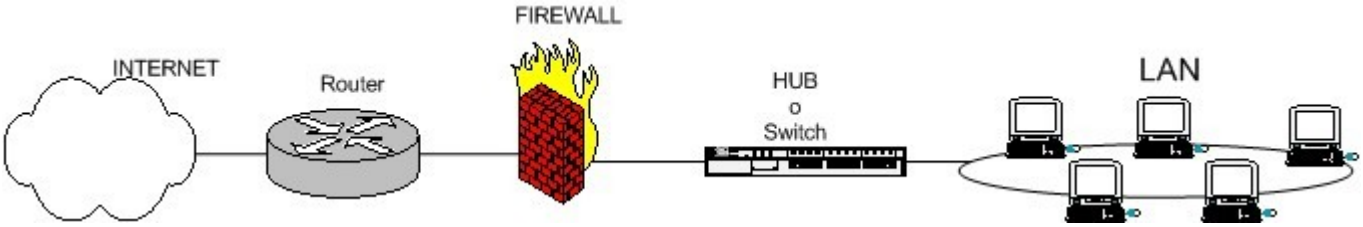
Unitat didàctica	5	Desenvolupament d'aplicacions: aplicacions client/servidor.
Nucli d'Activitat	6	Tallafocs IPTables

Firewall amb IPTables

Què és un firewall

Un firewall és un dispositiu que filtra el tràfic entre diferents (com a mínim dues) xarxes. El firewall pot ser un dispositiu físic o un software sobre un sistema operatiu. En general, el veurem com una caixa amb dues o més interfaces de xarxa a la que s'estableixen unes regles de filtrat amb les que es decideix si una connexió determinada pot establir-se o no. Inclús pot realitzar modificacions sobre les comunicacions (com ho fa la taula NAT d'IPtables).

L'anterior seria la definició genèrica, però avui en dia un firewall és un hardware específic que filtra el tràfic TCP, UDP o d'altres protocols (ICMP,IP,...) i decideix si un paquet passa, es modifica, es converteix o es descarta. Per que un firewall entre xarxes funcioni, haurà de tenir almenys dues targes de xarxa. Veiem la tipologia clàssica d'un firewall:

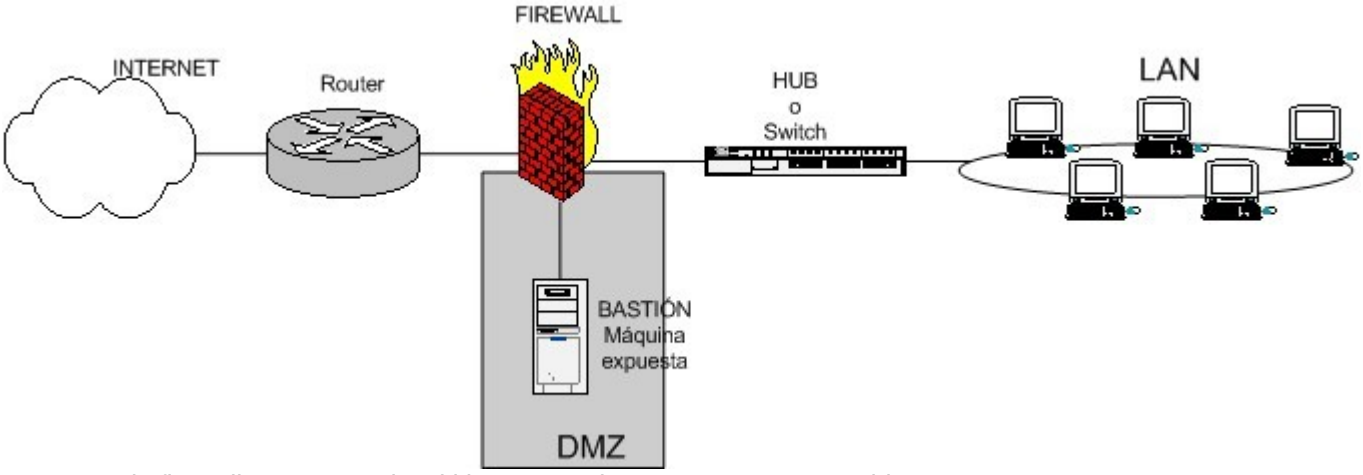


Esquema de firewall típic entre xarxa local i internet

col.loquem el firewall entre el router (amb un únic cable) i la xarxa local (connectat al switch o al hub de la LAN)

Segons les necessitats de cada xarxa, es pot ficar un o més firewalls per establir diferents perímetres de seguretat al voltant del sistema.

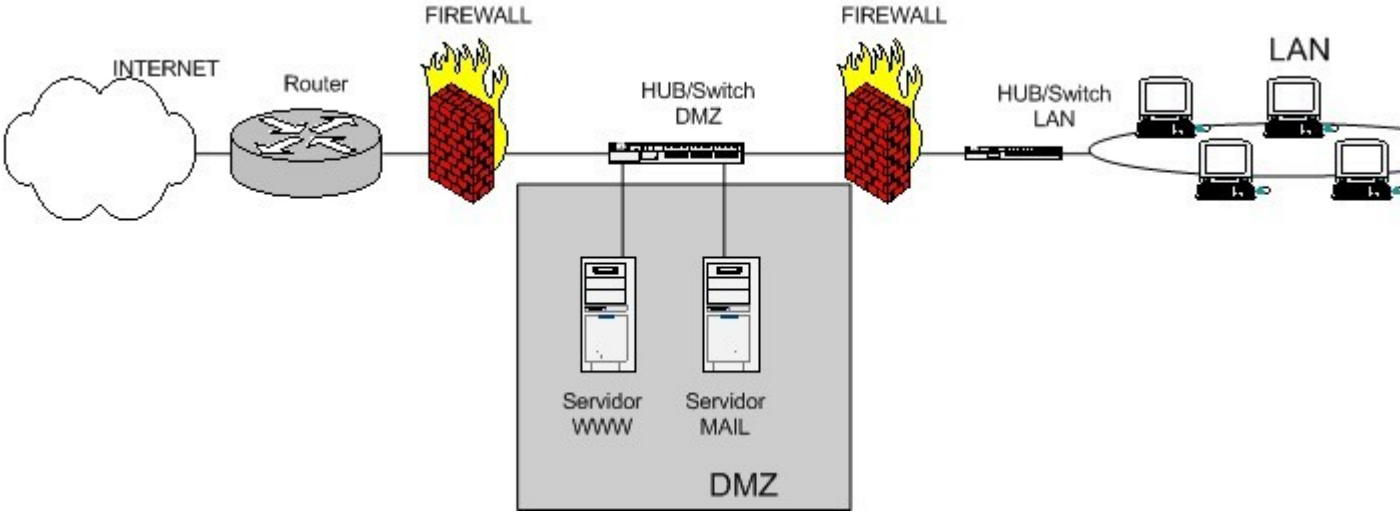
També és freqüent que necessitem exposar algun servidor a internet (com és el cas d'un servidor web, un servidor de correu, etc..), en aquests casos òbviament caldrà acceptar qualsevol connexión a ells. En aquesta situació es recomana situar dit servidor en un lloc apart de la xarxa, normalment anomedada DMZ o zona desmilitaritzada. El firewall tindrà llavors tres entrades:



Esquema de firewall entre xarxa local i internet amb zona DMZ per a servidors exposats

Amb aquest esquema permetrem que el servidor de la zona DMZ (podrien ficar-s'hi tants com fos necessari) sigui accessible per tothom, però mantindrem filtrat l'accés a la xarxa local.

Una altra possibilitat consistiria en ficar un firewall abans de la zona DMZ i un altre després per establir els filtres desitjats per a cada cas:



Esquema de firewall entre xarxa local i internet amb zona DMZ per a servidors exposats creat amb doble firewall.

Els firewalls es poden utilitzar a qualsevol xarxa. És habitual tenir-los com a protecció d'internet a les empreses, encara que és important tenir en compte que solen tenir una doble funció: controlar els accessos externs cap a dintre però també els interns cap a l'exterior; aquests darrers es realitzen amb el firewall o freqüentement amb un proxy (que també utilitza regles, encara que de més alt nivell).

A partir dels esquemes vistos, l'estructura es pot anar complicant, definint diferents proteccions per a diferents zones de la xarxa.

Com a característica comuna, veurem que tots els firewalls es poden reduir a un conjunt de regles en les que s'examina l'origen i destí dels paquets del protocol tcp/ip. També cal dir que els firewalls seran capaços de filtrar molts protocols diferents (tcp, udp, icmp, gre,...).

Com a primera aproximació, veurem en pseudo-llenguatge un exemple de regles típiques d'un firewall:

Política per defecte ACCEPTAR.
Tot el que vingui de la xarxa local al firewall ACCEPTAR
Tot el que vingui de la ip de casa meva al port tcp 22 ACCEPTAR
Tot el que vingui de la ip de casa del cap al port tcp 1723 ACCEPTAR
Tot el que vingui de hora.rediris.es al port udo 123 ACCEPTAR
Tot el que vingui de la xarxa local cap a l'exterior ENMASCARAR
Tot el que vingui de l'exterior al port tcp 1 al 1024 DENEGAR
Tot el que vingui de l'exterior al port tcp 3389 DENEGAR
Tot el que vingui de l'exterior al port udp 1 al 1024 DENEGAR

En definitiva, el que es fa és:
- Habilita l'accés a ports d'administración a determinades IPs privilegiades
- Emmascara el tràfic de la xarxa local cap a l'exterior (NAT, una petició d'un pc de la LAN surt a l'exterior amb la ip pública), per poder sortir a internet
- Denega l'accés des de l'exterior a ports d'administració i a tots els ports entre 1 i 1024.

Podem implementar un firewall de dues formes:

1) Política per defecte ACCEPTAR: en principi tot el que entra i surt pel firewall s'accepta i només es denegarà el que es digui explícitament.
2) Política per defecte DENEGAR: tot està denegat, i només es permetrà passar pel firewall allò que es permeti explícitament.

La primera política facilita molt la gestió del firewall, ja que només ens haurem de preocupar de protegir aquells ports que sabem que ens interessa. Per exemple, si volem protegir una màquina linux, podem fer un netstat -ln (o netstat -an, o netstat -puta | grep LISTEN), saber els ports oberts, posar regles per protegir aquests ports i no preocupar-nos dels altres que, realment mai s'obriran.

L'únic problema que podem tenir és que no controlem el que tenim obert, o que en un moment donat s'instal·li un software nou que obri un port determinat, o que no sabem que uns nous paquets siguin perillosos. Si la política per defecte és ACCEPTAR i no es protegeix explícitament, ens estem arriscant una mica.

Contràriament, si la política per defecte és DENEGAR, a no ser que ho permitim explícitament, el firewall es converteix en un autèntic MUR infranquejable. El problema és que és molt més difícil preparar un firewall d'aquest tipus i, a més, cal tenir molt clar com funciona el sistema, ja que haurem d'anar especificant el que volem obrir, però sense caure en la tentació de definir regles massa permisives.
Aquesta configuració de firewall és la recomanada, encara que no és aconsellable utilitzar-la si no es domina mínimament el sistema.

És molt important recalcar que l'ordre en el que posem les regles del firewall és determinant. Normalment, quan cal decidir què fer amb un paquet, es va comparant amb cadascuna de les regles del firewall fins a trobar una que li afecti (match), i llavors es fa lo indicat per aquesta regla (acceptar o denegar); després de tractar una regla, NO ES CONTINUARÀ MIRANT MÉS REGLES per a aquell paquet.

Hem de tenir en compte que si posem regles molt permisives entre las primeres del firewall, correm el perill que ho deixin colar tot i que les següents no s'apliquin mai i que, per tant, no serveixin per res.

IPTABLES

IPtables és un sistema de firewall vinculat al kernel de linux que s'ha extès enormement a partir de la versió 2.4 del kernel. IPtables és el successor d'un sistema de firewall anterior que es deia ipchains

IPtables no és com les eines vistes anteriorment, un servidor que iniciem i detenim, o que pugui caure per un error de programació(encara que sí ha tingut alguna vulnerabilitat, mai tindrà tant perill com les aplicacions que escolten en un determinat port TCP).

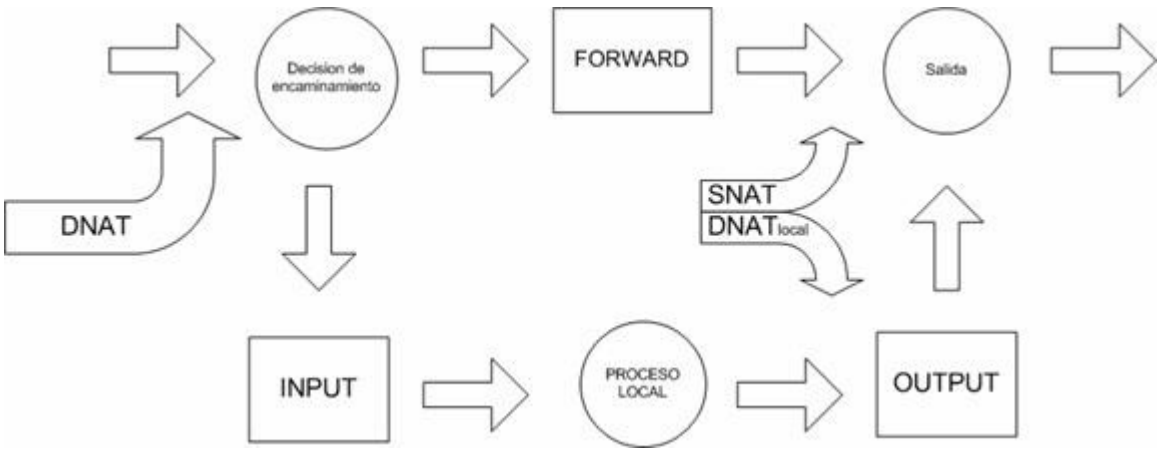
IPtables està integrat amb el kernel, és part del sistema operatiu. IPtables no s'engega i s'atura, sinó que sempre és present i aplica regles. Disposem de la comanda iptables, amb la que afegim, eliminem, o creem regles. Per això un firewall de iptables no és res més que un simple script de shell en el que es van executant les regles de firewall.

També cal dir que podem implementar un script d'inici a /etc/rc.d/INIT.d (o /etc/INIT.d) i fer que iptables s'inicii i s'aturi com un servidor més. A Red Hat ja trobarem aquest script inclòs a la distribució.

També disposem de la comanda iptables-save, que ens permetrà salvar les regles aplicades a un fitxer i gestionar-lo amb una aplicació front-end des de l'entorn gràfic o des de webmin.

El funcionament del kernel es podria resumir dient que rebrà paquets (definits en els formats dels protocols TCP/IP) i haurà de decidir què fer amb ells. El primer que caldrà fer és diferenciar entre els que són de la pròpia màquina (paquets que arriben o marxen de la màquina) i els que són per a una altra màquina (només passen per la màquina). Seguidament el kernel es basarà en el firewall per saber què fer amb ells. Avaluarà les regles definides al firewall, que seran les responsables d'indicar l'acció a realitzar en funció de les característiques del paquet.

Per tant, les taules estaran formades per les cadenes definides. Per a cada cadena definirem un conjunt de regles, que seran les que utilitzarem per indicar les accions a realitzar en cada moment.



Com es veu al gràfic, bàsicament es mira si el paquet està destinat a la pròpia màquina o si va a una altra. Pels paquets (o datagrames, si el protocol és UDP) que van a la pròpia màquina s'apliquen les cadenes INPUT i OUTPUT, i per filtrar paquets que van a altres xarxes o màquines s'apliquen cadenes FORWARD. Per a cadascuna d'aquestes cadenes podrem especificar regles, que indicaran les accions a prendre.

INPUT,OUTPUT i FORWARD són els tres tipus de cadenes de filtrat. Però abans d'aplicar aquestes cadenes és possible aplicar cadenes de NAT: aquestes s'utilitzen per fer redireccions de ports o canvis en les IPs d'origen i de destí.

Inclús abans de les cadenes de NAT, es podrà aplicar cadenes de tipus MANGLE, destinades a modificar els paquets; aquestes cadenes són poc conegudes i és probable que no s'utilitzin.

Resumint, podem dir que tenim tres tipus de cadenes, que s'emmagatzemaran a tres tables a iptables:

- MANGLE: permetran modificar paquets per a utilitats específiques
- NAT: cadenes PREROUTING, POSTROUTING . Aquesta tabla contindrà les cadenes que permetin alterar paquets d'inici de connexió segons el que especifiquem
- FILTER: cadenes INPUT, OUTPUT, FORWARD. Aquestes formaran la tabla per defecte de gestió de paquets

Per tant, la configuració de IPtables consistirà en mantenir (afegir, eliminar o modificar) el conjunt de regles per a cada cadena que conformen les tables definides a la màquina. Per exercitar el que comentem serà convenient anar provant aquestes configuracions a una màquina Linux.

Per comprovar l'efecte de les nostres configuracions, disposem de l'eina iptraf. Aquesta eina ens permetrà saber si les connexions TCP/IP s'arriben a establir o no. Una connexió tcp/ip comença amb el three-way-handshake:
- La màquina que desitja connectar-se a una altra envia un paquet amb flag SYN. Al protocol tcp/ip el paquet SYN no conté dades i és l'encarregat d'iniciar la comunicació
- Si l'altra màquina accepta, envia un SYN/ACK
- Llavors la màquina estableix la connexió

Si el firewall denega la connexió, amb iptraf veurem que la màquina origen només envia paquets amb el flag S (de SYN), i que de l'altra banda no surt res. Saber utilitzar iptraf ens ajudarà molt.

Anem a veure les diferents accions a indicar a cadascuna de les cadenes en funció de la taula en la que es fiqui aquesta:

Taula FILTER:

- INPUT: s'aplicarà als paquets rebuts de l'interface de xarxa
- OUTPUT: s'aplicarà a paquets enviats cap a l'interface de xarxa

- FORWARD: s'aplicarà als paquets que es rebuts per un interface de xarxa i enviats cap a un altre

Taula NAT:

- PREROUTING: alterarà els paquets rebuts de l'interface de xarxa quan arriben
- OUTPUT: permetrà alterar paquets generats localment abans d'enviar-los cap a l'interface de xarxa
- POSTROUTING: alterarà els paquets abans d'enviar-los cap a l'interface de xarxa

Taula MANGLE:

- PREROUTING: alterarà els paquets rebuts de l'interface de xarxa abans de ser dirigits
- OUTPUT: permetrà alterar paquets generats localment abans de dirigir-los cap a l'interface de xarxa

Tots els paquets que passin per la nostra màquina hauran d'estar subjectes almenys a una de les tres tables. Un paquet podrà complir diverses regles, però serà tractat per la primera regla que es satisfaci. Això implica que és molt important l'ordre en el que es defineixin les regles d'una cadena de IPtables.

A més de les tres tables indicades, podem afegir noves tables a IPtables, que seran guardades a:

/lib/modules/<versió del kernel>/kernel/net/ipv4/netfilter/

Quan un paquet compleix una regla, se li assigna un objectiu (target) fixat per la regla. Els objectius que es poden especificar seran:

- ACCEPT: el paquet es salta la resta de validacions i segueix cap al seu objectiu
- DROP: es nega l'accés del paquet al sistema. No s'envia cap notificació al remitent
- QUEUE: passem el paquet a l'espai d'usuari
- REJECT: s'entrega el paquet, però s'envia al remitent del paquet un missatge d'error

Si no hi ha cap regla que s'acompleixi pel paquet, llavors li aplicarem la política per defecte (ACCEPTAR o DENEGAR).

Un altre aspecte que serà molt important és l'ordre de les diferents opcions dins d'una regla, ja que la sintaxi de les regles és molt estricta. Per exemple, en les regles, el protocol (ICMP, TCP, UDP) s'ha d'especificar abans del port origen o destí.

També especificarem les interfaces d'entrada (opció -i) amb les accions que li corresponen: INPUT o FORWARD i les interfaces de sortida (opció –o) amb les accions que li corresponen OUTPUT o FORWARD. Les accions INPUT no són vistes pels paquets que es mouen cap a les interfícies de sortida, mentre que les accions OUTPUT no es podran utilitzar amb les interfícies d'entrada.

Les regles s'especificaran donant-les d'alta mitjançant la comanda iptables, en la que haurem d'especificar una de les següents opcions:

- Packet Type: indiquem el tipus de paquet que filtra la comanda
- Packet Source/Destination: indica els paquets a filtrar en funció del seu origen i/o destí
- Target: indiquem la acció a realitzar pels paquets que compleixin els criteris especificats anteriorment

Per tant l'estructura general serà:

$ iptables [-t <nom_tabla>] <comanda> <nom_acció> <paràmetre 1><opció 1>...<paràmetre n><opció n>

<nom_tabla> : indicarem la tabla sobre la que volem treballar. Si no s'especifica, per defecte és FILTER

<comanda>: indicarem la opció a realitzar amb la regla; afegir o eliminar la regla identificada per <nom_acció>

Finalment, els paràmetres i opcions seran els que indicaran el que cal fer quan un paquet coincideixi amb la regla.

La comanda iptables pot variar molt de tamany i ser molt curta o molt llarga en funció del que especifiqui. Teclejant:

$ iptables –h

podrem veure una llista detallada de l'estructura de les comandes iptables.

Comandes de la sentència iptables

Les comandes d'IPtables li indicaran les accions a realitzar. Totes les comandes s'escriuen amb majúscula (excepte la d'ajuda). Només podrem posar una comanda per cadena de comandes IPtables.

Veiem les diferents comandes:

-A: Afegir una regla al final de les regles d'una cadena. Utilitzarem aquesta comanda quan definim regles en les que l'ordre no és important

-C: Verificar una regla abans d'afegir-la a una cadena donada per l'usuari. Serà d'utilitat quan confeccionem regles complexes

-D: Eliminar una regla d'una cadena. La podrem indicar pel número de posició que ocupa o teclejant tota la regla

-E: Renombrar una cadena definida. No afecta l'estructura de la tabla (no provoca canvis d'ordre)

-F: Allibera una cadena de totes les regles que tenia definides

-h: Ajuda

-I: Afegeix una regla a una cadena. Podem especificar el número de posició en el que volem afegir la regla o, si no n'especifiquem cap, es col·locarà al tope(inici) de la cadena

-L: Llista totes les regles d'una cadena. Podem especificar que volem veure totes les cadenes d'una tabla:

iptables –L <nom-cadena> -t <nom-tabla>

-N: Permet crear una nova cadena amb el nom que li especifiquem

-P: Definim la política per defecte per a una cadena. És a dir, l'opció per defecte (ACCEPT o DROP) a aplicar quan un paquet corresponent a una cadena no acompleixi cap de les regles definides per a la seva cadena

-R: Substituir una regla definida a una cadena. Seleccionarem la regla per número (començant per 1)

-X: Elimina una cadena sencera. Només permet eliminar les cadenes definides per l'usuari. Les per defecte no poden ser eliminades

-Z: Posa a zeros els comptadors de bytes i de paquets de totes les cadenes definides a una tabla especificada

Paràmetres de la sentència iptables

Moltes de les comandes que actuen sobre regles que acabem de veure, necessitaran anar acompanyades d'una sèrie de paràmetres per construir les regles de filtrat dels paquets.

Veiem els possibles paràmetres que acompanyaran a les comandes:

-c: reseteja els comptadors d'una regla. Podrà anar acompanyat de les opcions PKTS i BYTES per especificar el comptador a resetejar

-d: configura el nom de la màquina destí, adreça IP o xarxa d'un paquet que coincideixi amb la regla. Per tant, ens permetrà definir regles que facin que tots els paquets que la acompleixin vagin a parar a on indiquem amb aquest paràmetre. Podem especificar:

N.N.N.N/M.M.M.M a on N.N.N.N és un rang d'adreces IP i M.M.M.M és la màscara de xarxa

N.N.N.N/M a on N.N.N.N és un rang d'adreces IP i M és la màscara de xarxa

-f: aplica la regla només a paquets fragmentats. També podem posar-li '!' al davant, negant el paràmetre i fent que només s'apliqui la regla als paquets no fragmentats

-i: configura l'interface de xarxa entrant (eth0, ppp0). Aquest paràmetre només tindrà sentit amb les cadenes relacionades amb entrada (INPUT i FORWARD a la tabla FILTER i PREROUTING a les taules NAT i MANGLE)

Per a aquest paràmetre podrem especificar '!', fent que excloem les interfícies especificades de la regla i també podem utilitzar un caràcter comodí '+', fent que sigui vàlid per a qualsevol valor. Per exemple, el paràmetre –i eth+ aplicarà la regla a totes les interfaces ethernet

Si definim –i sense especificar interface, la regla s'aplicarà a totes les interfícies

-j: indica a IPtables que salti a un determinat objectiu pels paquets que acompleixin la regla. Amb l'opció –j podrem especificar els objectius estàndard (ACCEPT, DROP, QUEUE, RETURN) i també altres objectius específics que es poden carregar de mòduls per defecte (LOG, MARK, REJECT,MASQUERADE).

Amb aquesta opció també podem saltar a una cadena definida per l'usuari per aplicar regles definides en aquella cadena.

Si s'especifica –j sense cap objectiu, es comptabilitza amb el comptador corresponent però no es realitza cap acció

-o: configura l'interface de xarxa de sortida per una regla. Per tant, de forma anàloga a l'opció –i, només es podrà utilitzar a les cadenes OUTPUT i FORWARD de la taula FILTER i a PREROUTING a les taules NAT i MANGLE

-p: Configura el protocol IP a aplicar per a la regla (tcp, udp, icmp i també accepta all). També podem aplicar qualsevol dels protocols que especifiquem a /etc/protocols. Si no s'especifica aquest paràmetre, s'aplica per defecte all

-s: configura el nom de la màquina font. Per tant, és l'opció anàloga a la –d

Opcions específiques del protocol TCP

Anem a veure una sèrie d'opcions que són específiques pel cas en que utilitzem el protocol TCP, per tant, quan haguem definit l'opció –p tcp.

--dport: Configura el port de destí per a un paquet. Podrem especificar un rang de ports o un nom de servei (www, smtp,...):

-p tcp –dport 3000:3200 El rang de ports és 0:65535

Amb aquesta opció també podem utilitzar la negació '!', indicant d'aquesta forma que volem fer que coincideixin tots els paquets que no utilitzin el rang de ports o el servei

--sport: Configura el port font per a un paquet. És l'opció anàloga a –dport i té els mateixos formats per especificar els ports o serveis

--syn: Opció per especificar els paquets inicials de comunicació. No la compliran els paquets amb dades, només els inicials. També podem utilitzar-la en sentit negat utilitzant '!' i així referenciar als paquets de dades

--tcp-flags: opció per fer que coincideixi amb els paquets que continguin flags. Portarà dos paràmetres, el primer és la màscara a cercar amb els flags del paquet. El segon paràmetre indicarà el flag a configurar per fer que coincideixin, els possibles valors són: ACK, FIN, PSH, RST, SYN, URG, ALL, NONE

Exemple:

-p tcp –tcp-flags ACK,FIN,SYN SYN

seleccionarà els paquets TCP que tinguin el flag SYN activat i ACK i FIN desactivats

Aquesta opció també ens permetrà utilitzar la negació '!' per invertir-la

--tcp-option: Intenta seleccionar els paquets que tinguin actives unes determinades opcions de TCP. També accepta utilitzar-se negada, mitjançant '!'

Opcions específiques del protocol UDP

--dport: Configura el port de destí per a un paquet UDP. Podrem especificar un rang de ports o un nom de servei (www, smtp,...):

--sport: Configura el port font per a un paquet UDP. És l'opció anàloga a –dport i té els mateixos formats per especificar els ports o serveis

Opcions específiques del protocol ICMP (Internet Control Message Protocol)

--icmp-type: permet seleccionar el nom o número de tipus ICMP que coincideixi amb el de la regla.

Fent iptables –p icmp –h obtindrem una llista de noms vàlids ICMP

Mòduls amb opcions de selecció adicionals

Tal com hem comentat en un punt anterior, disposarem de mòduls que es poden carregar per permetre utilitzar altres opcions.

Per carregar un mòdul d'opcions utilitzarem l'opció –m <nom_mòdul>

Existeix una gran quantitat de mòduls disponibles per a ser carregats. A més, també tindrem la possibilitat de definir els nostres propis mòduls.

Anem a veure els més populars:

Limit module: permet establir límits en funció dels comptadors de paquets que són tractats per a cada regla. Això serà útil per evitar que es sobrecarregui el sistema amb missatges repetitius. Aquest mòdul habilita les següents opcions:

--limit: Configura el nombre de coincidències en un intèrval de temps determinat:

--limit 5/hour

especificarem que no podem tractar més de 5 paquets amb aquesta regla cada hora.

Si especifiquem aquesta clàusula sense especificar res més, agafarà per defecte 3/hour

--limit-burst: opció complementària a la anterior que permet especificar el nombre de paquets que podran ser tractats per una regla en un temps determinat

State module: aques mòdul dóna la consciència d'estat, per tant permetrà especificar opcions d'estat:

--state: coincideix si el paquet té un determinat estat de connexió:

ESTABLISHED: el paquet pertany a una connexió ja establerta

INVALID: el paquet no pertany a cap connexió establerta

NEW: paquet que estableix una nova connexió

RELATED: paquet que estableix una nova connexió en un punt determinat d'una connexió ja existent

Exemple: -m state –state INVALID,NEW

Mac module: habilita la coincidència de direccions MAC de hardware

--mac-source: coincideix l'adreça MAC amb la de la tarja de xarxa que ha enviat el paquet. També podrem excloure-les negant la opció amb '!'

Opcions d'objectius

Una vegada hem especificat les opcions per indicar les condicions que faran que el paquet coincideixi amb la regla o no, ens resta comentar les opcions d'objectius, que seran les que indicaran les accions a realitzar amb aquest paquet.

Veiem els objectius estàndards:

ACCEPT: permet que el paquets vagi cap al seu destí.

DROP: el paquet cau, és a dir, no el reenviem ni donem cap avís al remitent

QUEUE: el paquet s'inserta a una cua que serà tractada per una aplicació d'usuari

RETURN: si el paquet amb destí RETURN acompleix una regla d'una cadena que ha sigut cridada des d'una altra cadena, retornem a la cadena original per seguir-lo verificant des de on ens havíem quedat al venir a la segona cadena. Si utilitzem RETURN en una cadena que no hagi sigut cridada des d'una altra, se li aplicarà l'objectiu per defecte de la cadena

CADENA DEFINIDA PER L'USUARI: Aquest objectiu passarà el paquet a la cadena que haguem especificat

Al igual que el cas anterior dels mòduls d'opcions de selecció, per les opcions d'objectius també podrem carregar altres opcions des de mòduls que poden ser predefinits o els podríem arribar a definir nosaltres.

Anem a veure els mòduls més utilitzats:

LOG: registra tots els paquets que coincideixen amb aquesta regla. Per defecte, aquests registres es guarden a l'arxiu (/var/log/messages), encara que això es pot configurar modificant l'arxiu /etc/syslog.conf

L'objectiu LOG permetrà indicar diverses opcions per indicar com registrar la informació del paquet:

--log-level: Configura els nivells de prioritat del registre d'events. Els nivells definits són (de major a menor): emerg, alert, crit, err, warning, notice, info i debug

--log-ip-options: qualsevol opció definida a la capçalera d'un paquet IP es guarda al registre

--log-tcp-sequence: escriu al log el número de seqüència del paquet TCP

--log-tcp-options: qualsevol opció definida a la capçalera d'un paquet TCP es guarda al registre

--log-prefix: permet insertar un prefixe a les línies de registre (fins a 29 caràcters). Això serà molt útil posteriorment per filtrar la informació continguda en els logs

REJECT: paquet que habilita la capacitat de deixar caure el paquet i enviar un error al sistema remot.

--reject-with <type>: condició que permet especificar l'error segons el tipus de rebutjament que enviarem al sistema remot. El valor per defecte és port-unreachable, encara que hi ha més valors possibles

Opcions de llistats

La comanda iptables –L proporciona una visió molt bàsica de les regles definides. Per aconseguir una informació més detallada, disposem d'una sèrie d'opcions més concretes:

-v: Treu per la pantalla informació referent al nombre de paquets i bytes que cada cadena ha vist, nombre de paquets i bytes que cada cadena ha trobat i els interfaces que s'apliquen a cada regla

-x: Força que al mostrar quantitats de paquets i bytes no s'utilitzin agrupacions com K per milers, M per millions o G per gillions. Amb aquesta opció sempre veurem les quantitats senceres

-n: mostra les adreces IP i els números de port en format numèric enlloc d'utilitzar el nom de servidor i de xarxa, que és l'opció per defecte

--line-numbers: llista totes les cadenes juntament amb el seu ordre numèric associat. Aquesta opció s'utilitzarà quan desitgem utilitzar opcions vistes anteriorment que poden utilitzar-se referint les cadenes per número. Amb aquesta opció, prèviament veurem les assignacions de números a les cadenes.

-t: especifica un nom de taula

Guardar el contingut de les taules de cadenes i regles a disc

El conjunt de cadenes i regles definit, si es reinicia la màquina es perdrà. Disposem d'una opció específica que ens permetrà guardar-ho a disc per poder restaurar-lo en rearrencar la màquina:

/sbin/service iptables save

Aquesta acció l'ha de realitzar l'usuari root. Això executa el script d'inici de iptables, que conté el programa /sbin/ iptables-save, que escriu la configuració actual de iptables a l'arxiu /etc/sysconfig/iptables.

Quan rearrenqui el sistema, el mateix script d'inici de iptables restaurarà les nostres configuracions mitjançant la comanda /sbin/iptables-restore.

Abans de gravar les regles a aquest arxiu serà recomanable assegurar-nos de que són correctes.

Copiant l'arxiu /etc/sysconfig/iptables d'una màquina a una altra i fent */sbin/service iptables restart* per rearrencar el servei, podrem fer que una altra màquina adopti les mateixes configuracions d'iptables que les nostres.

Exemples

Una vegada hem vist tota la sintaxi per a la definició de la configuració d'un firewall utilitzant iptables, anem a veure alguns exemples de configuracions per entendre l'ús de les comandes i opcions explicades

---

1 Protegir la pròpia màquina

Suposem que tenim una màquina connectada a internet i volem protegir-la amb el seu propi firewall. L'únic que caldrà que fem serà crear un script de shell al que definirem les regles.

Abans de passar al script, farem un pseudocodi d'aquest script:

Comentaris (echo)
Esborrat de les regles aplicades fins ara (flush)
Aplicació de polítiques per defecte per a INPUT, OUPUT, FORWARD
Llistat de regles iptables.

Recordem que l'ordre de les regles és sumament important

Veiem ja el script:

```
#!/bin/sh
## SCRIPT de IPTABLES - exemple del manual d'iptables
echo -n Aplicant Regles de Firewall...

## FLUSH de regles
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establim política per defecte
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## Comencem a filtrar

# El localhost es deixa (p.e. connexions locals a mysql)
/sbin/iptables -A INPUT -i lo -j ACCEPT

# A la nostra IP li deixem tot
iptables -A INPUT -s 195.65.34.234 -j ACCEPT

# A un amic (231.45.134.23) li deixem entrar a mysql (port 3306) per que mantingui la BBDD
iptables -A INPUT -s 231.45.134.23 -p tcp --dport 3306 -j ACCEPT

# A un altre li deixem utilitzar FTP
iptables -A INPUT -s 80.37.45.194 -p tcp -dport 20:21 -j ACCEPT

# El port 80 de www ha d'estar obert, ja que és un servidor web.
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# tanquem la resta
iptables -A INPUT -p tcp --dport 20:21 -j DROP
iptables -A INPUT -p tcp --dport 3306 -j DROP
iptables -A INPUT -p tcp --dport 22 -j DROP
iptables -A INPUT -p tcp --dport 10000 -j DROP

echo " OK . Verifiqui que s'aplica el que hem configurat amb iptables -L -n"

# Final del script
```

Recordem que a aquest script li haurem de posar els permisos adients: chmod +x firewall1.sh o chmod 750 firewall1.sh

Degut a que aquest script és molt senzill, una vegada executat serà convenient fer un netstat per veure quins ports estan oberts i en estat d'escolta, ja que aquest script no filtra paquets UDP ni ICMP. Podríem ampliar-lo amb unes noves clàusules de control:

```
# tanquem el rang dels ports privilegiats. Cal tenir en compte que abans de posar aquest tipus de barreres
#caldrà haver obert els que sí han de tenir accés

iptables -A INPUT -p tcp --dport 1:1024 -j DROP
iptables -A INPUT -p udp --dport 1:1024 -j DROP

# Tanquem altres ports oberts
iptables -A INPUT -p tcp --dport 3306 -j DROP
iptables -A INPUT -p tcp --dport 10000 -j DROP
iptables -A INPUT -p udp --dport 10000 -j DROP
```

- Versió amb DROP per defecte

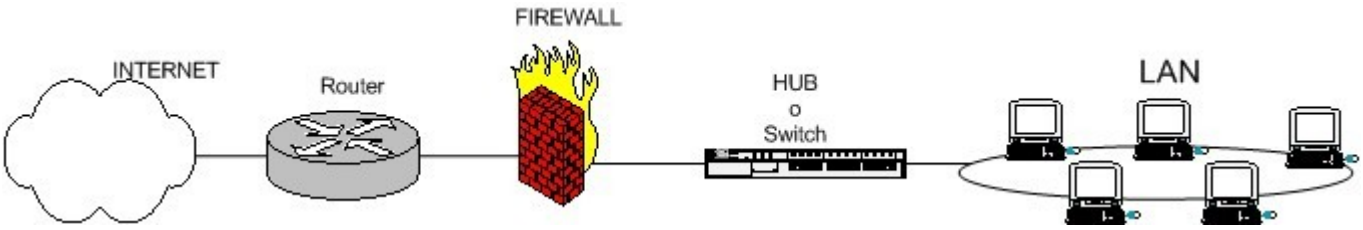
Si volem protegir la nostra màquina d'una forma més exigent, fixarem DROP com a opció per defecte. Llavors, només podrà entrar-se per on nosaltres especifiquem. Veiem com seria l'script en aquest cas:

```
#!/bin/sh
## Exemple de script per protegir la pròpia màquina amb DROP per defecte


echo -n Apliquem Regles de Firewall...

## FLUSH de regles
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establim política per defecte DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

## Ara haurem d'indicar explícitament tot allò que volguem obrir

# Operar en localhost sense limitacions
/sbin/iptables -A INPUT -i lo -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -j ACCEPT

# A la nostra IP li deixem tot
iptables -A INPUT -s 195.65.34.234 -j ACCEPT
iptables -A OUTPUT -d 195.65.34.234 -j ACCEPT

# Pel port pel que es dona servei a internet també ho acceptem tot

/sbin/iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m tcp --sport 80 -m state --state RELATED,ESTABLISHED -j ACCEPT

# Permitim que la màquina pugui sortir a la web
/sbin/iptables -A INPUT -p tcp -m tcp --sport 80 -m state --state RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT

# I també a webs segures (https)
/sbin/iptables -A INPUT -p tcp -m tcp --sport 443 -m state --state RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT

# Regles necessàries per a FTP passiu i actiu. Es permeten connexions entrants ja establertes
/sbin/iptables -A INPUT -p tcp -m tcp --sport 20:21 -m state --state RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m tcp --dport 20:21 -j ACCEPT
/sbin/iptables -A INPUT -p tcp -m tcp --sport 1024:65535 --dport 1024:65535 -m state --state ESTABLISHED -j
ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m tcp --dport 1024:65535 -m state --state NEW,RELATED,ESTABLISHED -j
ACCEPT

# Permitim la consulta a un primer DNS
/sbin/iptables -A INPUT -s 211.95.64.39 -p udp -m udp --sport 53 -j ACCEPT
/sbin/iptables -A OUTPUT -d 211.95.64.39 -p udp -m udp --dport 53 -j ACCEPT

# Permitim la consulta a un segon DNS
/sbin/iptables -A INPUT -s 211.95.79.109 -p udp -m udp --sport 53 -j ACCEPT
/sbin/iptables -A OUTPUT -d 211.95.79.109 -p udp -m udp --dport 53 -j ACCEPT

# Permitim consultar el rellotge de hora.rediris.es per sincronitzar-se
/sbin/iptables -A INPUT -s 130.206.3.166 -p udp -m udp --dport 123 -j ACCEPT
/sbin/iptables -A OUTPUT -d 130.206.3.166 -p udp -m udp --sport 123 -j ACCEPT
```

```
# Barrera de backup per si camviem a mode ACCEPT temporalment
# Amb això protegirem els ports reservats
/sbin/iptables -A INPUT -p tcp -m tcp --dport 1:1024 -j DROP
/sbin/iptables -A INPUT -p udp -m udp --dport 1:1024 -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --dport 1723 -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --dport 3306 -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --dport 5432 -j DROP

echo " OK . Verifiqui que s'aplica correctament amb: iptables -L -n"

# Final del script
```

2 Firewall d'una LAN amb sortida a internet

Veurem ara el típic cas d'una xarxa local que necessita sortida a internet, tal com es representa a l'esquema a continuació:



En aquest cas necessitem emmascarement de la LAN cap a fora. D'aquesta manera, totes les màquines de la LAN, enlloc de sortir amb la seva IP, sortiran totes amb la mateixa IP (la del firewall).

Normalment, hi haurà dos emmascaraments, al firewall i al router, i entre el router i el firewall hi haurà una xarxa privada (192.168.1.1 i 192.168.1.2 per exemple), encara que segons les necessitats totes dues podrien tenir IP pública.

En aquest tipus de firewalls, normalment es posarà una política per defecte de FORWARD denegant (DROP), però de moment, comencem per la més senzilla de ACCEPT:

```
#!/bin/sh
## Exemple de script per a firewall entre xarxa local i internet


echo -n Aplicant Regles de Firewall...

## FLUSH de regles
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establim política per defecte
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## Comencem a filtrar
## Nota: eth0 és l'interface connectat al router i eth1 a la LAN
# Deixem localhost (per exemple connexions locals a mysql)
/sbin/iptables -A INPUT -i lo -j ACCEPT

# Al firewall tenim accés des de la xarxa local
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT
```

```
# Ara fem enmascarament de la xarxa local
# l'emmascarament consistirà en substituir qualsevol IP de les diferents màquines per la de la de la màquina
# que té eth0
# i activem el BIT DE FORWARDING (imprescindible!!!!!)
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE

# Amb això permitim fer forward de paquets al firewall, o sigui
# que altres màquines puguin sortir pel firewall.
echo 1 > /proc/sys/net/ipv4/ip_forward

## Ara tanquem els accessos indesitjats de l'exterior:
# Nota: 0.0.0.0/0 significa: qualsevol xarxa

# Tanquem el rang de ports ben conegut
iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 1:1024 -j DROP
iptables -A INPUT -s 0.0.0.0/0 -p udp -dport 1:1024 -j DROP

# Tanquem el port de gestió de webmin
iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 10000 -j DROP

echo " OK . Verifiqui que s'aplica correctament amb: iptables -L -n"

# Final del script
```

Ara afegim que els usuaris de la xarxa local només puguin navegar per internet, denegant l'accés a Kazaa o edonkey. Per aconseguir-ho, a la configuració anterior afegiríem les següents clàusules, obrint l'accés de les màquines de la LAN només als ports que explícitament permitim i denegant els altres:

```
## filtrem amb la regla FORWARD l'accés de la xarxa local cap a l'exterior.  Els paquets que no van dirigits al
## propi firewall se'ls aplica la regla FORWARD


# Acceptem que vagin a ports 80
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -p tcp --dport 80 -j ACCEPT
# Acceptem que vagin a ports https
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -p tcp --dport 443 -j ACCEPT

# Acceptem que consultin els DNS
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -p tcp --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -p udp --dport 53 -j ACCEPT

# deneguem la resta. Si es necessita algun més s'habilitarà posteriorment
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -j DROP
```

Suposem que aquest firewall té alguna funció adicional: és un servidor proxy i, a més a més, és un servidor de correu. Donar-li funcionalitats d'aquest tipus a un firewall no és recomanable, ja que si no es protegeixen bé els ports d'aquests serveis o si no està actualitzat el software poden entrar al firewall fàcilment, comprometent tota la xarxa local. Moltes empreses no es poden permetre o no volen tenir una màquina per a cada cosa (moltes no volen ni posar un firewall). Per tant: si s'afegeixen serveis que han d'estar oberts al públic al propi firewall, ens l'estem jugant, i lo millor seria passar aquets serveis a una altra màquina i posar-la a una DMZ.
Suponsem també que el cap de l'empresa vol accedir a la xarxa local des de casa amb una connexió ADSL. Al firewall haurem de tenir instal·lats servidors SMTP, pop3 i PPTPD.

Veiem, respecte al script anterior, les clàusules que caldria afegir:

```
## Obrim l'accés a ports de correu

# Obrim el port 25 pel servidor SMTP
iptables -A INPUT -s 0.0.0.0/0 -p tcp --dport 25 -j ACCEPT
# Obrim el pop3
iptables -A INPUT -s 0.0.0.0/0 -p tcp --dport 110 -j ACCEPT
```

```
# Obrim el port pptpd per a la ip del adsl de casa del cap
iptables -A INPUT -s 211.45.176.24 -p tcp --dport 1723 -j ACCEPT

# Tenquem el port del servei PPTPD, només obert pel cap.
iptables -A INPUT -s 0.0.0.0/0 -i eth0 -p tcp --dport 1723 -j DROP
```

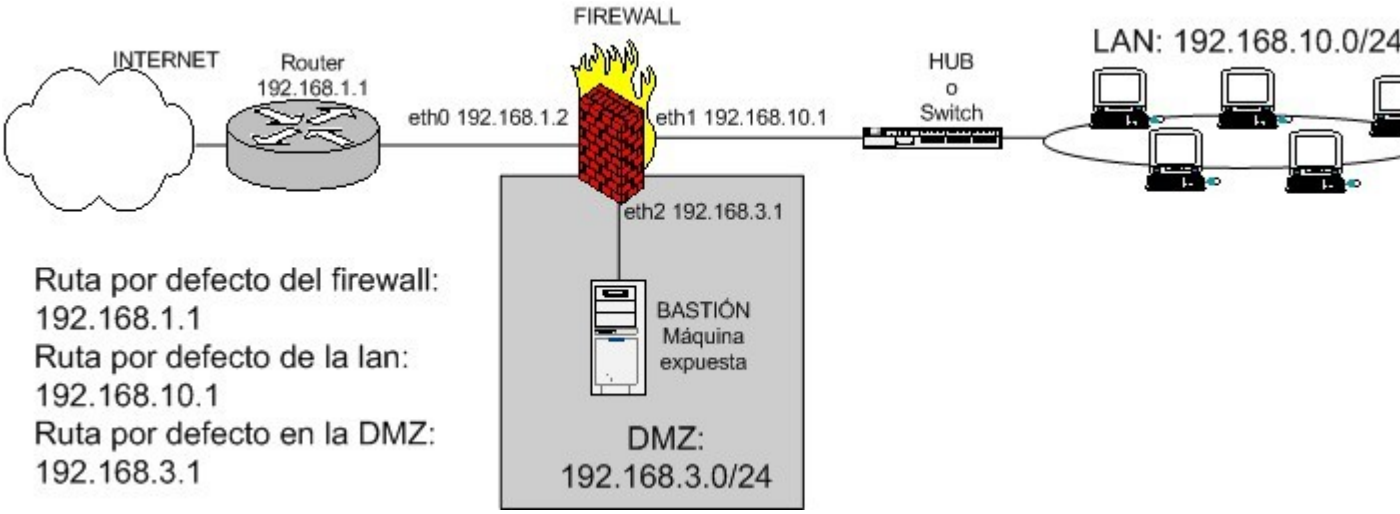
Ara volem compartir un servei però d'un servidor que tenim dins de la xarxa local, per exemple el IIS d'un servidor windows2000, i, a mé a més, permetre la gestió remota per terminal server per a aquesta màquina per a una empresa externa. Caldrà fer una redirecció de port. Afegirem noves regles DNAT per efectuar les redireccions:

```
## REDIRECCIONS

# Tot el que vingui de l'exterior cap al port 80 ho redirigim a una màquina interna
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.10.12:80

# Els accessos d'una ip determinada a Terminal server es redirigeixen e aquesta màquina
iptables -t nat -A PREROUTING -s 221.23.124.181 -i eth0 -p tcp --dport 3389 -j DNAT --to 192.168.10.12:3389
```

Cal tenir en compte que aquesta darrera configuració amb les redireccions i, si ja teníem els serveis de correu funcionant al firewall és bastant insegura. Si algú aconsegueix entrar al servidor IIS de la xarxa local, el firewall no servirà de gran cosa. Si realment necessitem aquest servidor IIS, serà millor comprar una tarja de xarxa i crear una DMZ.

3 Firewall d'una LAN amb sortida a internet amb DMZ

El que comentàvem al final del punt anterior es podria representar de la següent forma:



Així tenim la màquina de la DMZ que està exposada, però la resta de la xarxa local queda protegida pel firewall, minimitzant els perills

A aquest tipus de firewall caldrà permetre:
- Accés de la xarxa local a internet.
- Accés públic al port tcp/80 y tcp/443 del servidor de la DMZ
- Accés del servidor de la DMZ a una BBDD de la LAN
- Bloquejar la resta d'accessos de la DMZ cap a la LAN.

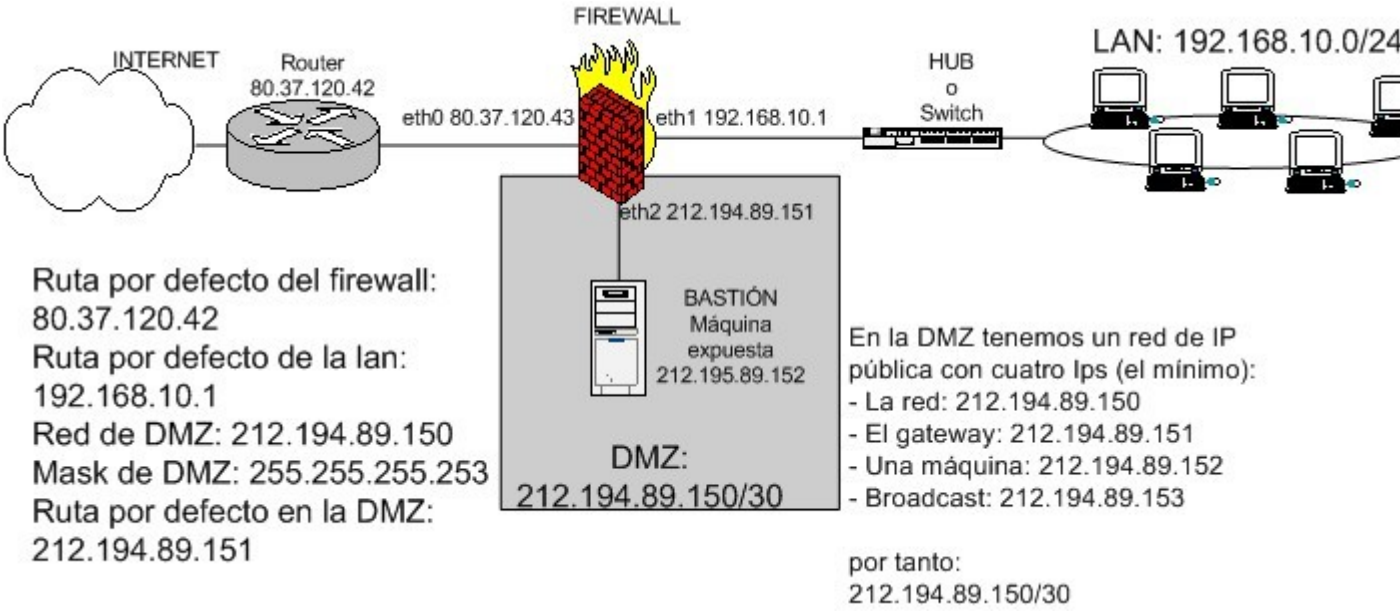
Veiem com ens quedaria l'script de configuració:

```
#!/bin/sh
## Exemple de script per a firewall entre xarxa local i internet amb DMZ


echo -n Aplicant Regles de Firewall...

## FLUSH de regles
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establim política per defecte
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## Comencem a filtrar
## Nota: eth0 és l'interface connectat al router y eth1 a la LAN
# Tot el que vingui de l'exterior cap al ports 80 i 443 (https) es redirigeixen a una màquina interna
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.3.2:80

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to 192.168.3.2:443

# El localhost es deixa (per exemple connexions locals a mysql)
/sbin/iptables -A INPUT -i lo -j ACCEPT

# Al firewall tenim accés des de la xarxa local
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT

# Ara fem l'emmascarament de la xarxa local i de la DMZ per que puguin sortir fora i activem el BIT
#FORWARDING (imprescindible!!!!!)
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.3.0/24 -o eth0 -j MASQUERADE

# Permitim fer forward de paquets al firewall, o sigui, que altres màquinas puguin sortir pel firewall.
echo 1 > /proc/sys/net/ipv4/ip_forward

## Permitim el pas de la DMZ a una BBDD de la LAN en els dos sentits:
iptables -A FORWARD -s 192.168.3.2 -d 192.168.10.5 -p tcp --dport 5432 -j ACCEPT

iptables -A FORWARD -s 192.168.10.5 -d 192.168.3.2 -p tcp --sport 5432 -j ACCEPT

## permitim obrir el Terminal server de la DMZ des de la LAN també en els dos sentits:
iptables -A FORWARD -s 192.168.10.0/24 -d 192.168.3.2 -p tcp --sport 1024:65535 --dport 3389 -j ACCEPT

iptables -A FORWARD -s 192.168.3.2 -d 192.168.10.0/24 -p tcp --sport 3389 --dport 1024:65535 -j ACCEPT

# Tanquem l'accés de la DMZ a la LAN
iptables -A FORWARD -s 192.168.3.0/24 -d 192.168.10.0/24 -j DROP

## Tanquem l'accés de la DMZ al propi firewall
iptables -A INPUT -s 192.168.3.0/24 -i eth2 -j DROP

## ara tanquem els accessos indesitjats de l'exterior:
# Nota: 0.0.0.0/0 significa: qualsevol xarxa

# Tanquem el rang de ports ben conegut
iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 1:1024 -j DROP
iptables -A INPUT -s 0.0.0.0/0 -p udp -dport 1:1024 -j DROP

# Tanquem un port de gestió: webmin
iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 10000 -j DROP
```

echo " OK . Verifiqui que s'aplica correctament amb: iptables -L -n"

Final del script

Si les màquines de la DMZ tenen una ip pública haurem d'anar amb molt de compte de no permetre el FORWARD per defecte. Si a la DMZ tenim ip pública NO CALDRÀ FER REDIRECCIONS de port, en tindrem prou amb enrutar els paquets per fer que arribin a la DMZ. Aquest tipus de necessitats surgeixen quan tenim, per exemple, dues màquines amb servidor web (un apache i un IIS); ¿A quina de les dues li redirigim el port 80? No ho podem saber (amb servidors virtuals tindríem el mateix problema), per això caldrà assignar IPs públiques o ports diferents.
Caldrà protegir tota la DMZ. Tampoc caldria emmascarar la sortida cap a l'exterior de la DMZ, si té una ip pública ja té un peu a internet; Caldrà dir-li al router com arribar fins a aquesta ip pública.



Al script anterior caldrà afegir-li les següents regles:

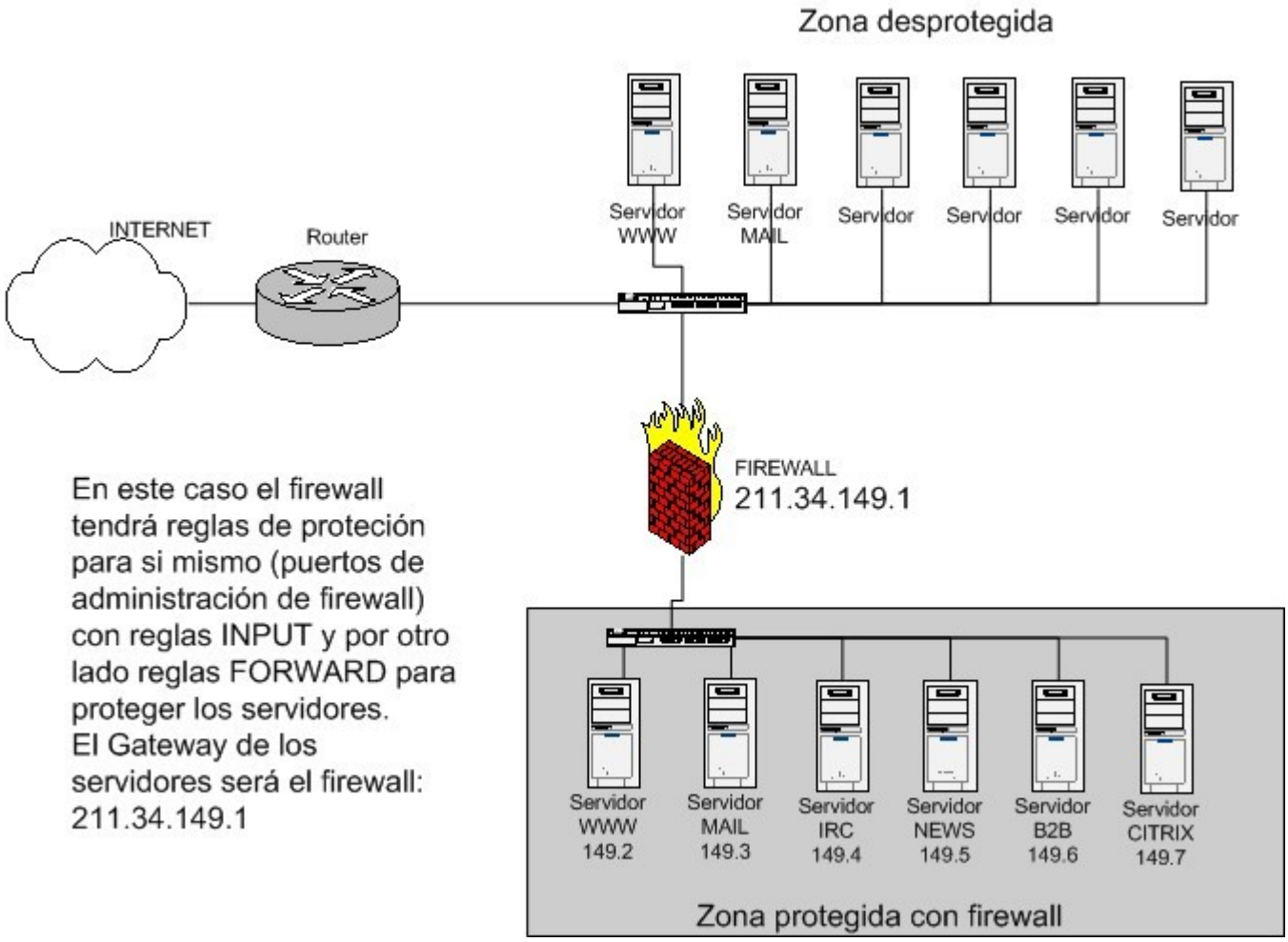
Permitim l'accés des de l'exterior als ports 80 i 443 de DMZ
iptables -A FORWARD -d 212.194.89.152 -p tcp -dport 80 -j ACCEPT
iptables -A FORWARD -d 212.194.89.152 -p tcp -dport 443 -j ACCEPT
iptables -A FORWARD -d 212.194.89.150/30 -j DROP

Tanquem l'accés de la DMZ a la LAN
iptables -A FORWARD -s 212.194.89.152 -d 192.168.10.0/24 -j DROP

Tanquem l'accés de la DMZ al propi firewall
iptables -A INPUT -s 212.194.89.152 -i eth2 -j DROP

4 Firewall entre xarxes utilitzant política DROP

Veurem finalment un exemple en el que configurarem un firewall entre dues xarxes, per tant, ens podem oblidar aquí del cas de xarxa local i de NAT. Només tindrem regles de filtrat INPUT i FORWARD:



El firewall contindrà les regles per protegir els equips que es troben a l'altre costat d'aquest dispositiu, a la xarxa 211.34.149.0/24
Cadascun d'ells dóna un servei determinat, i pot ser gestionat des de distintes IPs; per tant, caldrà donar accés a determinats ports de gestió (22, 3389, etc..).
Veiem com quedarà el script del firewall:

A més, hem dit que aplicarem una política per defecte de denegació. Això implicarà:

- Explicitar cada connexió permesa en tots dos sentits.
- Caldrà conèixer perfectament el que haurà d'estar obert i el que romandrà tancat
- És molt més complicat de mantenir i si es fa, convé fer-ho des del principi
- A més de més feina, suposa un firewall molt més segur

```
#!/bin/sh
## FLUSH de regles
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establim política per defecte: DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

## Comencem a filtrar
## Nota: eth0 és l'interface connectat al router i eth1 a la LAN

# Tenim accés total al nostre firewall des de la nostra IP
iptables -A INPUT -s 210.195.55.15 -j ACCEPT
iptables -A OUTPUT -d 210.195.55.15 -j ACCEPT

# Per a a la resta no hi ha accés al firewall
iptables -A INPUT -s 0.0.0.0/0 -j DROP

## Ara especificarem regles per a cada servidor
## Com seran paquets amb destí a altres màquines s'aplica FORWARD

## Servidor WEB 211.34.149.2
# Accés a port 80
iptables -A FORWARD -d 211.34.149.2 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -s 211.34.149.2 -p tcp --sport 80 -j ACCEPT

# Accés a la nostra ip per gestionar-lo
iptables -A FORWARD -s 210.195.55.15 -d 211.34.149.2 -p tcp --dport 22 -j ACCEPT

iptables -A FORWARD -s 211.34.149.2 -d 210.195.55.15 -p tcp --sport 22 -j ACCEPT

## Servidor MAIL 211.34.149.3
# Accés a ports 25, 110 y 143
iptables -A FORWARD -d 211.34.149.3 -p tcp --dport 25 -j ACCEPT
iptables -A FORWARD -s 211.34.149.3 -p tcp --sport 25 -j ACCEPT

iptables -A FORWARD -d 211.34.149.3 -p tcp --dport 110 -j ACCEPT
iptables -A FORWARD -s 211.34.149.3 -p tcp --sport 110 -j ACCEPT

iptables -A FORWARD -d 211.34.149.3 -p tcp --dport 143 -j ACCEPT
iptables -A FORWARD -s 211.34.149.3 -p tcp --sport 143 -j ACCEPT

# Accés a gestió SNMP
iptables -A FORWARD -s 210.195.55.15 -d 211.34.149.3 -p udp --dport 169 -j ACCEPT

iptables -A FORWARD -s 211.34.149.3 -d 210.195.55.15 -p udp --sport 169 -j ACCEPT

# Accés a la nostra ip per gestionar-lo
iptables -A FORWARD -s 210.195.55.15 -d 211.34.149.3 -p tcp --dport 22 -j ACCEPT

iptables -A FORWARD -s 211.34.149.3 -d 210.195.55.15 -p tcp --sport 22 -j ACCEPT


## Servidor IRC 211.34.149.4
# Accés a ports IRC
iptables -A FORWARD -d 211.34.149.4 -p tcp --dport 6666:6668 -j ACCEPT
iptables -A FORWARD -s 211.34.149.4 -p tcp --sport 6666:6668 -j ACCEPT

# Accés a la nostra ip per gestionar-lo
iptables -A FORWARD -s 210.195.55.15 -d 211.34.149.4 -p tcp --dport 22 -j ACCEPT
```

```
iptables -A FORWARD -s 211.34.149.4 -d 210.195.55.15 -p tcp --sport 22 -j ACCEPT


## Servidor NEWS 211.34.149.5
# Accés a port news
iptables -A FORWARD -d 211.34.149.5 -p tcp --dport news -j ACCEPT
iptables -A FORWARD -s 211.34.149.5 -p tcp --sport news -j ACCEPT

# Accés a la nostra ip per gestionar-lo
iptables -A FORWARD -s 213.194.68.115 -d 211.34.149.5 -p tcp --dport 22 -j ACCEPT

iptables -A FORWARD -s 211.34.149.5 -d 213.194.68.115 -p tcp --sport 22 -j ACCEPT

# Tanquem la resta
iptables -A FORWARD -d 211.34.149.5 -j DROP


## Servidor B2B 211.34.149.6
# Accés a port 443
iptables -A FORWARD -d 211.34.149.6 -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -s 211.34.149.6 -p tcp --sport 443 -j ACCEPT

# Accés a una ip per gestionar-lo
iptables -A FORWARD -s 81.34.129.56 -d 211.34.149.6 -p tcp --dport 3389 -j ACCEPT

iptables -A FORWARD -s 211.34.149.6 -d 81.34.129.56 -p tcp --sport 3389 -j ACCEPT


## Servidor CITRIX 211.34.149.7
# Accés a port 1494
iptables -A FORWARD -d 211.34.149.7 -p tcp --dport 1494 -j ACCEPT
iptables -A FORWARD -s 211.34.149.7 -p tcp --sport 1494 -j ACCEPT

# Accés a una ip per gestionar-lo
iptables -A FORWARD -s 195.55.234.2 -d 211.34.149.7 -p tcp --dport 3389 -j ACCEPT

iptables -A FORWARD -s 211.34.149.7 -d 195.55.234.2 -p tcp --sport 3389 -j ACCEPT

echo " OK . Verifiqui que s'aplica correctament amb: iptables -L -n"

# Final del script
```

Amb aquest script hem aixecat un veritable mur entre internet i el conjunt de servidors que es troba darrere el firewall. No es pot ni fer un ping a les màquines, excepte si donem accés total a una ip.

5. Depuració del funcionament del firewall

Programes útils

IPTRAF. És un dels programes més pràctics per depurar el firewall, ja que permet observar si les connexions s' estableixen o no. És un programa de consola que és aconsellable controlar, ja que mostra en temps real el tràfic que atravessa la nostra màquina amb tot luxe de detalls: origen/destí de ips i ports, tràfic total o tràfic total segons l'interface de xarxa, etc… Si veiem moltes connexions simultànies i ens perdem, permet aplicar filtres per captar només el que ens interessa en cada moment.

NMAP. Eina per escanejar ports per excelència. És una eina de consola ràpida, efectiva i amb multitud d'opcions. Podem utilitzar-la des de màquines externes a la nostra xarxa per comprovar si realment el firewall està filtrant correctament i per fer-nos una idea de la visió que els hackers puguin tenir del nostre sistema.

SHELL. Al propi script del firewall podem afegir algunes opcions per descubrir errors de sintaxi en les regles. Degut a que, com sempre, amb els missatges d'error és molt probable que no sabem interpretar el problema, podem afegir al final de cada regla uns literals de l'estil de:

```
...
iptables -A INPUT -s 195.55.234.2 -j ACCEPT && echo " regla-21 ok"
iptables -A INPUT -s 213.62.89.145 -j ACCEPT && echo " regla-22 ok"
...
```

Si la regla s'executa bé mostrarà el missatge afegit.

També podem jugar a anar afegint i eliminant regles (comentant-les) per esbrinar quina és la regla que ens està provocant l'error, encara que aquesta opció és més primitiva.