Una Introduzione a IPSEC

In questo articolo vogliamo dare una introduzione sia ideologica che tecnica a IPSEC (ovvero: *IP SECurity*). IPSEC è una famiglia di protocolli dell'IETF che ha lo scopo di rendere più sicure le comunicazioni che utilizzano il protocollo IP (sia la versione 4 che la versione 6).

IPSEC è vasto e complicato, è uno standard pubblico e la maggior parte delle implementazioni da parte dei diversi vendor sono compatibili. Non è un singolo protocollo ma una intera architettura di protocolli che interagiscono tra loro. Nuovi protocolli sono apparsi, quelli originali sono già stati modificati ed aggiornati negli 8 anni di vita di IPSEC. Nuovi protocolli appariranno (basta guardare alle sezioni *ipsec*, *ipseckey* e *ipsp* in http://www.ietf.org/internet-drafts/1id-index.txt) e saranno inseriti nella struttura di IPSEC. Con un semplice aggiornamento del software dovrebbe essere sempre possibile incorporare nella propria implementazione di IPSEC le nuove funzionalità e le correzioni di possibili precedenti problemi.[1]

I protocolli principali di IPSEC sono descritti nei seguenti RFC:

- RFC 2401 : Security Architecture for the Internet Protocol
- RFC 2402 : IP Authentication Header
- RFC 2406 : IP Encapsulating Security Payload (ESP)
- RFC 2409 : The Internet Key Exchange (IKE)

1. A cosa server IPSEC

Come ben noto, i protocolli IP sino alla versione 4, sono stati sviluppati senza un grande riferimento alle problematiche di sicurezza. In particolare, le comunicazioni tra due host tramite IP soffrono dei seguenti principali problemi:

- *Integrità*: sebbene ogni pacchetto IP abbia un controllo di integrità tramite una checksum (CRC), gli algoritmi utilizzati sono molto deboli e non sufficienti a garantire l'integrità dei pacchetti; pertanto il ricevente non può essere sicuro che il pacchetto che riceve sia identico a quello spedito e che non sia stato modificato, volontariamente o casualmente, durante il percorso.
- *Autenticità*: chiunque può inviare un pacchetto ponendo come mittente (*sorgente*) il numero IP di un altro host (questo è di solito chiamato *spoofing*) pertanto il ricevente (destinatario), utilizzando solo i dati presenti negli header IP, non può essere sicuro di chi sia il mittente del pacchetto.
- *Confidenzialità*: tutto il pacchetto, ed i dati (*payload*) in esso trasportati, sono in chiaro, pertanto

1 Questo non vuol dire che IPSEC sia implementato solo in software, molti algoritmi crittografici possono essere implementati in hardware, in specifiche ASIC ad esempio.

chiunque possa intercettare (detto anche *sniffing*) il pacchetto può leggerne il contenuto. Ovviamente, dal punto di vista della sicurezza, questo è un disastro.

IPSEC risolve questi tre problemi facendo in modo che le comunicazioni fra due host possano rispondere a criteri di integrità, autenticità ed opzionalmente anche confidenzialità. Poiché IPSEC è stato progettato in modo modulare ed è completamente configurabile, permette di specificare i livelli di integrità, autenticità e nel caso confidenzialità richiesti. E' chiaro che più alto è il livello di sicurezza richiesto, maggiori è l'utilizzo da parte di IPSEC delle risorse dei sistemi e più lente sono le comunicazioni (e questa è una delle ragioni per cui la sicurezza non era stata inserita nei protocolli IP originali).

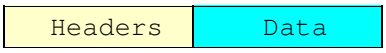
IPSEC viene utilizzato per la comunicazione tra due host, questi possono essere due elaboratori, due router, due firewall, due concentratori di VPN-IPSEC ecc., oppure due a scelta tra questi. In ogni caso IPSEC collega **due** host (questo è detto traffico *unicast* in opposizione al traffico *multicast* da un host a molti host riceventi) e si può utilizzare solo con traffico IP (nel caso, altri tipi di traffico devono essere prima incapsulati in IP). Inoltre bisogna distinguere i due seguenti *modi* di IPSEC:

1. il caso in cui IPSEC è utilizzato fra due elaboratori che sono i punti estremi del traffico, detto *modo di Trasporto (Transport mode)*
2. tutti gli altri casi (router-router, elaboratore-router ecc.) ove almeno uno dei due estremi non è il destinatario finale del traffico (*Gateway*), detto *modo di Tunnel (Tunnel mode)*; in questo caso il Gateway invia/riceve il traffico a/da il destinatario finale utilizzando il protocollo IP usuale.

La differenza logica più significativa tra i due modi è che nel secondo caso nel pacchetto trasportato da IPSEC ci deve essere sia l'indicazione del Gateway che del destinatario finale non-IPSEC, mentre nel primo caso quest'ultimo non è necessario.

2. La soluzione di IPSEC

La soluzione adottata da IPSEC per rendere sicuro IPv4 è di creare un nuovo pacchetto IP nel quale viene inserito il pacchetto IP originale, questo è detto *incapsulare* il pacchetto originale. Ricordiamo che un pacchetto IP è fatto da due componenti:



1. gli Header IP che specificano il numero IP del mittente e del destinatario più vari altri parametri come il TTL (Time-To-Live), Type-of-Service, Flags IP o tipo/numero di frammento, ed il *Protocollo IP* di livello superiore (Transport) quali TCP, UDP, ICMP ecc.
2. i Dati trasportati, all'interno dei quali possono esservi anche altri Header di protocolli di livello

superiore.[2]

Quando due host comunicano via IPSEC, i pacchetti originali vengono inseriti da IPSEC all'interno di nuovi pacchetti IP (alternativamente i pacchetti IP vengono modificati con l'inserzione degli Header di IPSEC) prima di essere inviati sulla rete. Nei nuovi pacchetti IP compaiono tutti i dati dei pacchetti originali, con il payload originale cifrato se ciò è stato richiesto, e con l'aggiunta degli Header IPSEC che permettono al partner di verificare l'autenticità ed integrità del pacchetto. Come vedremo più avanti, i nuovi pacchetti IP indicano come protocollo di livello Transport AH (51) oppure ESP (50), invece dei TCP, UDP ecc. originari, così che tutto il traffico dati fra i due host a livello IP risulta essere solo di pacchetti di tipo AH o ESP.[3]

E' ovvio che incapsulando i pacchetti in nuovi pacchetti IP ed aggiungendo gli Header IPSEC, il pacchetto finale inviato sulla rete risulta più lungo. Questo può creare dei problemi poiché vi è di norma un limite alla dimensione dei singoli datagrammi che si possono inviare sulla rete. Ad esempio il comunissimo Ethernet ammette *frame* di lunghezza massima di 1500 Byte. Se un firewall riceve un frame di 1500 Byte e lo deve inviare in un tunnel IPSEC, l'incapsulamento ed il limite sulla lunghezza dei frame obbligano il firewall a spezzare il pacchetto originario in più parti, detti *frammenti* e questo può creare problemi a chi riceve i dati. Per evitare ciò, è sempre consigliato su tutti gli elaboratori connessi a reti Ethernet e a tunnel IPSEC, di ridurre l'MTU (Maximum Transfer Unit) delle interfacce Ethernet ad esempio a 1400 Byte.

Tutte le comunicazioni fra i due host connessi da un tunnel IPSEC ed identificati dal loro numero IPv4, devono passare via IPSEC: pertanto se un host riceve un pacchetto IPv4 non-IPSEC con numero IP sorgente di un partner IPSEC, lo deve scartare. In questo modo viene a formarsi un tunnel virtuale tra i due host, detto anche Virtual Private Network (VPN), tutte le comunicazioni IP tra di loro devono essere IPSEC (vedi Figura 1 e Figura 2). Non solo, ogni tunnel è unidirezionale, perciò per comunicare i due host devono stabilire due tunnel, uno per ogni direzione. Visto che IPSEC collega direttamente due e solo due host, per ogni coppia di host che comunicano con IPSEC bisogna stabilire una coppia di tunnel.

Poiché IPSEC è modulare ed altamente configurabile, è immediatamente chiaro a tutti che può facilmente diventare un vero problema gestire in grandi realtà la configurazione dei tunnel IPSEC dato il numero di tunnel e, come vedremo nel seguito, il numero di parametri necessari a specificare ogni tunnel.

---

2 IP è un protocollo al livello Network nello stack ISO/OSI, mentre TCP, UDP, ICMP ecc. sono al livello superiore detto Transport.

3 E' stata recentemente introdotta una eccezione a questa regola: è il NAT-Traversal che consente di incapsulare ulteriormente i pacchetti IPSEC in pacchetti UDP per poter attraversare router o firewall che applicano NAT; questo protocollo è ancora allo stato di Internet Draft, ma è già stato implementato da molti Vendor.

3. I principi di funzionamento di IPSEC

Supponiamo di avere una coppia di host con uno stack TCP/IP in cui sia implementato IPSEC e di avere configurato entrambi gli host ad utilizzare IPSEC per comunicare tra loro. Il primo host, denotiamolo con A, vuole inviare un pacchetto (datagramma) IPv4 a B. IPSEC si accorge che questo pacchetto deve passare attraverso un tunnel IPSEC (oppure non essere inviato per nulla) e quindi si attiva. Prima di poter inviare il pacchetto, IPSEC deve creare i due *tunnel*, e per far questo è necessaria una complicata fase di inizializzazione. Vi sono fondamentalmente due motivi che richiedono questa fase di inizializzazione:

1. Come abbiamo visto, l'autenticità degli host non può essere stabilita solo con gli indirizzi IP che sono facilmente *spoofabili*; pertanto i due host si devono scambiare delle credenziali stabilite a priori, le più semplici sono delle parole chiave (o password) note solo ai due host, ma IPSEC non obbliga ad utilizzare un solo protocollo di autenticazione per cui vi è una scelta fra metodi più o meno sicuri. E' da notare che le credenziali non possono di norma essere scambiate in chiaro, altrimenti un terzo host in ascolto potrebbe ottenerle e poi impersonare uno dei due host, quindi prima della fase di autenticazione è necessario stabilire un canale sicuro tra i due host.
2. Una volta che l'autenticazione ha successo, i due host devono concordare come trasferire i pacchetti in modo che le richieste di autenticità, integrità e nel caso anche confidenzialità siano soddisfatte; questo è meno facile di quanto si possa pensare, in quanto bisogna da esempio garantire l'autenticità di ogni pacchetto trasferito, senza però dover ri-eseguire la lenta procedura di autenticazione per ogni pacchetto; bisogna evitare gli attacchi *man-in-the-middle*, ove un terzo host si inserisce nella comunicazione prendendo le parti alternativamente di uno dei due host, ed anche evitare i *replay-attack* nei quali un terzo host che è in grado di ascoltare la comunicazione fra i due, ri-invia ad un host un pacchetto già inviato dall'altro,[4] e così via.

La procedura di inizializzazione di IPSEC è detta IKE (Internet Key Exchange), di cui discuteremo alcuni dettagli più avanti. Una volta terminata la procedura positivamente, sono create due *Security Association (SA)*, che sono la raccolta di tutti i parametri che caratterizzano i due tunnel IPSEC che collegano i due host; in pratica le SA identificano univocamente i tunnel. A questo punto finalmente l'host A, utilizzando l'opportuna SA, trasforma il pacchetto originario in un pacchetto IPSEC e lo invia a B. B, avendo anch'esso le due SA, può trasformare al contrario il pacchetto IPSEC nel pacchetto originale. Poi, in modo simile B può rispondere ad A. Poiché IKE è complicato, capita sovente che il primo pacchetto inviato in un tunnel IPSEC arrivi con un ritardo anche di qualche secondo, il tempo necessario per eseguire tutta la procedura di inizializzazione.

Purtroppo non è finita qui. Infatti le SA hanno una breve durata nel tempo e nella quantità di dati

4 Analizzando le risposte dell'host che riceve i doppi pacchetti è possibile in alcuni casi violare le chiavi crittografiche utilizzate.

per i quali sono utilizzate. Il motivo di ciò è che se si riutilizzano gli stessi parametri crittografici per codificare molti dati, si rende più facile rompere la cifratura ed ottenere le chiavi (ad esempio questo è uno dei problemi fondamentali del WEP utilizzato dalle schede WiFi per le Wireless-LAN). Pertanto periodicamente le SA devono essere rigenerate, cambiando tutti i parametri sensibili dei tunnel (fondamentalmente le chiavi simmetriche utilizzate dai vari algoritmi).

Poiché la fase di inizializzazione è lenta e complicata, abbiamo quindi che un tunnel IPSEC viene creato e rimane attivo per un certo periodo di tempo, sia che vi sia traffico che non vi sia. Allo scadere del tempo, se non vi è traffico il tunnel viene chiuso altrimenti le SA vengono rigenerate. Questo comportamento è indipendente dal tipo di traffico che vi passa, siano essi pacchetti TCP, UDP, ICMP o altro.

La tecnica di base utilizzata per garantire l'integrità, l'autenticità e la confidenzialità dei pacchetti è la crittografia. IPSEC utilizza nelle proprie varie fasi e per i diversi compiti, diverse classi di algoritmi e protocolli crittografici. In questo articolo non entreremo nei dettagli degli algoritmi crittografici utilizzabili in IPSEC. Le principali famiglie di algoritmi utilizzate da IPSEC sono:

1. Message Authentication Code (MAC): questi algoritmi generano, a partire da una stringa di dati di solito di lunghezza arbitraria, una stringa di lunghezza fissa che ha (tra altre) la proprietà di essere praticamente unica per quella stringa, in altre parole un minimo cambiamento nella stringa di dati risulta in una grande modifica del MAC; sono utilizzati per verificare l'integrità di ogni singolo pacchetto; algoritmi di questo tipo sono MD5 e SHA1.
2. Algoritmi a Chiave Pubblica e Privata (o Asimmetrici): questi algoritmi al giorno d'oggi molto famosi, sono utilizzati prevalentemente nella fase di autenticazione in quanto richiedono molte risorse e risultano quindi lenti, il più famoso è l'RSA; una variante di particolare importanza è il protocollo di Diffie-Hellman che permette a due entità di scambiarsi dei dati (in particolare dei numeri primi sufficientemente grandi) e con questi costruire una chiave segreta; i dati scambiati sono tali che oggi è computazionalmente impossibile costruire la chiave segreta utilizzando solo i dati che sono scambiati pubblicamente.
3. Algoritmi a Chiave Simmetrica: questi sono i più antichi e noti algoritmi di cifratura dei dati, sono veloci e quindi adatti a cifrare i dati scambiati fra gli host; richiedono che entrambi gli host siano a conoscenza della stessa chiave segreta che è utilizzata sia per cifrare che per decifrare i dati;[5] i più conosciuti sono DES, 3-DES e AES.

Terminata la fase iniziale della creazione dei tunnel (IKE), finalmente IPSEC incomincia a scambiare i dati fra i due host. Come già detto lo può fare con uno di due protocolli:

5 In alcuni casi la chiave per cifrare è diversa da quella usata per decifrare, ma è semplice ottenere una chiave una volta che si sia in possesso dell'altra, al contrario di quanto succede per gli algoritmi a chiavi Asimmetriche.

Authentication Header (AH) e Encapsulating Security Payload (ESP).[6] Le differenze principali fra i due protocolli sono:

1. AH non offre la confidenzialità, ovvero i dati (payload) non sono cifrati, mentre ESP permette di cifrare i dati.
2. AH garantisce l'integrità anche degli Header IP del pacchetto (con l'eccezione di alcune parti mutabili come discusso più avanti), mentre ESP garantisce l'integrità solo del payload del pacchetto IP.

Mettendo insieme i 2 modi di trasmissione (Tunnel e Transport) con i 2 Protocolli di IPSEC (AH e ESP), si ottengono i 4 principali tipi di pacchetti di dati creati e trasportati da IPSEC indicati in Figura 3.

4. Esempi di utilizzo di IPSEC

Non vogliamo qui dilungarci su come progettare ed implementare VPN IPSEC discutendo delle varie possibili topologie dei tunnel, dal *Single Hub and Spokes* al *Full Mesh*, ai problemi dei *Mobile User* quali gli *Split Tunnel*, oppure discutere delle applicazioni possibili di questa tecnologia. Consideriamo quindi solo le più tipiche implementazioni.

1. Tunnel Gateway-Gateway: questi sono i tunnel più comuni di solito realizzati fra router o firewall e permettono di connettere, usualmente in modalità Tunnel+ESP, sedi remote di aziende o reti locali diverse, attraverso Internet o reti non fidate. In pratica le due reti locali *dietro* i Gateway possono comunicare direttamente come se le reti che vengono attraversate non esistessero in quanto i pacchetti sono trasportati da IPSEC nella tratta pericolosa. Usando il modo Tunnel, obbligatorio per i Gateway, ed ESP, il contenuto dei pacchetti IPSEC è cifrato ed il traffico dall'esterno appare essere solo tra i due Gateway. Un attaccante che possa ottenere copia dei pacchetti IPSEC scambiati dai Gateway riesce solo a scoprire che si tratta di traffico IPSEC ESP fra i Gateway e quindi molto probabilmente tra due reti dietro agli stessi, ma non riesce ad ottenere alcuna informazione sulle reti dietro i Gateway e sul contenuto dei pacchetti scambiati. E' chiaro che in questo caso la sicurezza risiede nella sicurezza del Gateway, che un attaccante cercherà di penetrare per avere accesso alle informazioni.
2. Tunnel Elaboratore-Gateway: anche questi tunnel sono comuni e sono spesso utilizzati per permettere la connessione alla rete aziendale od ad una rete locale, di singoli elaboratori che risiedono su reti remote. I tipici esempi sono i tele-lavoratori che si collegano da casa tramite un qualunque ISP alla rete aziendale, od i *Road-Warriors* ovvero i lavoratori mobili che hanno necessità di accedere alla rete aziendale da alberghi o luoghi pubblici in qualunque località. Di nuovo in questi casi viene adottata usualmente la modalità Tunnel+ESP con l'aggiunta di

6 E' anche possibile utilizzare entrambi i protocolli incapsulando l'uno nell'altro.

modalità forti di autenticazione, poiché tra l'altro il numero IP sorgente dell'host mobile è variabile. In questo caso di solito il punto più debole per la sicurezza è l'host remoto che non è sotto il diretto controllo dell'amministratore e che accendendo ad internet od altre reti può trasformarsi in un veicolo di ingresso indiretto attraverso il tunnel IPSEC alla rete protetta dal Gateway.

3. Tunnel Elaboratore-Elaboratore: questi tunnel sono meno frequenti poiché spesso si preferisce utilizzare soluzioni più specializzate alla soluzione dello specifico problema che la soluzione generale offerta da IPSEC. Questi tunnel comunque permettono a due elaboratori di comunicare direttamente in modo sicuro attraverso qualunque rete non sicura. Un esempio potrebbe essere un gruppo di server DNS pubblici, uno principale e altri secondari, che comunicano tra loro, ad esempio per effettuare i *zone transfer*, solo via IPSEC, mentre comunicano con tutto il resto di internet utilizzando il normale IPv4. In alcuni casi questa modalità viene adottata per rendere sicura la comunicazione tra due Gateway che all'interno del tunnel IPSEC inseriscono un altro tunnel tra di loro, ad esempio GRE, per poter trasportare traffico non di tipo IP unicast.

Bisogna comunque sottolineare che IPSEC garantisce l'integrità, autenticità e confidenzialità delle comunicazioni, non della sicurezza degli host che sono gli estremi dei tunnel. Pertanto non è sufficiente adottare IPSEC per avere comunicazioni sicure, bisogna anche che gli host agli estremi dei tunnel siano ragionevolmente sicuri.

5. IKE

Passiamo ora a dare qualche indicazione su IKE, la parte più complessa di IPSEC. Per rendersi conto della complessità di IKE basta cercare di leggere il relativo RFC, oppure guardare alla storia della sua nascita. Prima di IKE furono creati due protocolli per l'autenticazione e la creazione delle chiavi di IPSEC, Photuris (definito in RFC 2522) e SKIP (definito in http://skip.incog.com/inet-95.ps). Purtroppo per motivi politici entrambi vennero abbandonati e si decise di creare IKE basandosi sull'Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP non è un vero protocollo ma una struttura di riferimento, e l'IETF sviluppò OAKLEY basandosi su questo. IKE prese molto da OAKLEY (anche se i due sono praticamente coetanei) ma incorporò anche un altro protocollo, SKEME. Questa breve cronistoria è complicata, ma IKE lo è ancora di più. Malgrado ciò, al momento IKE sembra essere un protocollo robusto, anche se ha molti detrattori proprio a causa della sua complessità.

Come abbiamo detto, gli scopi di IKE sono due: autenticare vicendevolmente i due host e stabilire i protocolli e le chiavi segrete da utilizzare per trasferire i dati. IKE utilizza di norma la porta UDP 500 e quindi il traffico su questa porta deve essere abilitato per poter stabilire le connessioni

IPSEC.

IKE agisce in due fasi successive:

1. la prima fase fornisce la creazione di chiavi di sessione per se stesso e la Fase 2 (ISAKMP Security Association) e l'autenticazione vicendevole degli host
2. nella seconda fase gli host creano le SA di IPSEC che saranno utilizzate per scambiarsi i dati.

IKE Fase 1

A sua volta la Fase 1 di IKE può essere in due modi: *Main Mode* e *Aggressive Mode*.

Main Mode

Descriviamo per primo il Main Mode che è più semplice. In questa modalità sono garantite la protezione dell'identità degli host ed offre maggiori possibilità di negoziazione dei parametri. Per prima cosa viene creato un canale sicuro, quindi cifrato, poi su questo canale viene fatta l'autenticazione. I due host si scambiano 6 pacchetti:

1. nel primo pacchetto (UDP verso la porta 500) l'iniziatore della sessione IPSEC invia al ricevente un cookie e la lista di algoritmi crittografici sia per l'autenticazione (ed il metodo di autenticazione) che per l'integrità e confidenzialità, questi formano la ISAKMP SA
2. il ricevente risponde anch'esso con un proprio cookie e la propria lista di algoritmi crittografici; se vi è accordo sulle due liste è possibile proseguire
3. l'iniziatore manda ora i propri parametri di Diffie-Hellman
4. il ricevente risponde con i propri parametri di Diffie-Hellman; a questo punto i due host hanno definito tutti i parametri della ISAKMP SA poiché hanno stabilito sia gli algoritmi crittografici che le chiavi, costruite con lo scambio di Diffie-Hellman[7], per poterli usare; d'ora in avanti possono quindi comunicare scambiandosi dati cifrati
5. l'iniziatore manda le proprie credenziali al ricevente
6. il ricevente manda le proprie credenziali all'iniziatore; a questo punto entrambi gli host sono sicuri dell'identità dell'altro.

In questa sede non descriviamo il funzionamento dell'algoritmo di Diffie-Hellman per la generazione della chiave simmetrica segreta. I principali metodi di autenticazione sono tramite *shared key*, ovvero una chiave segreta nota ad entrambi gli host, oppure i certificati digitali. I certificati digitali possono essere usati in modi diversi, ad esempio verificando che il certificato sia firmato da una CA nota, oppure che il certificato sia presente in una lista di certificati ammessi presente sugli host, e così via. Il Main Mode viene di solito adottato per la creazione di tunnel

7 La costruzione della chiave simmetrica utilizzando Diffie-Hellman richiede spesso notevoli risorse di sistema, in particolare tempo macchina.

IPSEC tra due Gateway.

Aggressive Mode

L'Aggressive Mode realizza le stesse cose del Main Mode scambiando solo 3 pacchetti ma senza garantire la protezione dell'identità degli host. La sequenza è

1. L'iniziatore manda un pacchetto con i dati contenuti nei pacchetti 1 e 3 del Main Mode
2. Il ricevente risponde con un pacchetto con i dati contenuti nei pacchetti 2, 4 e 6 del Main Mode; in questo scambio le credenziali del ricevente sono mandante in chiaro all'iniziatore
3. L'iniziatore manda i dati del pacchetto 5 del Main Mode.

L'Aggressive Mode è ovviamente più rapido del Main Mode, anche se meno sicuro, e viene di solito utilizzato per la creazione di tunnel IPSEC tra un elaboratore ed un Gateway.

IKE Fase 2

Al termine della prima fase si è stabilita la ISAKMP SA, quindi esiste un canale sicuro protetto dagli algoritmi crittografici scelti, inoltre i due host si sono autenticati vicendevolmente. Utilizzando il canale sicuro ISAKMP i due host possono ora negoziare la IPSEC SA, ovvero l'insieme dei parametri che permettono di costruire il tunnel IPSEC. Questa fase è detta *Quick Mode* o Fase 2. A sua volta la Fase 2 si divide in due modalità:

1. *Basic Quick Mode*: in questa modalità non vengono generate nuove chiavi segrete ma vengono riutilizzate anche per la IPSEC SA quelle generate nella prima fase
2. *Perfect Forward Secrecy*: in questa modalità vengono generate nuove chiavi segrete per gli algoritmi utilizzati nella IPSEC SA, pertanto è necessario effettuare un nuovo scambio di Diffie-Hellman; questo modo è più sicuro ma richiede ovviamente più tempo e più risorse.

La negoziazione porta alla creazione delle IPSEC SA. Una IPSEC SA è identificata da 3 parametri:

1. *Security Parameters Index (SPI)*: un numero che identifica la SA utilizzata e che viene posto negli Header del pacchetto IPSEC; all'arrivo di un pacchetto IPSEC un host associa il pacchetto ad una propria SA attiva basandosi su questo numero
2. *Destination Address*: l'indirizzo IP dell'altro estremo del tunnel IPSEC
3. *Security Protocol Identifier*: stabilisce se il tunnel adotta AH o ESP.

In particolare i parametri rilevanti contenuti in una SA sono:

- il *Security Protocol*, ESP o AH
- l'*Authentication Algorithm*, di norma SHA-1 o MD5, utilizzati sia per autenticare che per verificare l'integrità dei pacchetti
- l'*Encryption Algorithm*, solo nel caso di ESP ed usualmente uno tra DES, 3-DES e AES, anche

se altri algoritmi sono talvolta utilizzati

- Gli indirizzi IP degli host di interesse, ad esempio gli estremi del tunnel e nel caso di gateway gli indirizzi delle reti dietro i gateway che comunicano tramite il tunnel IPSEC.

La Fase 2 di solito si svolge con lo scambio di 3 pacchetti: nel primo pacchetto uno dei due host invia la propria proposta per la IPSEC SA (ovvero tutti i parametri che la caratterizzano) incluso lo SPI di un tunnel; nel secondo pacchetto l'altro host accetta la proposta della IPSEC SA inviata dal primo host ed invia la propria proposta includendo lo SPI del secondo tunnel; nel terzo pacchetto il primo host accetta la proposta fatta dal secondo host.

Al termine della Fase 2 entrambi gli host hanno tutti i parametri necessari, incluse le chiavi segrete, per poter attivare i tunnel IPSEC ed incominciare a scambiarsi pacchetti.

Alcune precisazioni sono d'obbligo. La negoziazione delle SA, sia la ISAKMP che la IPSEC, è più rigida di quello che si può immaginare. A livello di configurazione un amministratore prepara una proposta di SA elencando i protocolli, gli algoritmi (incluse le lunghezze delle chiavi) ed i parametri che la specificano: ad esempio può specificare di utilizzare ESP con MD5, 3DES ecc. L'amministratore può preparare più di una proposta di SA, una seconda potrebbe avere ESP, SHA-1 e AES ad esempio. Se sui due host gli amministratori hanno preparato due proposte di SA speculari[8], allora questa viene adottata; non è possibile fare intersezioni fra proposte, cioè se sul secondo host l'amministratore ha preparato solo una proposta di SA con ESP, SHA-1 e 3DES la negoziazione fallisce.

Come abbiamo visto, l'amministratore deve preparare le proposte per due diversi tipi di SA, la ISAKMP SA e la IPSEC SA. I parametri richiesti per i due tipi di SA sono praticamente gli stessi, e spesso il fatto che debbano essere inseriti *due volte* in posti diversi genera confusione negli amministratori. Cerchiamo comunque di fare un riassunto dei principali dati necessari per la configurazione di un tunnel IPSEC:

1. il protocollo scelto, ESP o AH
2. il modo scelto, tunnel o transport
3. gli indirizzi IP dei gateway e/o delle reti dietro di essi
4. il metodo di autenticazione dei due host e le credenziali
5. le classi di Diffie-Hellman, di solito la classe 2
6. il modo della Fase 1 di IKE, main o aggressive
7. il modo della Fase 2 di IKE, PFS o Basic Quick
8. l'algoritmo MAC, ad esempio MD5 o SHA-1
9. per ESP l'algoritmo di cifratura, ad esempio 3DES o AES.

Di norma vi sono molti altri parametri configurabili, ad esempio il tempo di vita di una SA ecc.,

8 Alcuni parametri, quali gli indirizzi IP, sono specificati specularmente sui due host.

ma di solito i default dati dai vendor sono ragionevoli.
Infine è anche possibile attivare tunnel IPSEC senza utilizzare IKE, questo è chiamato *Manual Keying* ma in generale non è consigliato adottarlo, sia perché la configurazione di IPSEC diventa molto più lunga dovendo specificare ad esempio tutte le chiavi segrete, sia perché la protezione contro gli attacchi di tipo *replay*, la rigenerazione automatica e periodica delle chiavi ecc. non sono più disponibili.

Una volta terminata la fase di inizializzazione tramite IKE, i due host si scambiano pacchetti utilizzando IPSEC. In pratica ogni qual volta lo stack TCP/IP riceve un pacchetto che deve passare nel tunnel IPSEC, il pacchetto è trasformato da IPSEC secondo la SA che controlla il tunnel. La decisione sull'ammissione o meno del pacchetto al tunnel dipende fondamentalmente dagli indirizzi IP specificati nell'Header IP del pacchetto originale. Una volta creato un pacchetto IPSEC per un particolare tunnel identificato dallo SPI, dal numero IP dell'altro host e dal protocollo ESP o AH, il pacchetto è spedito all'altro host. Quando il secondo host riceve il pacchetto individua la SA IPSEC dai tre parametri, SPI, numero IP e protocollo ESP o AH, ed utilizza questa SA per ri-ottenere il pacchetto originale. La struttura dei pacchetti IPSEC è leggermente diversa per il protocollo AH ed ESP visto il diverso tipo di protezione che i due protocolli offrono.

6. AH

AH garantisce l'integrità e autenticità, ma non la confidenzialità, di tutto il pacchetto esclusi solo degli Header IP detti Mutabili. Questi sono dei campi nell'header IP che possono essere modificati dai router attraverso i quali passa il pacchetto IPSEC. Gli Header definiti Mutabili da AH sono

1. ToS - Type of Service
2. Flags
3. Fragment Offset
4. TTL - Time To Live
5. Header Checksum.

I campi fondamentali nell'Header AH (si vedano le Figure 3 e 4) sono lo SPI che abbiamo già descritto, un *Sequence Number* che è continuamente incrementato ad ogni pacchetto inviato e serve a proteggersi contro i *replay attacks*, e l'*Authentication Data*, un campo di dimensione variabile che contiene o un Integrity Check Value (ICV) od un Message Authentication Code (MAC) che garantisce l'integrità e l'autenticità di tutto il pacchetto eccetto i campi Mutabili dell'Header IP appena menzionati.

7. ESP

Un pacchetto ESP ha una struttura leggermente diversa da un pacchetto AH. Innanzitutto la parte del pacchetto originale (tutto il pacchetto nel caso di modo Tunnel) inclusa nel *Payload* con la possibile aggiunta di *Padding* è cifrata (si vedano le Figure 3 e 5). L'Header ESP contiene, come per AH, lo SPI ed il *Sequence Number*. L'*Authentication Data* è posto come ultimo Trailer e non copre gli Header IP come invece fa AH. Quindi ESP rispetto ad AH autentica una porzione inferiore del pacchetto finale, ma cifra il Payload.

8. Conclusioni

In questo articolo abbiamo presentato la struttura e qualche dettaglio tecnico di IPSEC. IPSEC è una architettura di protocolli che rende sicuro il trasporto di informazioni su di una rete informatica utilizzando IP. IPSEC adotta pertanto un approccio generale, non specialistico, che garantisce che qualunque tipo di informazione, servizio ecc. trasportato su di una rete IP sia sicuro. La generalità dell'approccio ha come quasi inevitabile conseguenza la grande estensione e complessità, visto che si vuole offrire una soluzione utilizzabile in ogni situazione. Similmente, il numero di parametri è alto e la configurazione di IPSEC è relativamente complicata quale che sia la GUI o linea di comando utilizzata.

Tutto ciò, trattandosi di Sicurezza, porta immediatamente ai seguenti potenziali problemi:

1. il numero dei protocolli, la loro estensione e complessità e le problematiche di interfacciamento fra gli stessi, rende più facile la possibilità che vi siano zone non coperte, minime aperture che permettano l'intrusione e la violazione dei requisiti di sicurezza. E' molto difficile, data l'estensione e complessità di IPSEC, poter dare la garanzia della consistenza formale, progettuale ed algoritmica di tutta l'architettura, includendo tutte le possibili interazioni fra tutti i componenti dell'architettura.
2. Sempre l'estensione e complessità di IPSEC rende più difficile evitare errori in fase di implementazione. Le linee di codice richieste sono molte, i diversi moduli devono soddisfare tutte le specifiche ed interagire correttamente fra di loro e con l'architettura che li ospita, ovvero il Sistema Operativo ed il suo stack TCP/IP nativo. L'Audit completo di una implementazione IPSEC non è cosa del tutto banale.
3. Infine anche gli utilizzatori, ovvero gli amministratori che devono configurare i tunnel IPSEC, a causa della complessità possono essere indotti in errori nelle configurazioni, adottando configurazioni che non rispecchiano i requisiti di sicurezza richiesti dalle politiche di sicurezza stabilite.

L'esperienza degli ultimi anni mostra che IPSEC è stato progettato abbastanza bene, poiché non

sono stati scoperti dei particolari problemi a livello progettuale, e che l'organizzazione modulare del protocollo ha anche aiutato gli implementatori a scrivere del codice con un numero ragionevolmente basso di errori. In altre parole, le implementazioni di IPSEC oggi sul mercato si sono dimostrare sinora robuste e ragionevolmente sicure. Quello che sicuramente è un problema è invece il fattore umano indicato nel punto 3: non è raro purtroppo che le configurazioni dei tunnel IPSEC scelte dagli amministratori diano un livello di sicurezza inferiore a quello che dovrebbe essere.

(L'autore ringrazia Fabrizio Croce, Country Manager Italy WatchGuard Technologies, fabrizio.croce@watchguard.com, per il contributo dato alla realizzazione di questo articolo.)

Andrea Pasquinucci
Consulente di Sicurezza Informatica
pasquinucci@ucci.it

Riferimenti Bibliografici:

[1] I riferimenti di base sono gli RFC dei protocolli, in particolare i numeri dal 2401 al 2412 ed i draft draft-ietf-ipsec-udp-encaps-06.txt, draft-ietf-ipsec-nat-t-ike-06.txt, anche l'IPSEC Charter alla pagina http://www.ietf.org/html.charters/ipsec-charter.html

[2] Per l'interoperabilità delle implementazioni di IPSEC si può vedere ad esempio http://www.vpnc.org/ ed http://www.icsalabs.com/

[3] Whitfield Diffie e Martin E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, 22 (1976) 644-654.

[4] Ido Dubrawsky, *Configuring IPsec/IKE on Solaris Part One*, http://www.securityfocus.com/infocus/1616

[5] William Stallings, *Cryptography and Network Security, Principles and Practice*, Prentice Hall, Upper Saddle River, NJ. 1999.

[6] Charlie Kaufman, Radia Perlman, Mike Speciner, *Network Security, Private Communication in a Public World*, Prentice Hall, Upper Saddle River, NJ. 2002.

[7] Pascal Le Digol, *Introduction to IPSEC VPN*, WatchGuard Corporate Presentation 2003.

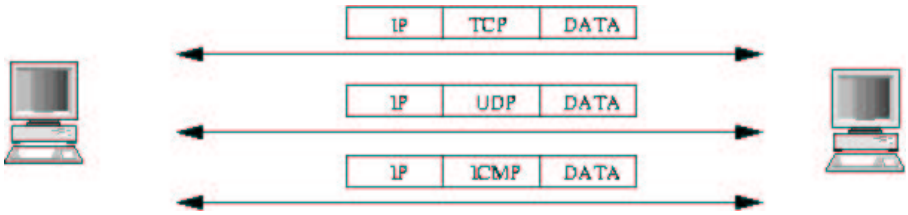


Figura 1. Comunicazioni IP dirette tra due host

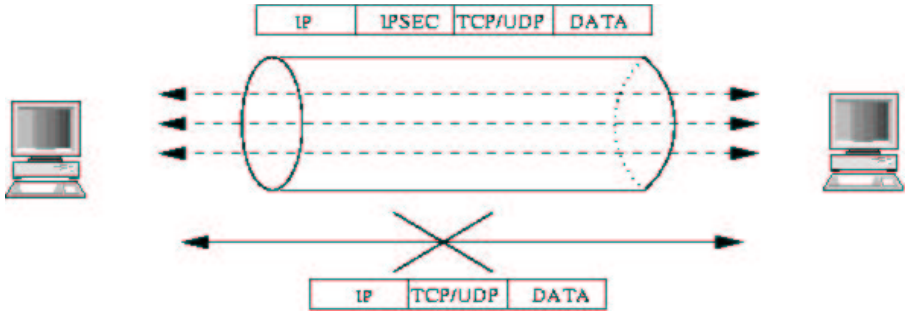


Figura 2. Comunicazioni IP mediate da un tunnel IPSEC

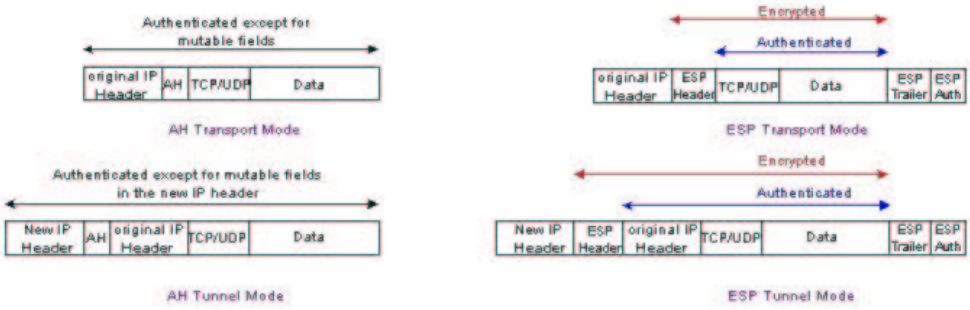


Figura 3. I quattro tipi di pacchetti dati di IPSEC (da [4])

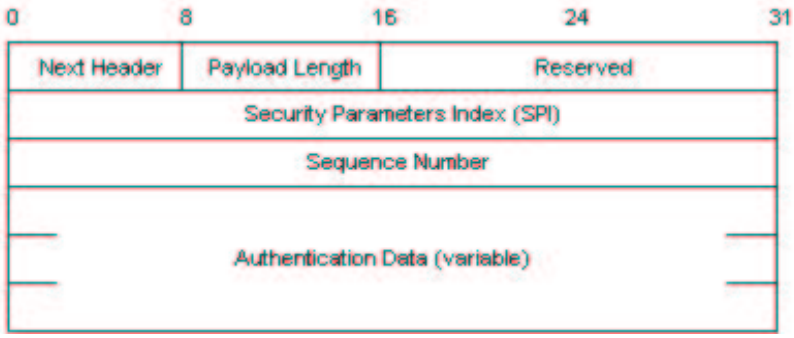


Figura 4. l'Header AH in dettaglio (da [4])

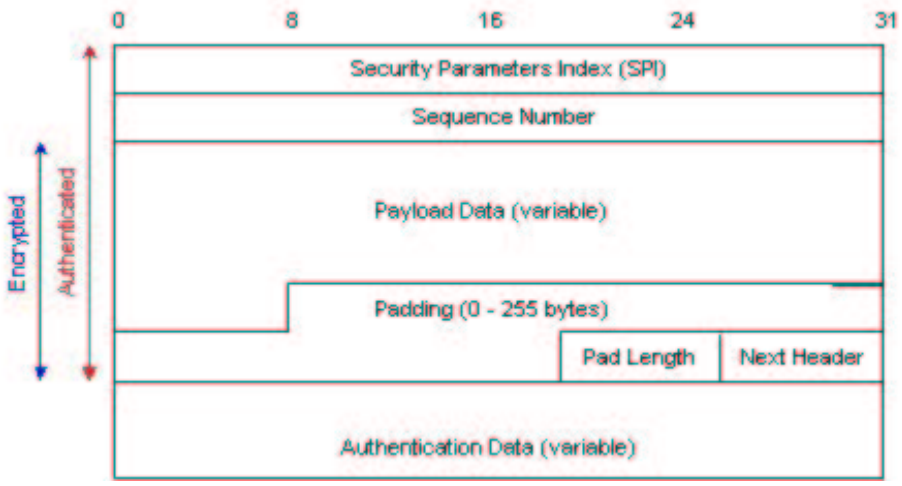


Figura 5. Il pacchetto ESP dall'Header *ESP Header* al Trailer *ESP Auth* (da [4])