CAPITOLO 8: Network Security

**8.1 Introduzione**

Comunicare in modo sicuro tra due utenti vuol dire che riescono a parlare tra loro senza che una terza persona possa intercettare i loro messaggi e in modo che solo i due utenti possano capirsi tra loro. Per la comunicazione sicura ci sono diverse proprietà:

-Riservatezza: solo il mittente e il destinatario dovrebbero comprendere il messaggio trasmesso e per far ciò questo va cifrato per renderlo incomprensibile agli altri.

-Integrità del messaggio: il messaggio non deve subire manipolazioni o alterazioni e per far ciò si usano tecniche di checksum.

-Autenticazione: il mittente e il destinatario devono essere sicuri della loro identità.

-Sicurezza operativa: ci sono dei dispositivi come firewall e sistemi di rilevamenti delle intrusioni che permettono di mantenere le reti collegate a Internet sicure da attacchi esterni.

Le azioni che un intruso può intraprendere sono per esempio l’intercettazione del traffico, inserimento di messaggio nella comunicazione, personificazione di un’altra identità oppure causare l’inaccessibilità di un servizio.

**8.2 Riservatezza**

Suppongo che un mittente voglia inviare un messaggio ad un destinatario. Il messaggio originario è detto plaintext (testo in chiaro) e successivamente viene trasformato da un algoritmo di cifratura, tramite una precisa chiave del mittente, in un messaggio ciphertext (messaggio cifrato), il quale è incomprensibile ad altri utenti. Infine, il destinatario trasforma il ciphertext in plaintext grazie ad un algoritmo di decifratura, tramite una sua chiave precisa. Ci sono due meccanismi per realizzare la crittografia: Crittografia a chiave simmetrica e a chiave pubblica.

**Crittografia a chiave simmetrica**

Si basa sull’esistenza di una chiave, che cambia a seconda del meccanismo usato per la cifratura, comune tra A e B usata sia per cifrare che per decifrare il messaggio. Esistono più tipi di algoritmi:

-il cifrario di Cesare, è un algoritmo che si basa sulla sostituzione di ciascuna lettera del messaggio originare con un’altra lettera sfasata rispetto alla prima di un numero k di posizioni. La chiave corrisponde alla variabile k.

-il cifrario monoalfabetico, è un algoritmo che si basa sulla sostituzione di ciascuna lettera del messaggio originale con un’altra lettera senza uno schema regolare. La chiave corrisponde al cifrario utilizzato.

-metodo più associazioni: il problema di questi due tipi di algoritmi di crittografia è che dal testo io posso ricostruire lo schema con l’analisi statistica. Per rendere più complicata la ricostruzione del messaggio da parte di intrusi viene usato un metodo con più associazioni di lettere ossia con più cifrari invece di uno. In questo modo uso diversi cifrari seguendo un pattern preciso che mi indica l’ordine di utilizzo dei cifrari. La chiave corrisponde agli n cifrari e al pattern usato.

-cifrari a blocchi: si cerca di aumentare il tipo di relazione tra i bit di ingresso e i bit di uscita. Per far ciò si divide il messaggio da cifrare in blocchi di k bit e ogni blocco viene cifrato in modo indipendente, ottenendo blocchi di bit di testo cifrato. Infine, i diversi blocchi vengono riassemblati e mescolati, alterandone così l’ordine per poi ripetere la procedura più volte. Il problema di questo meccanismo è che, per quanto possa essere articolata la cifratura dei messaggi, se viene dato in ingresso lo stesso messaggio in input, verrà restituito sempre lo stesso messaggio cifrato in output.

-cifrari a blocchi concatenati: per risolvere il problema dello stesso input, viene considerato lo schema a blocchi concatenati. Ogni messaggio, prima di cifrarlo, lo combino con un’operazione opportuna (xor) tra la sequenza di bit in ingresso (m(i)) e il codice i-1 esimo che ho cifrato per il blocco precedente (c(i-1)). Ogni codice cifrato, quindi, dipende da tutta la sequenza in ingresso.

Un esempio è il DES che usava chiavi a 56 bit e 64 bit di input ed era molto sicuro, considerando che a volte si usava il 3-DES e la tecnica dei cifrari a blocchi concatenati. Oggi non si usa il 3-DES ma una tecnica più avanzata detta AES, che usa una tecnica simile al DES ma lavora con 128 bit e ciò permette una maggiore sicurezza.

**Crittografia a chiave pubblica**

Permette di usare chiavi diverse e quindi non devono essere scambiate, il cui processo rappresenta il principale rischio di intercettazione. L’idea è che non c’è una chiave comune segreta bensì due chiavi per ciascuna parte: una chiave pubblica (K+) e una privata (K-) che non va condivisa e conoscere la chiave pubblica non da informazione sulla chiave privata, nonostante siano correlate. Applicando prima la chiave pubblica e poi la privata si ottiene il messaggio (K-(K+(m))=m). L’algoritmo principale a chiave pubblica è **l’RSA**.

**RSA**

Ricordiamo alcune proprietà del modulo come x mod n è il resto della divisione tra x e n e (a mod n)dmod n = admod n e ricordiamo che un messaggio non è altro che una sequenza di bit che può essere rappresentata in modo univoco da un numero intero. Quindi cifrare un messaggio equivale a cifrare un numero. RSA si basa su due punti principali: la scelta delle chiavi (pubblica e privata) e gli algoritmi di cifratura e decifratura. Per generare le chiavi vengono eseguiti i seguenti passi:

1. Vengono scelti due numeri primi *p* e *q* dell’ordine di 1024 bit e maggiore è il loro valore più sarà sicura RSA ma sarà più complicata la cifratura e la decifratura.
2. Viene calcolato *n=p\*q* e *z=(p-1)(q-1)*
3. Viene scelto un numero *e* minore di *n*, diverso da 1 e che non abbia divisori comuni con *z*.
4. Viene scelto un numero d tale che *e\*d*-1 sia divisibile per *z*
5. La coppia (*n, e*) è la chiave pubblica e la coppia (*n, d*) è la chiave privata.

Ora per cifrare un messaggio m vuol dire:

1. Calcolare c = memod n ossia il messaggio cifrato.
2. Calcolare m = cd mod n ossia il messaggio decifrato, utilizzando la chiave segreta.

Esempio:

scelte iniziali:

scelgo p=5 e q=7.

calcolo n= p\*q=35 e z=(p-1)(q-1)=24

scelgo e=3 e d=29

ho la chiave pubblica (35,3) e la chiave privata (35,29)

cifratura:

parto dalla lettera “l” che è la dodicesima lettera dell’alfabeto e quindi m=12

calcolo me=1524832

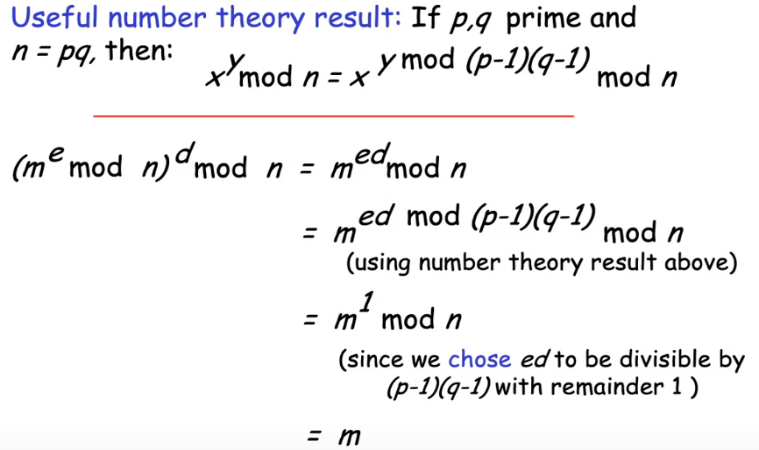
calcolo c = memod n = 17

decifratura:

calcolo cd=…

calcolo m = cdmod n=12

il messaggio è la n-esima lettera ossia “l”

Questo funziona perché

**Proprietà RSA**

1. Una proprietà importante del RSA è che se prendo un messaggio, lo cifro con la chiave pubblica e poi lo decifro con la chiave privata ottengo il messaggio ma vale anche l’opposto, ossia se prendo il messaggio, lo cifro con la chiave privata e poi lo decifro con la chiave pubblica ottengo il messaggio.

2. Inoltre RSA è sicuro perché nota la chiave pubblica è difficile calcolare *d* e quindi la chiave privata.

3. L’RSA è un processo molto più lungo rispetto al DES, a causa dell’elevamento a potenza ma allo stesso tempo il DES è meno sicuro del RSA. Perciò si può utilizzare una combinazione dei due: uso un meccanismo a chiave pubblica per comunicare la chiave simmetrica da usare e poi uso la cifratura a chiave privata per lo scambio di informazioni.

**8.3 Integrità del messaggio**

Quando un utente riceve un messaggio, vuole essere sicuro che è stato veramente mandato dal mittente che pensa lui e che il messaggio ricevuto sia identico da quello di partenza. Per far ciò si usano gli Hash crittografici:

si prende un ingresso m e si calcola una stringa H(m) di lunghezza fissata detta hash che viene usata per verificare che il messaggio non sia stato modificato. Inoltre, dati due messaggi diversi, è difficile che hanno la stessa hash. Ciò non avviene nel checksum, infatti, se due messaggi sono leggermente diversi, presentano comunque lo stesso checksum.

**Codice autenticazione dei messaggi**

Per garantire l’integrità dei messaggi, mittente e destinatario che vogliono scambiarsi un messaggio usano, oltre che alle hash, una stringa di bit chiamata chiave di autenticazione. Così facendo, il mittente crea un messaggio m, concatena la chiave s con m, calcola l’hash H(m+s) e trasmette il messaggio (m,H(m+s)). Il destinatario riceve il messaggio (m,h) e conoscendo s, si calcola l’hash e confronta che H(m+s) sia uguale a h. In caso affermativo, il messaggio ricevuto è uguale al messaggio inviato, altrimenti il messaggio è stato modificato.

Una variante di questo meccanismo è l’MD5 che funziona senza la chiave segreta e calcola un hash di 128 bit in 4 fasi.

Un altro algoritmo usato è lo SHA-1.

**Firma digitale**

È una tecnica crittografica simile alla firma a mano. Chi invia un documento può firmarlo in modo da certificare la sua provenienza ed è verificabile e non può essere alterato. L’idea è che quando un mittente scrive un documento, solo lui lo può firmare usando la sua chiave privata e cifrarlo. Quando un destinatario riceve un messaggio firmato, per verificare che effettivamente è stato inviato e firmato da un certo mittente e non è stato firmato da nessun altro, applica la chiave pubblica del mittente e verifica l’autenticità. L’uso della chiave pubblica per le firme digitali ha il problema per cui la cifratura e la decifratura risultano essere complicati e onerosi ed è per questo che viene usato un approccio più efficiente che consiste nell’introduzione delle funzioni hash nella firma. In pratica, il mittente non firma l’intero documento ma soltanto una sua “impronta”, ottenuta applicando al messaggio una funzione hash che genera una sequenza di lunghezza minore. Se si combinano le due cose, un mittente invia un messaggio, si calcola con la funzione hash il digest H(m) e lo firma con la chiave privata, in modo che il destinatario riceva il messaggio originario e il digest criptato. A questo punto, il destinatario prende il messaggio, ci applica la funzione hash, calcola H(m), usa la chiave pubblica del mittente per decifrare la firma e confronta se questi due digest sono uguali.

**8.4 Autenticazione del messaggio**

Consiste nel processo di autenticazione per il quale un’entità dimostra la propria identità e questa autenticazione avviene tramite dei protocolli detti ap, di cui esistono diverse versioni.

**Ap1.0**

Versione non usata perché un’entità afferma solamente di essere sé stessa ma l’altra entità non può verificarlo in alcun modo e un intruso potrebbe tranquillamente essersi inserito nella comunicazione.

**Ap2.0**

In questa versione un’entità afferma di essere sé stessa e fornisce inoltre il suo indirizzo IP, ma anch’essa è facilmente attaccabile.

**Ap3.0-Ap3.1**

In questa versione un’entità afferma di essere sé stessa, fornisce inoltre il suo indirizzo IP e invia una password segreta condivisa con l’altra entità. Purtroppo, però, la comunicazione non è sicura e un intruso potrebbe scoprire facilmente la password. Inoltre, può avvenire l’attacco di replica ossia un intruso registra i messaggi della prima entità e successivamente li utilizza per fingere di essere l’entità precedente. Un ulteriore versione ap3.1 può essere creata con la password cifrata ma ciò non cambierebbe la situazione.

**Ap4.0**

Consiste nell’utilizzo di numeri “nonce” usati una volta sola (once in a lifetime). In pratica due entità si scambiano un nonce R e il mittente invia il messaggio cifrato con la chiave nota ad entrambi e in questo modo si evita l’attacco di replica. Il problema è che richiede lo scambio della chiave simmetrica.

**Ap5.0**

È una versione leggermente modificata di quella precedente ma vi è l’utilizzo della crittografia a chiave pubblica, ovviando così al problema dello scambio della chiave simmetrica condivisa. Il problema di questa versione è il cosiddetto “man in the middle attack”, il quale consiste nell’inserimento di un intruso durante l’autenticazione che riesce a leggere tutto il traffico scambiato tra due entità. Questo avviene perché le chiavi pubbliche non sono certificate e per risolvere ciò si introduce la certificazione della chiave pubblica. Le “Certification Authority” sono delle entità che rilasciano certificati che associano chiavi pubbliche a identità e questi vengono firmati digitalmente dalle CA. Ora, se un’entità mittente vuole comunicare con un’altra entità destinatario, viene richiesta la certificazione del destinatario, si applica la chiave pubblica della CA al certificato e si ottiene la chiave pubblica del destinatario. A questo punto sono sicuro che quella ottenuta è la chiave pubblica del destinatario.

In particolare, un certificato contiene:

-un numero seriale

-le entità a cui fa da CA

-la firma del certificato

**8.5 E-mail sicura**

Applichiamo i principi di crittografia per migliorare la sicurezza della posta elettronica. Un’entità vuole inviare un’e-mail ad un destinatario. Si usa un meccanismo a chiave simmetrica dove la chiave viene scambiata usando la chiave pubblica del destinatario, la quale si ottiene dalla CA.

-Un esempio confidenziale, in cui il destinatario non sa nulla riguardo il mittente, è il seguente: il mittente scrive un messaggio, genera una chiave simmetrica e la usa per cifrare il messaggio (K­(m)). La chiave simmetrica deve essere comunicata al destinatario, viene cifrata usando la chiave pubblica del destinatario ed invio in rete il messaggio cifrato e la chiave simmetrica. Se qualcuno intercetta il messaggio non riesce a risalire alla chiave simmetrica. Il destinatario usa la sua chiave privata per estrarre la chiave simmetrica e la usa per decifrare il messaggio cifrato dal mittente.

-Un esempio che garantisce autenticazione e integrità è il seguente: il messaggio viene inviato in chiaro e viene calcolato la hash del messaggio e applica la firma digitale sul digest del messaggio. Entrambe le cose vengono inviate e il destinatario dal messaggio si ricalcola l’hash, usa la chiave pubblica del mittente (ottenuta dal certificato della CA) per estrarre il contenuto del messaggio cifrato (H(m)) e confronta se sono uguali per capire se il messaggio è quello originale e se gli è stato inviato proprio dal mittente che lui si aspetta.

-Un esempio che combina sia confidenzialità, autenticazione e integrità è il seguente: il mittente prende il messaggio, si calcola l’hash e applica la firma digitale sul digest del messaggio. Il mittente genera una chiave simmetrica, la cifra con la chiave pubblica del destinatario e invia un messaggio con la chiave simmetrica cifrata e un messaggio cifrato che contiene il messaggio originario con la firma digitale. Il destinatario usa la sua chiave privata per estrarre la chiave simmetrica, la usa per decifrare il messaggio, applica l’hash al messaggio, applica la chiave pubblica del mittente così autentica il messaggio per estrarre (H(m)) e fa il confronto. Così so che il messaggio è stato inviato proprio dal mittente che mi aspettavo, sono sicuro della sua identità e so che il messaggio non è stato modificato. Un protocollo che funziona come sopra riportato è il PGP, il quale permette la sicurezza della posta elettronica ed usa crittografia a chiave simmetrica, a chiave pubblica, funzione hash e firma digitale.

**8.6 SLL**

La crittografia può essere usata per arricchire TCP dando luogo ad un protocollo detto SSL, una versione arricchita di TCP con servizi di sicurezza, tra cui la riservatezza, l’integrità dei dati e l’autenticazione del client e del server. Rispetto agli altri protocolli questo permise il commercio sul web grazie alla sua sicurezza e cercò di rimediare alle mancanze degli altri protocolli come la cifratura e l’autenticazione di un web-server. Inoltre, era di facile applicazione ed era utilizzabile da qualsiasi applicazione che supportasse TCP. A differenza di un’applicazione normale, la quale interagisce tramite l’API con il protocollo TCP, nel caso di SSL, questo è un protocollo intermedio tra l’applicazione e TCP (protocollo intermedio di livello tra il 4 e il 5). SSL è implementato con delle librerie particolari ossia librerie software che permettono di invocare l’apertura e la chiusura di connessioni.

**Fasi di SSL**

È caratterizzato da diversi passi: l’handshake, in cui le due parti si scambiano il certificato e le chiavi per realizzare la comunicazione, la derivazione delle chiavi, in cui le parti usano una chiave comune per generare le altre chiavi, il trasferimento dati, in cui vengono trasferiti i dati, e la chiusura della connessione, in cui viene terminato lo scambio di dati.

Handshake: una volta inizializzata la connessione, nella fase di handshake il client invia un messaggio e il server risponde con il proprio certificato firmato con la chiave pubblica. Il client allora genera la chiave “master” e la invia, tramite la chiave pubblica, al server, il quale la decifra.

Derivazione delle chiavi: tramite la chiave master vengono generate 4 chiavi, due per la cifratura (una per i dati da client a server e un’altra per il viceversa) e due per la parte di integrità (MAC).

Trasferimento dati: il flusso di dati viene suddiviso in record, ne viene calcolato il MAC insieme alla chiave, ottenendo il messaggio e la firma. Tutto ciò, insieme ad un SLL number, viene cifrato con la chiave simmetrica, gli viene aggiunto il tipo di messaggio, la versione di SSL e la lunghezza e il tutto viene inviato.

Oltre a questi passi, c’è una fase di negoziazione, in cui viene deciso che tipo di algoritmo di cifratura usare.

**8.7 Sicurezza a livello di rete**

L’idea è che l’entità cifrano il contenuto del payload (segmenti TCP o UDP, messaggi OSPF, etc). Il traffico scambiato non è quindi visibile. Un modo per vedere la sicurezza al livello IP è l’uso, da parte di molti istituti, del protocollo di sicurezza IPsec per creare delle VPN, la cui idea è quella di far passare il traffico tra uffici al di sopra dell’internet pubblica, creando così una rete privata di tipo virtuale, in cui il traffico viene cifrato. Questo perché la creazione di una vera e propria rete privata ha un costo di acquisto, di manutenzione e di installazione molto elevati.

**Servizi IPsec**

I servizi offerti da IPsec sono la cifratura a livello di datagramma, l’integrità dei dati, l’autenticazione dell’origine, evitare attacchi di replay e confidenzialità dei dati. Esistono due principali protocolli: AH (authentication header) ed ESP (encapsulation security payload). Il primo fornisce solo l’autenticazione della sorgente e l’integrità dei dati mentre il secondo fornisce anche la riservatezza e per questo è più diffuso. Inoltre, IPsec ha due modalità di funzionamento: transport e tunnel. Nel primo solo il payload del datagramma è cifrato e autenticato, nel secondo l’intero datagramma viene cifrato e autenticato e questo viene incapsulato in un nuovo datagramma con un nuovo indirizzo IP, inviato successivamente verso la destinazione. Tra le 4 combinazioni di protocollo AH ed ESP con le due modalità di funzionamenti, quella con ESP in modalità tunnel è la più utilizzata e la più importante.

Prima di inviare i datagrammi, l’host sorgente e destinatario creano un canale logico a livello di rete detto associazione di sicurezza (SA). Essendo unidirezionali, se i due host vogliono scambiarsi datagrammi in modo sicuro devono esistere due associazioni di sicurezza. Le entità che si scambiano traffico devono mantenere le informazioni sulle SA e per garantire la comunicazione sicura ho bisogno di una coppia di SA per ogni coppia di entità che vogliono comunicare tra loro.

Tutte queste informazioni riguardo le SA vengono salvate nel loro database di associazione di sicurezza (SAD). Queste informazioni, infatti, servono ogni volta che devo scambiare traffico perché per ogni pacchetto da inviare da un router, devo sapere se è soggetto a politiche IPsec e vedere quali SA è associata a quel pacchetto. Se ci sono n venditori ogni router dovrà gestire 2+2n SA.

**Datagramma IPsec**

Un datagramma è formato dal pacchetto originale, un trailer ESP che aggiusta, tramite il padding, la lunghezza in modo che sia multipla di quanto previsto dal protocollo e queste due parti vengono cifrate. Inoltre, c’è l’intestazione ESP che include l’SPI, che indica all’entità ricevente a quale SA il datagramma appartiene, e il numero di sequenza e questa parte, insieme alla parte cifrata, viene autenticata. Tutto questo viene messo all’interno di un nuovo pacchetto IP. Il sequence number per una nuova SA parte da 0 e ogni volta che devo scambiare su una SA un datagramma lo incremento e ciò permette di prevenire un replay attack.

Tutte queste informazioni vengono memorizzate dall’entità IPsec e dal un SAD in una nuova struttura detta SPD (security policy database) e dunque per ogni datagramma le entità devono sapere come gestire quel pacchetto secondo IPsec. In particolare, l’SPD mantiene l’informazione su cosa fare quando arriva un pacchetto.

**Conclusioni IPsec**

Dunque, un intruso non riesce ad accedere al contenuto dei pacchetti perché sono cifrati, non riesce a vedere le sorgenti e le destinazioni perché queste informazioni fanno parte del payload che viene cifrato, non può usare l’indirizzo IP come indirizzo IP sorgente per sostituirsi ad esso perché la cifratura usa una chiave nota solo ai router e, infine, non può fare un replay attack perché i datagrammi vengono numerati e la numerazione è protetta con l’integrità. Quindi un intruso può vedere il traffico ma non può estrarne informazioni.

**IKE: Internet Key Exchange**

È un protocollo definito per scambiarsi tutte le informazioni che servono a stabilire SA tra nodi e router. Il compito di questo protocollo è quello di garantire l’autenticazione tra le parti, gestendo l’autenticazione e la generazione di chiavi trai nodi. L’autenticazione può avvenire in due modalità: con una chiave condivisa precedentemente (PSK) o con un meccanismo che usa lo scambio di certificati con chiave pubblica/privata. Nel primo caso i nodi hanno una password preimpostata e si autenticano tra loro generando SA in entrambi i versi, nel secondo caso utilizzando i certificati con chiave pubblica/privata si autenticano tra loro e stabiliscono le SA.

**Firewalls**

Il firewall è una combinazione di hardware e software che separa una rete privata da Internet e permette di controllare il traffico tra il mondo esterno e le risorse interne, scegliendo quali pacchetti far transitare e quali no. Gli obbiettivi di un firewall sono:

1. Impedire gli attacchi di SYN flooding, ossia un attacco di tipo denial of service nel quale un intruso invia una serie di richieste SYN-TCP verso il sistema oggetto dell’attacco, senza rispondere con un ACK e riempendo così la struttura dati del server in cui viene tenuta traccia delle connessioni instaurate.
2. Prevenire modifiche dei dati interni alla rete.
3. Impedire accessi dall’esterno della rete verso l’interno.

I firewall possono essere divisi in tre categorie:

1. Firewall stateless che fanno filtraggio di pacchetti, il cui compito è quello di analizzare il traffico in entrata ed in uscita e decidere se farlo passare o meno. Questa scelta viene effettuata in base alle informazioni di livello 3 o 4 come l’indirizzo IP sorgente e destinazione, i numeri di porta sorgente e destinazione, il tipo di messaggio ICMP e i flag TCP come i SYN e gli ACK. Per esempio, se non si vuole avere traffico Web uscente allora si scartano tutti i pacchetti con destinazione porta 80, oppure se non si vogliono avere connessioni TCP eccetto per quelle che riguardano un preciso Web server allora tutte le richieste SYN vengono scartate tranne quelle con un indirizzo stabilito e con porta destinazione 80.

Per effettuare questi controlli, ci sono delle regole che vengono implementate tramite delle liste di controllo di accesso (ACL): se un pacchetto non rispetta determinate regole allora non viene fatto passare.

1. Firewall con stato che fanno filtraggio di pacchetti: a differenza del precedente, in cui le decisioni del filtraggio venivano prese su ciascun pacchetto in modo indipendente senza guardare la storia precedente, in questi firewall viene tenuto traccia delle connessioni TCP in modo da vedere se determinati pacchetti hanno senso in quel momento. In questo caso, le ACL hanno un controllo sullo stato della connessione in quel momento.
2. Firewall di tipo application gateways: nel caso in cui si voglia fornire un servizio Telnet solo a determinati utenti, bisogna avere informazioni sull’identità degli utenti e queste informazioni non si trovano nelle intestazioni IP, TCP o UDP ma sono nel livello applicativo. Per far ciò, i firewall fanno il filtraggio dei pacchetti unito ad un gateway a livello applicativo, che prende decisioni in base ai dati applicativi. Ci sono, però, delle limitazioni perché i nodi possono cambiare il proprio indirizzo IP sorgente, ogni applicazione ha bisogno del proprio application gateway e ci sono dei costi in quanto i dati devono passare attraverso il gateway.

**Intrusion detection systems**

Sono degli apparati che analizzano l’intero contenuto di un pacchetto ossia vanno a guardare oltre i campi dell’intestazione, facendo un controllo più approfondito del pacchetto e controllando i dati applicativi che vengono trasportati. Questo viene fatto per evitare molti tipi di attacchi come per esempio virus, attacchi di mappatura della rete, scansione delle porte o sequenze riconducibili ad attacchi. Spesso vengono impiegati più IDS all’interno di una rete istituzionale, la quale può essere, per esempio, divisa in regioni in base al livello di sicurezza richiesto: zona interna ad alta sicurezza, che può comprendere i server interni, e zona demilitarizzata a bassa sicurezza, che può comprendere i server che comunicano con il mondo esterno.