Azzolini Riccardo 2021-05-13

IPSec

1 Associazioni di sicurezza

AH e ESP utilizzano delle tecniche crittografiche, dunque le due parti comunicanti devono condividere un certo numero di chiavi segrete e accordarsi sugli algoritmi da impiegare. A tale scopo, le due parti definiscono delle **associazioni di sicurezza** (**SA**, **Security Association**).

In sostanza, una SA determina come i pacchetti devono essere processati, specificando gli algoritmi crittografici, le chiavi e i vettori di inizializzazione utilizzati per AH o ESP, il tempo di validità della SA stessa, l'uso della modalità trasporto o tunnel, ecc.

Le associazioni di sicurezza hanno alcune caratteristiche importanti:

- Una SA è un'associazione **unidirezionale**, che riguarda uno solo verso del traffico tra le due parti comunicanti, dunque per una comunicazione bidirezionale servono almeno due SA.
- Una SA fornisce **un solo servizio**, AH o ESP (non entrambi). Avere più servizi sullo stesso traffico nello stesso verso è possibile creando combinando più SA, perché *uno stesso pacchetto può essere processato da più SA*. Ad esempio, per ottenere una comunicazione autentica (AH) e riservata (ESP) in entrambi i versi tra due nodi A e B della rete sono necessarie almeno 4 SA: una per l'applicazione di AH nel verso $A \to B$, una per ESP nel verso $A \to B$, e altre due per AH ed ESP nel verso $B \to A$. Inoltre, è possibile creare una qualsiasi combinazione di SA in modalità trasporto e tunnel, e a un pacchetto possono applicarsi SA diverse in diversi tratti della rete (ad esempio, si potrebbero avere delle SA in modalità trasporto tra due host e delle ulteriori SA tra due gateway che forniscono protezione in modalità tunnel per un tratto della rete attraverso la quale gli host comunicano).

Una SA è identificata univocamente da tre informazioni:

- un codice di 32 bit chiamato **Security Parameter Index** (**SPI**);
- l'**indirizzo IP di destinazione**;
- l'**identificatore del protocollo di sicurezza**, che indica se l'associazione è di tipo AH o ESP.

L'SPI viene riportato nell'header AH o ESP per permettere al destinatario di identificare la SA da utilizzare per elaborare tale header, il che è necessario perché tra due host possono esistere più SA.

1.1 Security policy

IPSec consente di definire delle regole che specificano a quale traffico si deve applicare un'associazione di sicurezza. Tali regole sono chiamate **security policy** e sono memorizzate in un apposito **Security Policy Database** (**SPD**). Una security policy è caratterizzata da:

- un insieme di **selettori** che permettono di selezionare il traffico;
- l'identificatore dell'SA da applicare al traffico selezionato.

I possibili selettori sono:

- gli indirizzi IP di sorgente e destinazione;
- l'ID dell'utente, preso dal sistema operativo;
- delle label che indicano il livello di sensibilità dei dati;
- il protocollo impiegato al livello di trasporto;
- i numeri di porta di sorgente e destinazione.

Gli eventuali pacchetti a cui non si applica alcun selettore vengono inviati senza servizi di sicurezza.

Le security policy potrebbero ad esempio essere utili in uno scenario di collegamento tra due gateway, nel quale si vorrebbero magari applicare tra i gateway SA diverse a seconda di quali siano gli host che comunicano tramite i gateway.

1.2 Generazione delle SA

Per creare una SA le due parti comunicanti comunicanti si devono mettere d'accordo sui parametri della SA (gli algoritmi da utilizzare, il tempo di validità, ecc.) e sulle chiavi da utilizzare. IPSec prevede che ciò possa avvenire in due modi:

- **Manuale**: un amministratore di sistema configura manualmente le macchine che implementano IPSec. Questa soluzione va bene in scenari semplici, per reti piccole e statiche (cioè la cui topologia non cambia di frequente).
- **Automatico**, tramite un sistema standard chiamato **Internet Key Exchange** (**IKE**), che comprende due aspetti:

– un protocollo per attivare, negoziare, modificare e cancellare le associazioni di sicurezza, chiamato **Internet Security Association Key Management Protocol** (**ISAKMP**), che è indipendente dal protocollo per lo scambio delle chiavi;

– un protocollo per lo scambio delle chiavi chiamato **Oakley**, che è basato sull'algoritmo Diffie-Hellman.

2 Codici MAC

Prima di presentare AH e ESP è necessario introdurre un meccanismo di autenticazione che essi impiegano.

In generale, si dice **autenticatore** una stringa di bit (un codice) che dà la prova dell'autenticità (e integrità) di un messaggio. Un tipo di autenticatore già visto è la firma digitale, ma IPSec fa uso di un altro tipo: il **codice MAC**, **Message Authentication Code** (dove "authentication code" può appunto essere tradotto come "autenticatore").

Un codice MAC viene generato mediante una **funzione MAC** C che, dati un messaggio M di lunghezza arbitraria e una chiave K, genera un blocco di dati di dimensioni fisse (e piccole) che dipende da M e K:

$$\text{MAC} = C_K(M)$$

È fondamentale che funzione MAC sia *irreversibile*: avendo il codice MAC non deve essere possibile risalire al messaggio originale. Si può allora osservare che le funzioni MAC assomigliano ad altre due famiglie di algoritmi:

- la cifratura simmetrica, che però è reversibile (per definizione, i messaggi cifrati devono poter essere decifrati) e produce un output lungo quanto il messaggio in input, non di dimensione fissa;
- le funzioni hash, che però non hanno una chiave ma solo un singolo input, il messaggio.

Al fine di utilizzare il MAC per fornire autenticazione e integrità, il mittente A e il destinatario B devono condividere una chiave segreta K_{AB}. Quando vuole spedire un messaggio M, il mittente:

1. dati M e K_{AB} genera il codice $\text{MAC} = C_{K_{AB}}(M)$;
2. spedisce a B il messaggio M e il MAC, tipicamente accodando il MAC al messaggio.

Quando poi il destinatario riceve il messaggio, sia esso M', che potrebbe essere stato modificato da errori di trasmissione o da un attaccante, calcola anch'esso il codice $\text{MAC} = C_{K_{AB}}(M')$, e se esso coincide con il MAC ricevuto da A allora B ottiene la prova che:

- il messaggio è stato effettivamente generato dal mittente A, poiché solo A condivide con B la chiave necessaria per calcolare questo MAC;
- il messaggio è integro, non è stato modificato dopo che il mittente l'ha generato.

In sostanza, il procedimento è simile alla firma digitale, con la principale differenza che qui l'autenticazione si basa su una chiave segreta condivisa, invece che sulla cifratura asimmetrica.

2.1 Funzioni MAC

Esistono due principali famiglie di funzioni MAC standard:

- I **CMAC**, **Cipher-based MAC**, sono basati sugli algoritmi di cifratura simmetrica. Un esempio è CBC-MAC, che al fine di ottenere una funzione irreversibile con output di dimensione fissa:

  1. applica a un messaggio M la cifratura in modalità CBC con chiave K_{AB} e $IV = 0$;
  2. usa come MAC solo l'ultimo blocco cifrato.

  In questo modo, infatti, il MAC dipende da M e da K_{AB}, ha una dimensione fissa pari alla dimensione del blocco ed è irreversibile perché dall'ultimo blocco non si riesce a recuperare l'intero messaggio.

- Gli **HMAC**, **keyed-Hash MAC**, sfruttano una funzione hash H alla quale viene passato come input il messaggio M combinato in qualche modo con la chiave K_{AB}. Una soluzione semplice sarebbe ad esempio concatenare il messaggio con la chiave

  $$\text{MAC} = H(M \parallel K_{AB})$$

  (la concatenazione è indicata dall'operatore $\parallel$). Invece, RFC 2104 definisce un "hash ricorsivo", che viene in sostanza calcolato concatenando al messaggio prima una parte della chiave e poi l'altra parte in due operazioni di hash successive; le due parti della chiave vengono selezionate tramite XOR con due costanti chiamate ipad (inner padding) e opad (outer padding):

  $$\text{MAC} = H\big((K \oplus \text{opad}) \parallel H((K \oplus \text{ipad}) \parallel M)\big)$$

  I tipi di codici HMAC vengono tipicamente indicati scrivendo ad esempio HMAC-SHA1-96, dove SHA1 è la funzione hash e 96 è la lunghezza di troncamento (spesso si usa come MAC non l'intero output della funzione hash, ma solo una parte dei suoi bit).

Riassumendo, i principali tipi di autenticatori sono:

- la firma digitale, basata sulla cifratura asimmetrica;

- i MAC, basati su una chiave segreta condivisa, che si suddividono a loro volta in:
  - CMAC, basati sulla cifratura simmetrica;
  - HMAC, basati sulle funzioni hash.

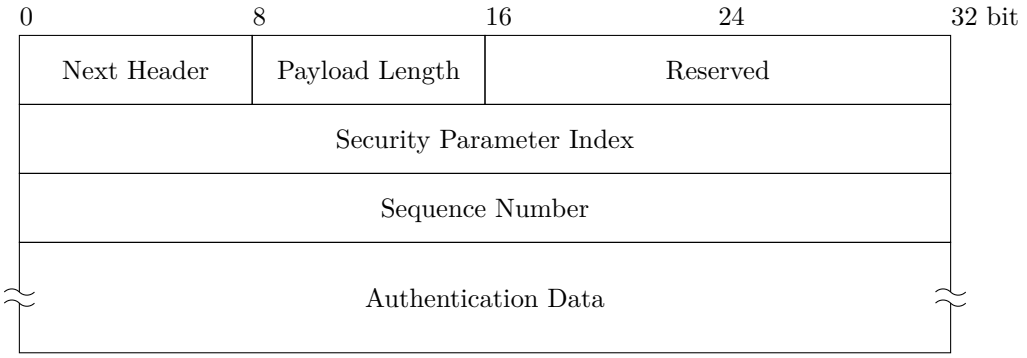
3 Authentication Header

L'Authentication Header (AH) fornisce servizi di:

- *integrità e autenticazione del payload IP*;
- *integrità e autenticazione di alcune informazioni dell'intestazione IP*;
- *anti-replay* — evita che il pacchetto possa essere riutilizzato, ad esempio per attacchi DoS.

Esso si basa sull'utilizzo di un MAC, dunque le due parti devono condividere una chiave segreta.

L'header è composto dai seguenti campi:



- **Next Header**: indica il tipo di intestazione che segue AH (che in IPv4 può essere un altro header IPSec, mentre in IPv6 può essere un qualsiasi header di estensione) oppure il protocollo del pacchetto contenuto nel payload (ad esempio TCP).
- **Payload Length**: indica la lunghezza dell'intero header AH, che varia a seconda del tipo di MAC impiegato.
- *Reserved* (un campo riservato per usi futuri).
- **Security Parameter Index**: l'SPI che identifica l'associazione di sicurezza che fornisce il servizio di autenticazione realizzato da questo header AH.
- **Sequence Number**: un numero di sequenza per il servizio anti-replay.

- **Authentication Data**: il codice MAC, chiamato anche *Integrity Check Value* (ICV). La lunghezza del MAC è variabile, ma deve essere un multiplo di 32 bit.

Gli elementi del pacchetto su cui viene calcolato il MAC, ovvero le informazioni per cui AH fornisce autenticazione e integrità, sono:

- Il *payload del pacchetto IP*, ovvero tutti i dati del pacchetto del protocollo di livello superiore (ad esempio il segmento TCP), e nel caso di IPv6 anche le eventuali intestazioni di estensione inserite nel pacchetto dopo AH.
- L'*intestazione AH*, tranne il campo Authentication Data (l'intera intestazione viene data in input alla funzione MAC, ma il campo Authentication Data viene impostato a 0 perché il suo contenuto sarà appunto il codice MAC da calcolare). È particolarmente importante il fatto che siano protetti i campi SPI e Sequence Number:
  - se un attaccante modificasse l'SPI, il destinatario non saprebbe come processare il pacchetto ricevuto;
  - se un si potesse modificare il Sequence Number, un attaccante sarebbe in grado di aggirare il servizio anti-replay che tale campo implementa.
- I *campi immutabili dell'intestazione IP*, cioè quelli che non cambiano durante la trasmissione, l'instradamento (routing); come l'header AH, tutto l'header IP viene dato in input alla funzione MAC, ma i campi mutabili (ad esempio TTL e checksum) vengono impostati a 0 (infatti, se i campi modificati durante l'instradamento non venissero azzerati prima di computare il MAC, il MAC calcolato dal destinatario sarebbe inevitabilmente diverso da quello calcolato dal mittente, anche in assenza di errori di trasmissione e modifiche intenzionali del pacchetto).

  Tra i campi immutabili che AH protegge sono importanti soprattutto gli *indirizzi IP di mittente e destinatario*, la cui modifica, come già detto, consentirebbe vari tipi di attacchi.

La protezione di questi campi assume una semantica diversa a seconda che AH sia applicato a IPv4 o IPv6 e che sia usato in modalità trasporto o tunnel.

3.1 AH in modalità trasporto

Quando viene applicato in modalità trasporto, cioè inserito tra l'header e il payload IP, AH assume la seguente semantica:

- Nel caso di IPv4 esso fornisce un servizio di autenticazione e integrità delle informazioni immutabili dell'header IP, delle informazioni di AH eccetto (inevitabilmente) il MAC e di tutto il payload.

- Nel caso di IPv6 il servizio fornito dipende dalla posizione in cui si inserisce AH tra gli eventuali altri header di estensione: tutti gli header che vengono dopo AH sono considerati parte del payload, dunque su di essi (insieme all'intestazione di base IPv6, ai campi di AH e al payload vero e proprio, come in IPv4) vengono fornite l'autenticazione e l'integrità. Allora, gli header che servono per il routing (ad esempio l'header di frammentazione) vengono messi prima di AH, in modo che possano essere liberamente modificati lungo il percorso, mentre gli header che servono al destinatario e non cambiano durante l'instradamento vengono inseriti dopo AH in modo che siano autenticati.

3.2 AH in modalità tunnel

In modalità tunnel, l'intero pacchetto originale viene messo come payload di un nuovo pacchetto IP dotato di header AH. Di conseguenza:

- Nel caso di IPv4 si autentica *tutto* il pacchetto originale (ovvero, come sempre, ciò che segue AH), insieme ai campi immutabili del nuovo header IP e ai campi di AH eccetto il MAC.
- Nel caso di IPv6, oltre alle informazioni analoghe a quelle autenticate in IPv4, si autenticano anche gli eventuali extension header inseriti nel pacchetto esterno dopo AH, ma non quelli (per il routing) messi prima di AH. Siccome nella modalità tunnel il destinatario del pacchetto esterno può non coincidere con quello del pacchetto interno, ad esempio può essere un security gateway, dopo AH si possono inserire header autenticati che devono essere processati dal gateway, mentre gli header rivolti al destinatario del pacchetto interno sono appunto inseriti nel pacchetto interno, che viene autenticato per intero (compresi eventuali campi e header mutabili, che però sicuramente non cambiano mentre il pacchetto interno viene trasportato come payload del pacchetto esterno — potrebbero invece cambiare quando ad esempio il gateway di destinazione immette il pacchetto interno nella rete locale, ma a quel punto l'autenticazione è stata già verificata dal gateway e rimossa).