica fa riferimento al fatto che se voglio **reinizializzare** il contatore, bisogna accordarsi su una **nuova SA**.

ALGORITMO SLIDING WINDOW

È tutto molto semplice. Prima di tutto si **inizializza a 0 un array** di lunghezza L (solitamente 64 bit) che **rappresenta la finestra temporale**:

```
window[L] = 0;
```

Non appena si riceve un pacchetto con un determinato indice *n* (per esempio 110), ne terrò traccia

inserendolo **alla fine** dell'array:

```
window[L] = n; // Se presuppongo che l'array inizia da 0 invece che da 1, l'ultimo elemento sarà L-1 e non L
```

Da questo momento arriveranno altri pacchetti con un contatore *i* generico. Possono verificarsi **3 casi**:

```
if(i <= n && i >= n-L+1 && check_integrity() == true) {

        1. Pacchetto integro e contenuto nella finestra

        /* Il campo dove dovrebbe stare l'indice del pacchetto ricevuto non
           è più inizializzato a 0, ciò significa che l'avevo già ricevuto */

        if( window[i+L+n] > 0 )
                Tentativo di Replay
        else
                window[i+L+n] = i; // In caso contrario memorizzo l'indice del pacchetto

}

else if( i<= n-L) {

        2. Pacchetto fuori dalla finestra, più vecchio, quindi lo scarto

}

else if(i>n && check_integrity() == true) {

        3. Pacchetto fuori dalla finestra ma più nuovo. Sposto (sliding) la finestra in modo tale che i diventi il nuovo limite accettabile

}
```

Algortimo riassunto: **inizializzo l'array** della finestra ed **aggiorno il limite accettabile** di pacchetti da ricevere con il l'indice del primo pacchetto che ricevo. Dopo di che, **ad ogni pacchetto che ricevo**, controllo se è contenuto nella finestra e se è integro. Se è contenuto ma l'ho già ricevuto in passato, allora è un **replay attack,** altrimenti ne memorizzo l'indice nell'array. Può capitare che ricevo un pacchetto che vada sotto il limite inferiore della finestra (**pacchetto vecchio**), ed in quel caso lo scarto, o che ricevo un pacchetto che va oltre il limite massimo della finestra. In quest'ultimo caso, dopo averne controllato l'integrità, setto questo **nuovo come indice massimo** della finestra, effettuando quindi uno sliding.

Come già detto, ESP serve per la crittazione dei pacchetti, la quale può avvenire in due modalità:

1. **Transport**: critto i livelli superiori a quello IP (transport (TCP/UDP) ecc..)
2. **Tunnel**: critto i pacchetti IP

Il problema di IPsec è che è **difficile da gestire** con un firewall e non è banale configurare sistemi come proxy e NAT che manipolano i pacchetti. Motivo per cui IPsec non ha riscontrato un grande successo.