
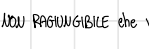


Firewall

Un firewall è un dispositivo (hardware) o un software che controlla il traffico di rete in entrata e in uscita verso e da una rete. Il suo obiettivo principale è quello di proteggere la rete da accessi non autorizzati e da attacchi informatici, impedendo l'accesso a risorse di rete non autorizzate.

- **Firewall software**  **OPERAZIONI FONDAMENTALI** → **BLOCCO DEL TRAFFICO** = Impedisce il passaggio dei pacchetti non autorizzati.
↳ **HA UN SICURO PERSPECTO AL FIREWALL HARDWARE**, poiché questo modello è implementato sulla stessa macchina che può essere compromessa.
↳ **PASSAGGIO DEL TRAFFICO** = fa passare i pacchetti autorizzati.
} *Scelta del pacchetto dipende da alcuni criteri di sicurezza specifici*

Il filtraggio in ingresso controlla il traffico di pacchetti che entra nella rete, mentre il filtraggio in uscita controlla il traffico di pacchetti che esce dalla rete. Entrambi i tipi di filtraggio possono essere configurati per bloccare o consentire il traffico in base a criteri di sicurezza specifici, come l'indirizzo IP di origine e destinazione, il numero di porta, il protocollo di rete, il tipo di dati contenuti nei pacchetti e così via. Il **blocco e il passaggio del traffico** sono le due operazioni principali eseguite dal firewall software. Il blocco del traffico impedisce il passaggio di pacchetti di rete non autorizzati o sospetti, mentre il passaggio del traffico consente il passaggio di pacchetti di rete autorizzati e legittimi. In questo modo, il firewall software può proteggere la rete da attacchi informatici, malware e altre minacce informatiche.

- **Firewall hardware**  **Application Gateway** è una macchina NON RAGGIUNGIBILE che viene interposta tra due LAN private per consentire il passaggio dei pacchetti da una rete privata all'altra.

Application Gateway è un Firewall HARDWARE che guarda il contenuto del pacchetto e stabilisce se scaricarlo o no, il suo costo è più elevato,
Un **application gateway** è una macchina **non raggiungibile** che viene interposta tra due LAN private per consentire il passaggio dei pacchetti da una rete privata all'altra. A differenza del gateway predefinito (che mi dice solo come trovare l'uscita da una rete) un **application gateway** è un **firewall hardware** che guarda il contenuto del pacchetto e prende decisioni sul proseguimento o scarto del pacchetto.

Il **costo** dei firewall hardware è molto più elevato dei firewall software, questo li rende molto meno utilizzati. Un firewall software è meno sicuro di uno hardware per il semplice motivo che si trova già sulla macchina, che può essere a sua volta compromessa.

Il concetto di porta aperta definisce una porta di ascolto dove un processo chiede di ricevere i pacchetti che il sistema operativo riceve su quella determinata porta, con porta chiusa si intende che il sistema operativo scarta a priori il pacchetto nonostante dei processi siano in ascolto su quella porta.

PORTA APERTA = Si definisce una porta di ASCOLTO dove un PROCESSO chiede di ricevere i pacchetti che il sistema operativo RICEVE SU QUELLA PORTA

PORTA CHIUSA = Il SO scarta a priori il pacchetto nonostante vi siano processi in ascolto.

DMZ (zona demilitarizzata)

La zona demilitarizzata (DMZ) è un'area di rete separata che viene creata all'interno di una rete aziendale o di un'altra rete protetta. La DMZ viene utilizzata per ospitare server e applicazioni che devono essere accessibili dall'esterno della rete, come server web, server di posta elettronica e server FTP. La DMZ è separata sia dalla rete interna che da quella esterna ed è protetta solo parzialmente dal router tramite uno o più firewall, che ne controllano l'accesso da parte degli utenti esterni e proteggono la rete interna da eventuali attacchi provenienti dalla DMZ. In genere, i firewall sono configurati per consentire solo il traffico di rete autorizzato tra la DMZ e la rete interna o esterna.

DMZ = è una zona NEUTRALE tra Rete interna (sicura) e una Rete esterna (NON SICURA). La sua principale funzione è quella di aumentare la SICUREZZA della RETE proteggendo i server dalla CONNESSIONE DIRETTA con INTERNET. (Di solito Ospita Server Web, Server Posta elettronica, SERVER FTP, che devono essere ACCESSIBILI dal ESTERNO).
L'è separata grazie ai Firewall limitando l'accesso ai server web.

Algoritmi di routing

ROUTING DISTRIBUITO = Si fa riferimento ad un approccio nel quale l'ALGORITMO DI ROUTING È IN ESECUZIONE SU OGNI NODO DELLA RETE

IN MODO DISTRIBUITO INDIPENDENTEMENTE dagli ALTRI NODI (in un approccio distribuito dobbiamo sapere quando l'algoritmo inizia e quando finisce).

Gli algoritmi di routing sono algoritmi utilizzati dai router per determinare il percorso ottimale per instradare i pacchetti attraverso una rete. Quando si parla di **routing distribuito** si fa riferimento ad un approccio nel quale l'algoritmo di routing è in esecuzione in ogni nodo della rete in modo distribuito ed indipendente dagli altri nodi. Ogni nodo utilizza le informazioni locali per determinare il percorso migliore per instradare i pacchetti. In un approccio distribuito, per garantire il corretto funzionamento della rete, bisogna che risulti chiaro quando l'algoritmo inizia la propria esecuzione e quando l'algoritmo si conclude.

Flooding

un algoritmo di routing che prevede l'invio del pacchetto a tutti i nodi; i nodi che lo ricevono lo inoltrano ai nodi adiacenti fino a che il pacchetto non raggiunge la destinazione.

Flooding è un algoritmo di routing che prevede l'invio di tutti i pacchetti di dati a tutti i nodi della rete.

Indipendentemente dalla destinazione finale del pacchetto. In altre parole, ogni nodo inoltra il pacchetto ricevuto a tutti i nodi ad esso adiacenti, che a loro volta lo inoltrano ai propri vicini, e così via, fino a quando il pacchetto raggiunge la destinazione finale. L'algoritmo di flooding è molto semplice da implementare e non richiede conoscenza della topologia della rete. Tuttavia, può causare un'elevata congestione della rete, poiché ogni pacchetto viene trasmesso a ogni nodo della rete. In tali reti, l'algoritmo di flooding può garantire una maggiore resilienza alla rete, poiché ogni nodo ha la possibilità di inoltrare il pacchetto a tutti i nodi vicini, indipendentemente dalla loro posizione o dal loro stato di connessione.

Si necessita di un modo per bloccare il flooding in quanto la ritrasmissione continua di pacchetti su tutti i link provoca una congestione dovuta al fatto che molti pacchetti vengono rispediti molte volte sugli stessi link. A tal proposito sono state proposte diverse soluzioni:

- **Usare un ID per i pacchetti** Si può pensare di dare un ID al pacchetto in modo da controllare la sua presenza nel determinato link. La dimensione della tabella diventa troppo grande.

Per bloccare il flooding si può pensare di aggiungere un identificativo ad ogni pacchetto così da poter tenere conto di tutti i pacchetti di cui si fa forwarding, in modo da non ritrasmettere copie dello stesso pacchetto se è già stato inoltrato. Questo approccio risolve la congestione ma sorge il problema della dimensione della tabella che tiene conto di tutti i pacchetti inoltrati e diventa difficoltoso memorizzarla a meno che il passaggio di un pacchetto per un link non sia sporadico.

- **TTL limite** Se si possono fare assunzioni sulla rete utilizziamo un TTL pari al DIAMETRO DEL GRAFO (ovvero distanza massima tra una coppia di nodi del grafo).

Se si possono fare assunzioni sulla rete si può invece pensare di stabilire un TTL limite pari al diametro del grafo, ovvero la distanza massima tra una coppia di nodi del grafo. Quindi si fissa il numero massimo di nodi che il pacchetto può attraversare prima di essere scartato pari al valore di diametro. Il valore limite di TTL viene utilizzato come meccanismo di sicurezza per prevenire il loop infinito dei pacchetti in una rete. Ogni volta che il pacchetto viene inoltrato a un nodo, il valore di TTL viene decrementato di uno. Se il valore di TTL raggiunge lo zero, il pacchetto viene scartato. In questo modo, il pacchetto viene inoltrato solo per un numero limitato di nodi, prevenendo il loop infinito. Il problema è che spesso non è possibile fare assunzioni sulla rete.

- **Spanning tree protocol** Probabilmente utilizzato per evitare la formazione di cicli indesiderati, costruisce un Albero eliminando archi che portano a nodi raggiungibili, come radice il dispositivo con MAC MINORE e si effettua una ^{ampiezza} (BFS), vi sono problemi di APPLICABILITÀ dove conoscere tutta la rete. (La rete può avere cambiamenti).

L'obiettivo dello STP è quello di creare un albero di copertura minimo, ovvero un sottografo dell'intera topologia di rete che contiene tutti i nodi e che non ha cicli ed elimina gli archi che portano a nodi già raggiungibili.

Si elegge un nodo radice sulla base dei confronti tra i vari MAC address dei nodi, in particolare la radice è costituita dal dispositivo che ha il MAC address minore e si effettua una visita in ampiezza (BFS) a partire da quel nodo.

Tuttavia, il protocollo STP riscontra problemi di applicabilità in quanto richiede la conoscenza completa della topologia di rete e può causare ritardi nella convergenza della rete in caso di cambiamenti nella topologia di rete.

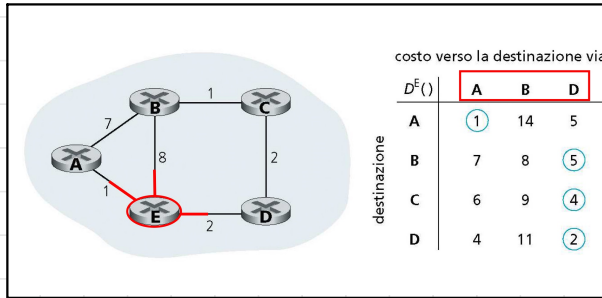
Il flooding, d'altra parte, è un algoritmo di routing distribuito che inoltra tutti i pacchetti di dati a tutti i nodi della rete, indipendentemente dalla destinazione finale del pacchetto. Questo algoritmo non viene utilizzato in Internet perché può causare una congestione di rete e perché non fornisce un percorso ottimale per il pacchetto.

Distance Vector (DV) : (Teorico)

Viene costruita una tabella dove con riferimento ad un certo nodo sorgente (nell'esempio E) si mettono in riga le destinazioni ed in colonna vengono inseriti i nodi verso i quali si obbliga il primo salto.

La tabella è rettangolare e non quadrata, in quanto si hanno nelle righe gli N nodi del grafo e nelle colonne solo gli M nodi verso i quali è possibile obbligare il primo salto, ovvero raggiungibili dalla sorgente con un solo arco.

Nella posizione (i, j) della tabella si trova il costo minimo per andare dalla sorgente alla destinazione 'i' obbligando il primo salto verso il nodo 'j'. Si nota che dal secondo in poi ogni salto può prendere qualunque direzione e si può tornare anche indietro. Si ricava il vettore delle distanze scegliendo i costi minimi in ogni riga, ovvero i percorsi di costo minore verso tutte le destinazioni.



Inizialmente non si possono fare assunzioni sulla tabella e vengono posti tutti i costi ad infinito. Ogni nodo costruisce il proprio vettore delle distanze e gli altri nodi nella rete cercano di migliorare i valori del proprio vettore utilizzando le proprie stime di cammino minimo, facendo variare la stima proposta dagli altri nodi.

Questo è un algoritmo distribuito basato su Bellman-Ford, e la convergenza viene raggiunta rapidamente in quanto ogni nodo lo esegue in modo distribuito ed indipendente dagli altri nodi.

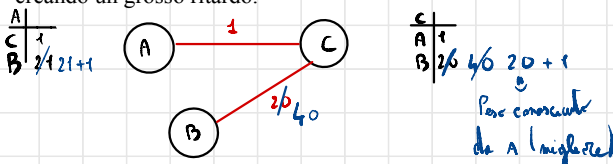
Sono noti i punti di inizio e fine del protocollo distribuito: Distance Vector inizia quando si ha una variazione nella stima dei cammini minimi e termina quando non si verifica nessun cambiamento durante l'iterazione precedente, ovvero quando si arriva a convergenza (e lo si fa rapidamente) e la situazione di rete diventa stabile quando non si hanno più variazioni nelle stime.

La tabella di inoltro dice quale è il prossimo passo da seguire nel cammino, quella di routing fornisce il quadro generale della rete. L'output dell'algoritmo distance vector è la tabella di inoltro. Si nota che questo algoritmo non implica la conoscenza a priori del grafo, quindi ogni nodo si affida alle informazioni ricevute dai nodi ad esso adiacenti.

Questo implica un forte problema di sicurezza dovuto al fatto che un nodo malevolo potrebbe dirottare il traffico destinato ad altri nodi. Non c'è soluzione a questo problema e si necessita di fidarsi. La terminazione dell'algoritmo avviene anch'essa in modo distribuito. Questo è stato il primo algoritmo ad essere utilizzato su internet seppur con delle precisazioni da stabilire, come il concetto di infinito uguale a 15, peso unitario dei link ecc...

Distance vector risponde male al peggioramento delle condizioni di rete

Un problema può sorgere quando il peso di un arco varia peggiorando le stime note ed i nodi che nella costruzione dei loro vettori tengono conto del peso di quell'arco avranno una stima fittizia, dato che tengono conto di un peso che è stato modificato. I nodi che avranno l'informazione sul peso effettivo dell'arco inviano messaggi di aggiornamento ai nodi adiacenti, i quali terranno conto di stime migliori (ma fittizie) sul peso di un arco. Quando vorranno attraversare quell'arco allora invieranno la propria tabella di inoltro ai router che migliorano le proprie stime (e aggiungono il solo peso dell'arco che li collega al dispositivo che ha inviato l'informazione). Il router non riesce a raggiungere la destinazione con quella stima e ritorna al mittente la propria tabella d'inoltro ed il pacchetto rimbalzerà continuamente creando un grosso ritardo.



Split horizon e Poisoned Reverse

Lo **split horizon** è una tecnica utilizzata in Distance Vector per prevenire i loop di rete dovuti al peggioramento di una stima. In pratica, lo split horizon prevede che un nodo non invii informazioni sul cammino minimo per raggiungere una destinazione a un altro nodo facente parte di tale cammino, al fine di evitare che i pacchetti vadano a ciclare indefinitamente nella rete.

Questo perché se il nodo ricevente utilizza il percorso ricevuto dal nodo mittente per migliorare la propria stima, essa terrà conto del peso fittizio dell'arco che ha peggiorato la propria stima, causando un loop di rete.

La tecnica del **poisoned reverse** viene utilizzata insieme allo split horizon per risolvere il problema delle stime fittizie causate dalla variazione del peso di un arco. **Quando c'è un peggioramento del peso di un arco, tutti i nodi che utilizzavano quell'arco per il calcolo del cammino minimo pongono la propria stima ad infinito, in modo da evitare che il pacchetto vada a ciclare nella rete.**

Considerazioni sull'approccio distribuito

Un sistema distribuito garantisce affidabilità ed alta efficienza e resistenza ai guasti. Si necessita di una sincronizzazione continua tra i dispositivi per mantenerli aggiornati e di uno **scambio limitato di informazioni** per garantire alti livelli di sicurezza, infatti se non si conosce l'informazione si rimanda a chi la conosce.

L'algoritmo in oltre deve arrivare velocemente a convergenza, per cui devono essere ben note le condizioni di inizio e fine di cui si tiene conto nell'esecuzione.