

FIREWALL

16/05/2023

É una componente alquanto necessaria poiché se ci sono vulnerabilità è possibile sfruttarle!
Se c'è anche solo un host viene infettato, tutta la mia LAN diventa vulnerabile.
Non basta quindi controllare gli endpoint (*endpoint security*)

Host poco controllati e mal configurati.

Questo è sempre più vero oggi giorno...
Si pensi ad esempio in ambito aziendale i cui dipendenti hanno i dispositivi aziendali (controllati e limitati) e i propri dispositivi personali (incontrollati)

É necessario controllare il traffico che entra nella mia rete locale (*sicurezza perimetrale*).
In tale ambito rientrano i Firewall

FIREWALL

IDEA: Tutto il traffico passa dal firewall che filtra e lo inoltra

Dall'inglese "porta antifuoco" poiché se scoppia un incendio fa sì che esso non si propaghi nelle altre stanze (LAN)

Affinché tutto il traffico passi per un unico punto (poiché non sono in grado di controllare tutti gli end-system) bisogna avere una topologia in cui il traffico proveniente dall'esterno (router) passi dal Firewall che poi provveda a filtrare e a inoltrare i pacchetti.

Quindi le funzioni di base del Firewall sono:

- Filtraggio
- Generazione Log (sul traffico ecc.)
- Generazione Allarmi

Oggi i Firewall sono concettualmente molto più complessi e pieni di numerose altre funzionalità (*NewGeneration-Firewall*)

Appliance = termine commerciale per indicare un firewall già programmato

TOPOLOGIE & TIPOLOGIE

Varie configurazioni:

• SCREENING ROUTER

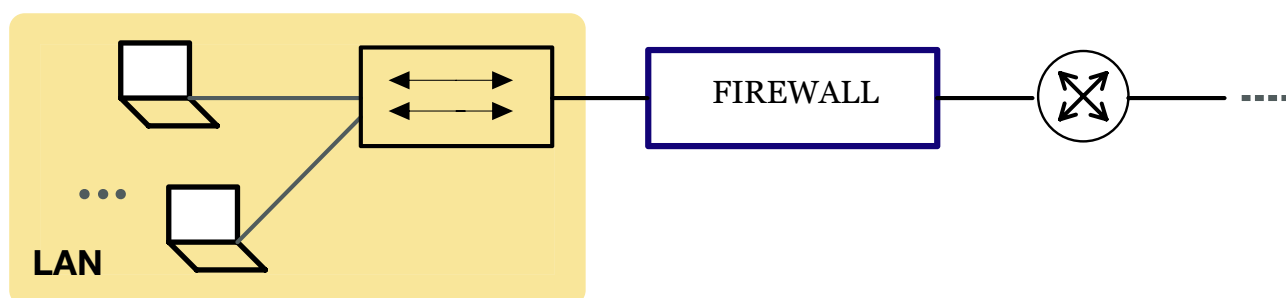
Router che filtra il traffico.

Può essere un'ottima soluzione in ambito casalingo in cui il router viene configurato (con delle semplici regole) per filtrare

• DUAL - HOMED GATEWAY

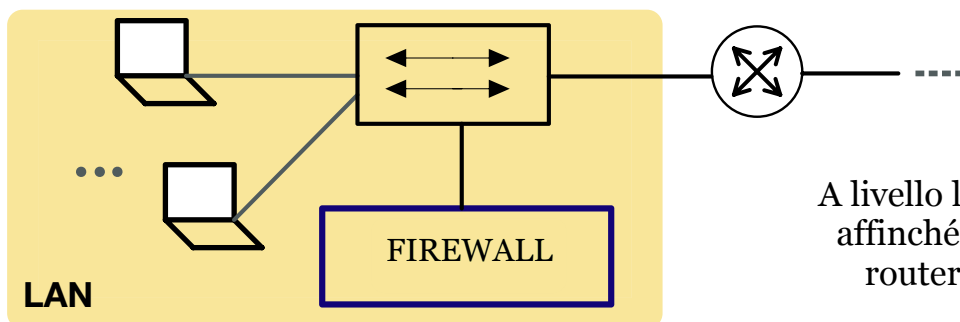
Firewall con 2 schede di rete

- una collegata alla LAN
- una collegata al router



• SCREENING HOST GATEWAY

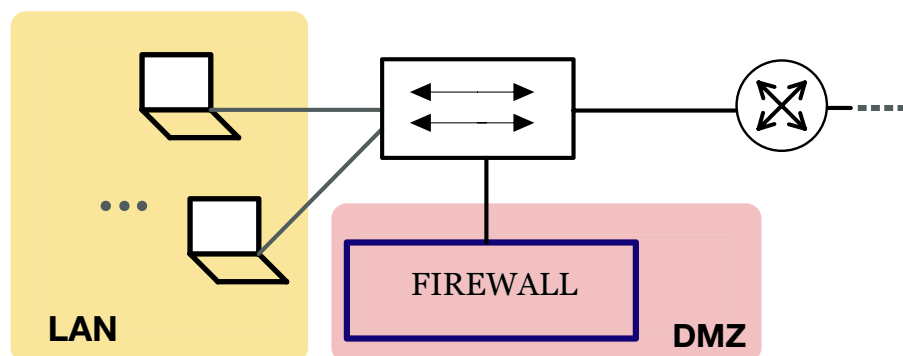
Firewall con 1 schede di rete



A livello logico poi programmo tutto affinché il traffico proveniente dal router passi prima dal firewall

• SCREENING SUBNET

Firewall con 1 schede di rete

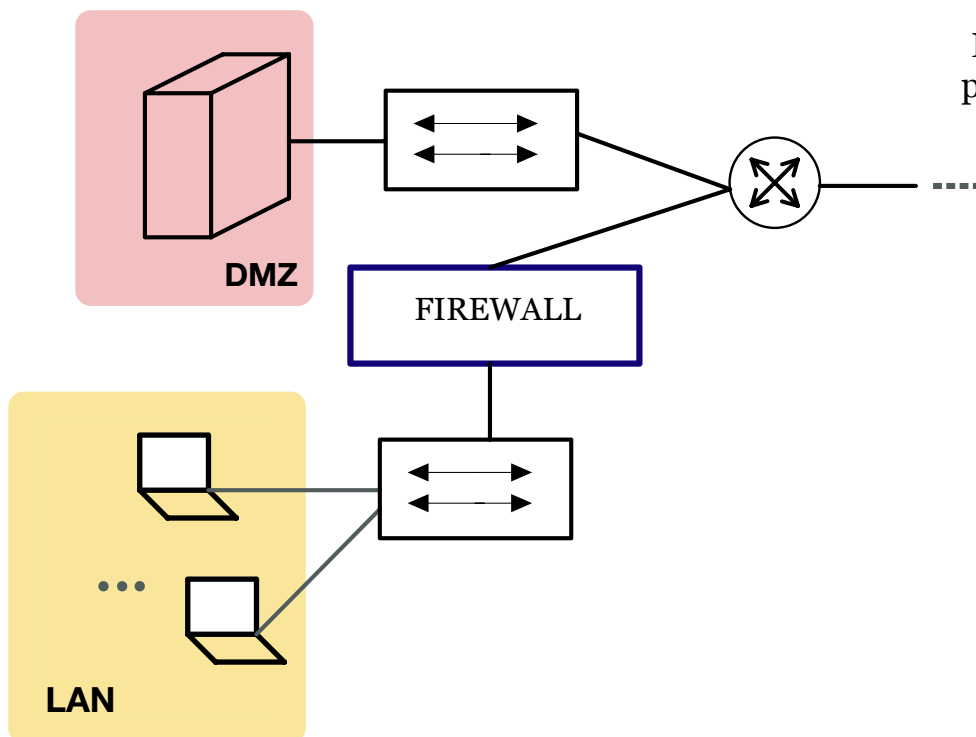


DMZ = De-Militarized Zone
(zona smilitarizzata)

↓
In tale zona è anche possibile ospitare dei servizi pubblici

• SCREENING SUBNET - continuazione

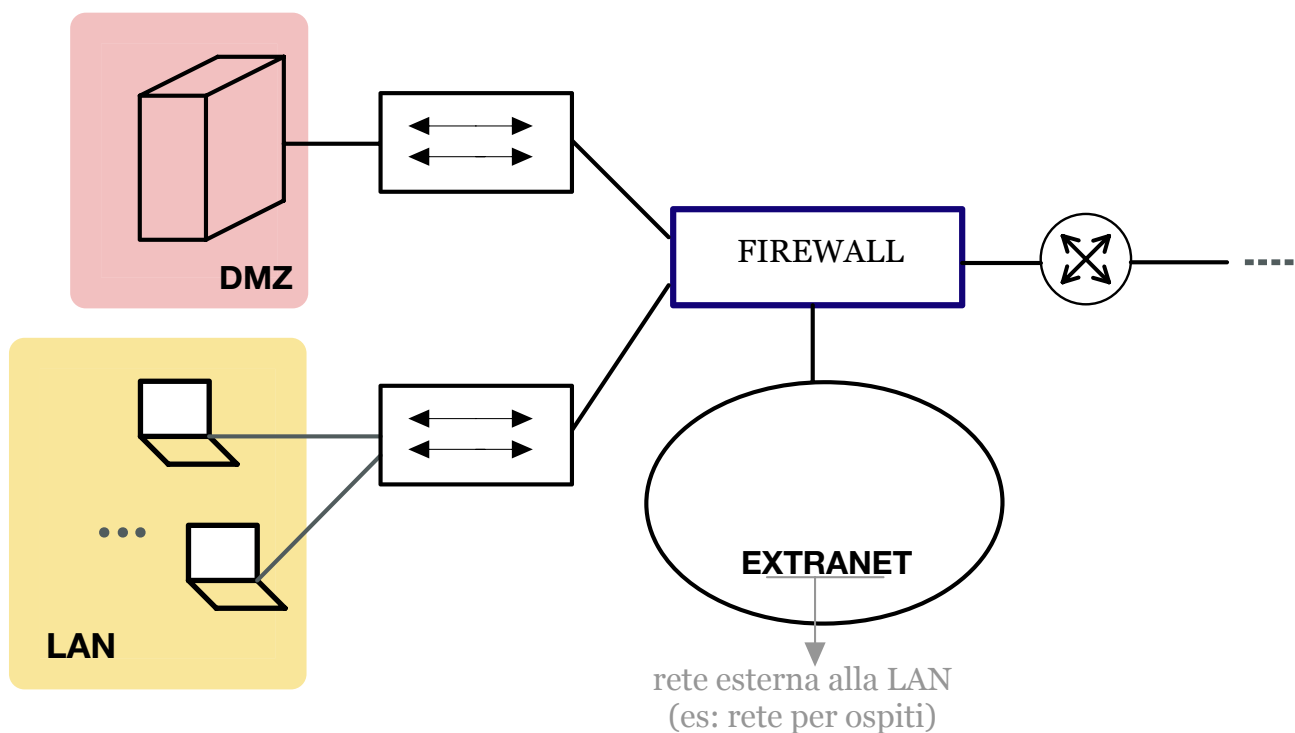
Firewall con 2 schede di rete



DMZ = De-Militarized Zone
(zona smilitarizzata)

↓
In tale zona é anche
possibile ospitare dei
servizi pubblici

Firewall con 4 schede di rete



Il concetto può essere generalizzando mettendo Firewall in cascata di tipo diverso e con regole più o meno restrittive.


DMZ = De-Militarized Zone (zona smilitarizzata)

Questa idea è nata nel momento in cui un'organizzazione ha dei servizi da offrire (es: HTTP/DNS...) verso l'esterno e allo stesso tempo ha una rete locale da proteggere dall'esterno.

Lo scopo della DMZ è quindi quello di aggiungere un ulteriore livello di sicurezza ad una organizzazione: un nodo appartenente ad una rete esterna può accedere soltanto ai servizi messi a disposizione, senza mettere a rischio e compromettere la sicurezza dell'intera rete.

HIGH AVAILABILITY

Bisogna garantire la *High Availability*: termine commerciale per indicare che se una macchina si rompe, il sistema nel complesso deve continuare a funzionare



Questo è necessario poiché nelle configurazioni viste prima il Firewall diventa un *single-point-of-fail*.
L'idea è collegare più firewall in cascata che comunicano attraverso un "ping applicativo" al fine di verificare il corretto funzionamento di tutte le macchine

TIPOLOGIE DI FIREWALL

• PACKET FILTER

Firewall che si limita a filtrare i pacchetti (a livello Rete)

• APPLICATION PROXY

Firewall che filtra e registra il traffico a livello Applicativo

PACKET FILTER

Firewall che filtra in base a:

- Direzione del pacchetto (da o verso l'esterno)
- Direzione della connessione TCP (da o verso l'esterno)
- Indirizzo IP sorgente e destinazione
- Servizio (porta sorgente/destinazione)

Alcune porte note:

Meglio Permettere		Meglio Bloccare	
20	dati FTP	43	whois
21	controllo FTP	67	bootp
23	Telnet	69	tftp
25	SMTP	79	finger
53	DNS	161	SNMP
80	HTTP	521	exec
110	POP3	517	talk

Problema: Frammentazione IP!

Soluzione: Scarto tali pacchetti!

Un problema importante nella configurazione di un firewall riguarda la frammentazione IP.

Infatti, se un pacchetto viene frammentato in pezzi molto piccoli, ogni parte può essere tanto ridotta da non includere neanche l'header TCP (che include la porta utilizzata dal firewall per filtrare).

Il Firewall si programma con le **ACL - Access Control List** che contiene delle regole

NB:

- Le regole vengono applicate nell'ordine.
 - Regole + stringenti: Sopra!
- Appena c'è match, applico tale regola

Permette HTTP dall'interno
(130.192.239.9 & Qualsiasi Porta)
verso l'esterno

Esempio:

Access Control List (ACL)

Action	Protocol	Src_Addr	Src_Port	Dst_Addr	Dst_Port	Flags
Allow	TCP	130.192.239.0	*	*	80	*
Allow	TCP	*	80	130.192.239.0	*	ACK
Deny	TCP	*	*	*	*	*

Permette il traffico di ritorno HTTP
dall'esterno (Qualsiasi Mittente &
Porta HTTP) verso l'interno

- **Action:** Azione (*allow/deny*)
- **Protocol:** Protocollo a livello superiore (TCP,UDP,...)

- **Src_Addr & Dst_Addr:** Indirizzo IP sorgente/destinazione
- **Src_Port & Dst_Port:** Porta sorgente/destinazione

In relazione ad essi viene aggiunta anche una **Src_SubNetMask & Dst_SubNetMask** (omesse nella tabella per manza di spazio)

Problema: Connessioni TCP!

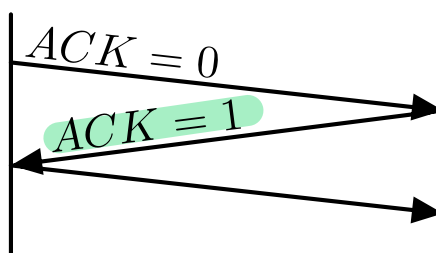
Devo impedire che un utente malevolo riesca ad aprire una connessione con un host della LAN

Soluzione: Flags!

Specifico il bit Flag dell'header da andare a controllare (SYN,ACK..)

Tramite al **Flags** é possibile evitare attacchi di questo, poiché verranno permessi (come traffico *out->in*) solo pacchetti ACK che contengono solo un riscontro (e non rappresenta una richiesta di connessione).
Per farlo nell'ACL devo scrivere ACK (che sarebbe equivalente a dire **ACK=1**)

questo semplice "trucco"
ci permette di non
utilizzare alcuna memoria
per capire se un pacchetto
è di connessione o meno



Per capire questo semplice concetto basta ricordarsi che una connessione TCP è composta da tre fasi:

Esempio:

Le seguenti regole permettono Telnet dall'interno verso l'esterno e ne impedisce connessioni dall'esterno verso l'interno.

Access Control List (ACL)

Action	Protocol	Src_Addr	Src_Port	Dst_Addr	Dst_Port	Flags
Allow	TCP	130.192.239.0	*	*	23	*
Allow	TCP	*	23	130.192.239.0	*	ACK

Supponiamo di voler impedire connessioni Telnet dall'Host interno 130.192.238.18. Per farlo utilizzo la seguente regola che va posta AL DI SOPRA delle precedenti regole; poiché regola più restrittiva.

SubNetMask = 255.255.255.255

Deny	*	130.192.239.18	*	*	23	*
------	---	----------------	---	---	----	---

É buona norma includere nelle ACL:

Deny	*	130.192.239.0	*	130.192.239.0	*	*
------	---	---------------	---	---------------	---	---

Bloccare il traffico proveniente dall'esterno verso l'interno, ma con un indirizzo di provenienza che risulta interno, cosa impossibile: indice di *Address Spoofing*

- Bloccare il traffico di tipo *Source Routing*

- Ideato per il monitoraggio della rete.
- Instradato alla sorgente (da qui il nome)
- Contiene tutti (o alcuni) gli hop dovrà seguire il pacchetto.
- Qualcuno potrebbe usarlo per finalità malevoli con *IP Spoofing*

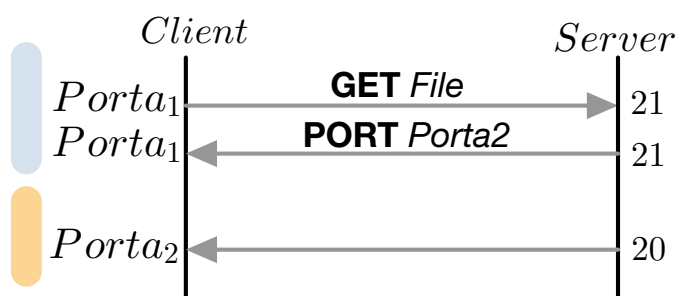
Problema: FTP! (O in generale protocolli *custom*)

Per capire questo concetto basta ricordarsi che FTP utilizza due porte e due connessioni:

- Una connessione *client->server* sulla porta 21
- Una connessione *server->client* sulla porta 20

Il problema risiede in quest'ultima connessione:

Action	Protocol	Src_Addr	Src_Port	Dst_Addr	Dst_Port	Flags
Allow	TCP	130.192.239.0	*	*	21	*
Allow	TCP	*	21	130.192.239.0	*	ACK
Allow	TCP	130.192.239.0	*	*	20	*
Allow	TCP	*	20	130.192.239.0	*	ACK



ACK=0 -> viene meno la sicurezza!
ACK=1 -> Non funziona FTP!

Soluzione 1: Vieto FTP!

Soluzione 2: Application Aware!

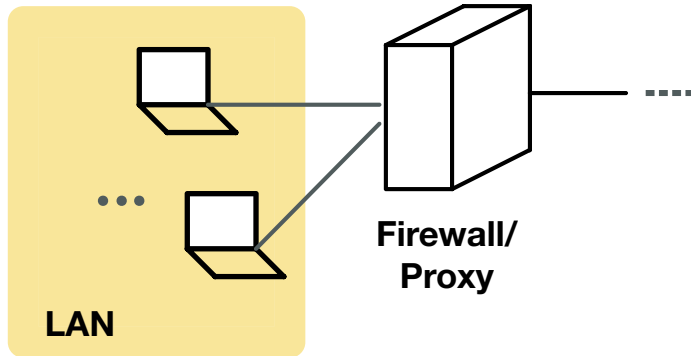
Che contiene una memoria in cui ricorda la porta lato client e permette una connessione solo su quella porta!

Firewall con stati (*stateful*) che riesce a fare una "*inspection*" del pacchetto.

Queste limitazioni del Packet Filter, lo rendono *semplice* e *veloce* poiché leggono il pacchetto "*on fly*" e senza conservare nulla in memoria (*stateless*) ma che è comunque *potente*...(seppur con i suoi limiti).

APPLICATION PROXY

È un Firewall che funge da Proxy: tutto il traffico passa da esso!



- Il Proxy fungerà da *server* per il Client della LAN
-
- Il Proxy fungerà da client per le richieste al Server effettivo (HTTP,...)

Complessità: Per ogni servizio (HTTP,FTP,...) deve esserci un Proxy specifico

Pro & Contro

Packet Filter:

- Nessuna modifica all'applicazione (*trasparente*)
- Economico (realizzato su router)
- Alte prestazioni
- Difficoltà con alcuni protocolli
- Non selettivo rispetto agli utenti
- Non mantiene log
- Difficile monitorare gli attacchi mentre avvengono

Application Firewall:

- Non trasparente
- Richiede un host dedicato
- Prestazioni medie
- Sicuro
- In grado di riferire il traffico dagli utenti
- Mantiene log sofisticati

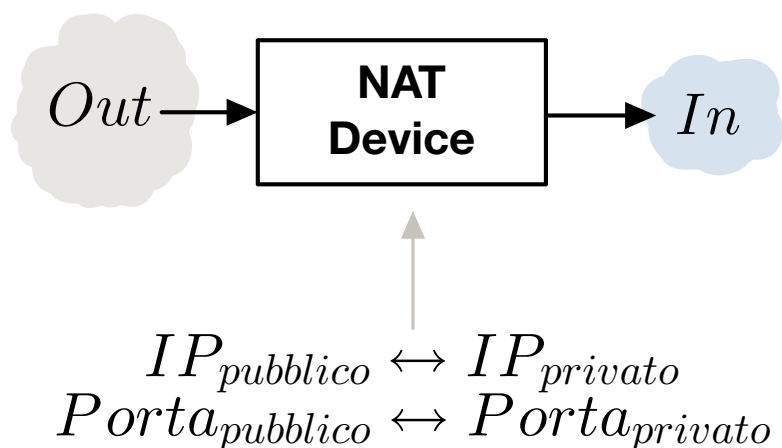
Concludiamo dicendo che l'ideale sarebbe combinare il concetto di "*trasparent*" (PacketFilter) con quello di "*stateful inspection*" (Application Aware)

Per ovviare ai problemi del Proxy, sono emerse soluzioni simili che prendono il nome di **Firewall Transparent FullInspection**, che consentono di fare l'ispezione anche sui livelli alti della pila (lv. Applicativo), pur rimanendo trasparenti e senza la necessità di installare il Proxy.

Concludiamo questa nostra trattazione parlando del **Mascheramento degli Indirizzi**.

Quello che si fa oggi, in mancanza di un Firewall, é *mascherare* gli indirizzi privati con degli indirizzi pubblici. Questo concetto prende il nome di **NAT - Network Address Translation** e per tale scopo si utilizza un **NAT Device**

Tale concetto é stato originariamente introdotto per mitigare il problema della scarsità di indirizzi IP pubblici disponibili.



In realtà oggi si parla di **NAPT - Network Address and Port Translation**, in cui si associa la coppia <IP,Porta>.

In questo modo, molti dispositivi possono condividere lo stesso indirizzo IP pubblico, differenziandosi per le porte utilizzate

È quindi una specie di Firewall, poiché si comporta come un Packet Filter nel caso in cui non esista alcuna associazione con la tabella.

WAP - Web Application Firewall

Firewall che protegge il WebServer (che generalmente é in DMZ e che quindi ha solo 1 livello di protezione). L'idea é indirizzare il traffico verso tale Proxy che filtra il traffico proveniente dall'esterno. A tal proposito si parla di *Reverse Proxy*

Personal Firewall

Firewall personale (su pc ecc) a livello software che svolge funzione di Firewall.

Rientra nei dispositivi di "end point security"

Attenzione: il Firewall non risolve tutti i problemi!