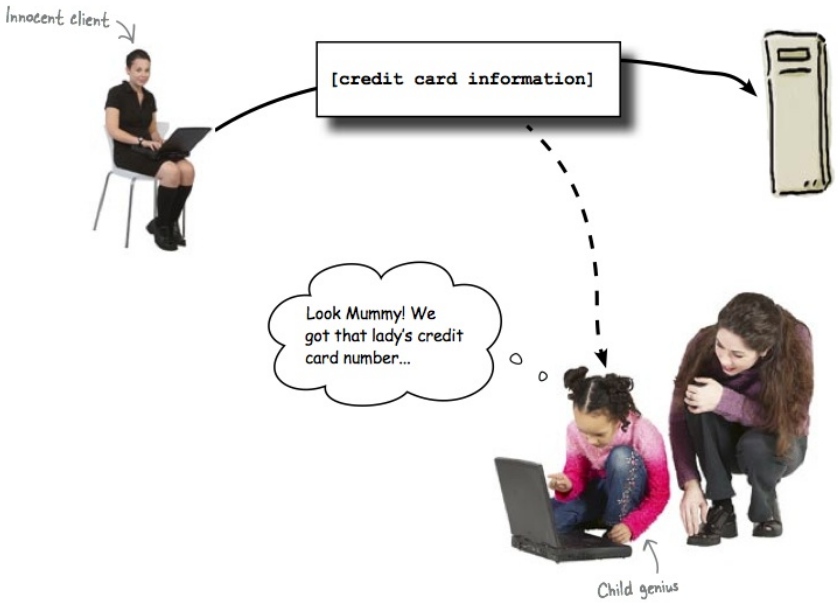
Protejarea retelei impotriva atacurilor

Orice retea trebuie protejata impotriva raufacatorilor care doresc sa fure informatii sau sa atace serverele.



Ce putem face pentru a proteja reteau impotriva unor astfel de persoane ?

1 **Intarirea switch-urilor** : switch-urile sunt vulnerabile la MAC address spoofing si ARP poisoning.

2 **Intarirea ruterelor**

3 **Instalarea unui firewall**

4 **Scrierea si aplicarea unei politici de securitate**

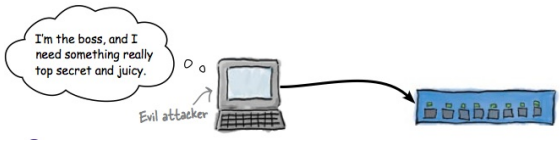
1 MAC address spoofing

Este ceea ce se intampla atunci cand un raufacator isi schimba adresa MAC astfel incat sa coincida ca adresa MAC a unui device din retea. Permite unui raufacator sa pretinda ca hardware-ul sau apartine unei alte persoane - de exemplu a sefului :).

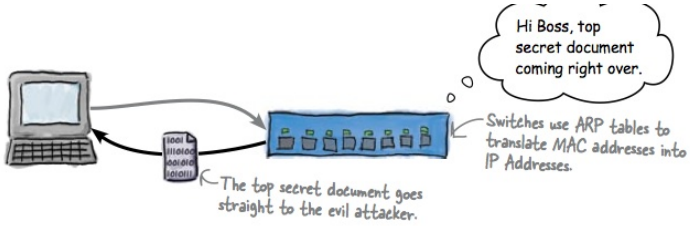
Prin MAC spoofing, un intrus poate fi considerat ca facand parte din retea, pacalind astfel celelalte device-uri sa il lase sa faca trafic sau sa primeasca trafic. Astfel, daca este vorba tocmai de calculatorul sefului, intrusul poate pacali switch-ul sa ii trimita informatii pe care numai seful ar trebui sa le vada.

Exemplu :

1 Intrusul isi modifica adresa MAC, aceasta fiind acum identica cu cea a sefului unei companii, si cere anumite informatii prin retea.



2 switch-ul vede un device cu adresa MAC a sefului care cere informatii, lasandu-l sa aiba acces la informatiile cerute.



Ce ar trebui facut pentru a proteja reteaua ?

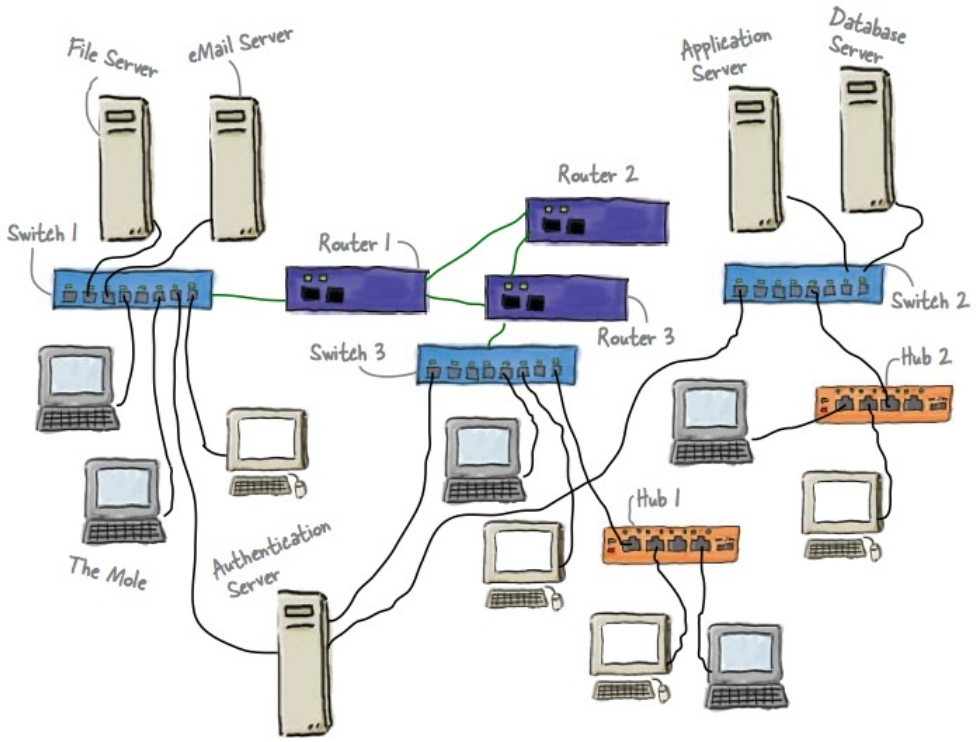


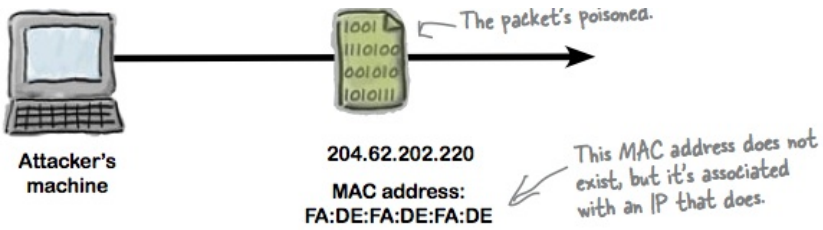
FIGURE 1 – Exemplu de retea vulnerabila. Cum ar trebui proiectata reteaua pentru a-si proteja mai bine resursele impotriva MAC address spoofing ?

O masura de protectie aceea de a aseza serverele in spatele unui ruter care va proteja reteaua impotriva unui astfel de atac, deoarece ele lucreaza cu adrese IP, nu cu adrese MAC cum fac switch-urile. O alta masura de a se proteja intr-o retea de dimensiuni reduse este accea de a setata ce tabele ARP ale swittche-urilor sa fie statice.

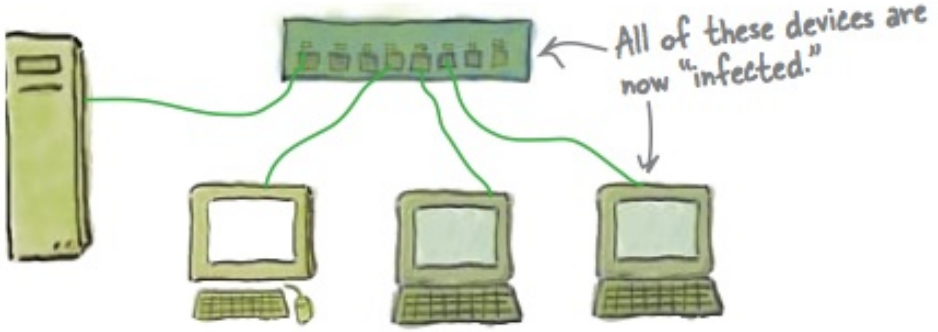
2 ARP poisoning

Un alt atac este ARP poisoning.
Exemplu :

1 **Raufacatorul trimite un pachet "otravit".** El difuzeaza un pachet cu o adresa IP, impreuna cu o adresa MAC care fie este falsa, fie nu exista.



2 **Dispozitivele din retea isi updateaza tabelele ARP, otravindu-le.** Celelalte calculatoare si dispozitive de retea primesc pachetul trimis prin difuzare si isi updateaza tabelele ARP cu informatie eronata. Astfel, aceste dispozitive folosesc in momentul de fata informatie "otravita" sau corupta in mod intentionat.



3 **Atacul continua.** Din moment ce tabele ARP sunt otravite, raufacatorul poate folosi una din urmatoarele metode : Denial of Service, Man in the Middle, or MAC flooding.

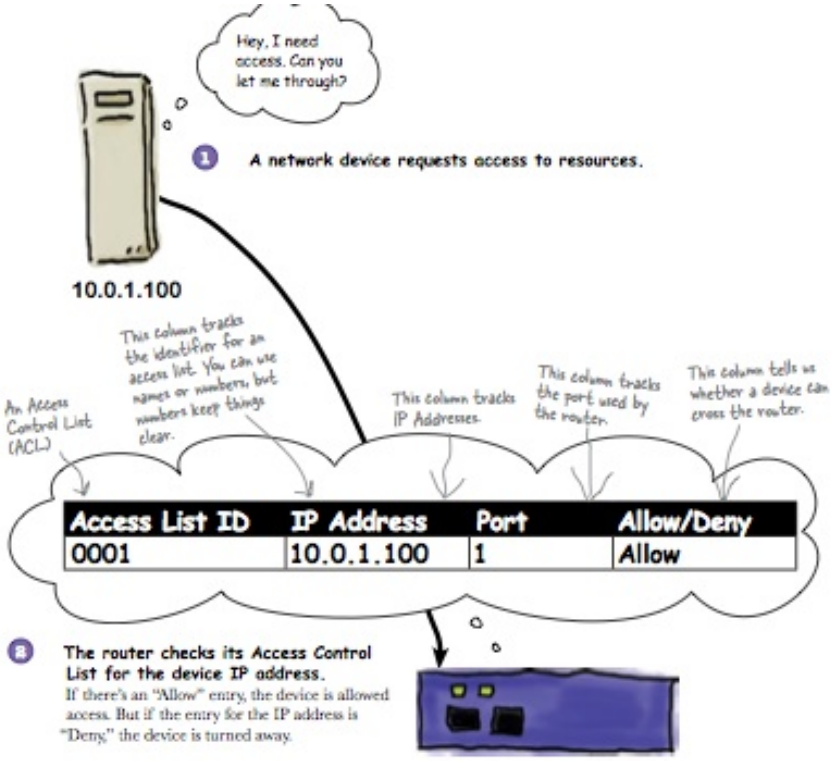
Ce ar trebui facut pentru a preveni un asemenea atac ? Securizarea porturilor. Majoritatea switch-urilor au deja incluse caracteristice de securitate a porturilor. Astfel poti asigna o singura adresa MAC per port. Daca a alta adresa MAC vine pe un alt port decat acela care ii este atribuit, switch-ul nu o va lasa sa treaca.

Intrebari :

De ce MAC spoofing si ARP poisoning nu afecteaza un ruter ?

3 Access Control List (ACL)

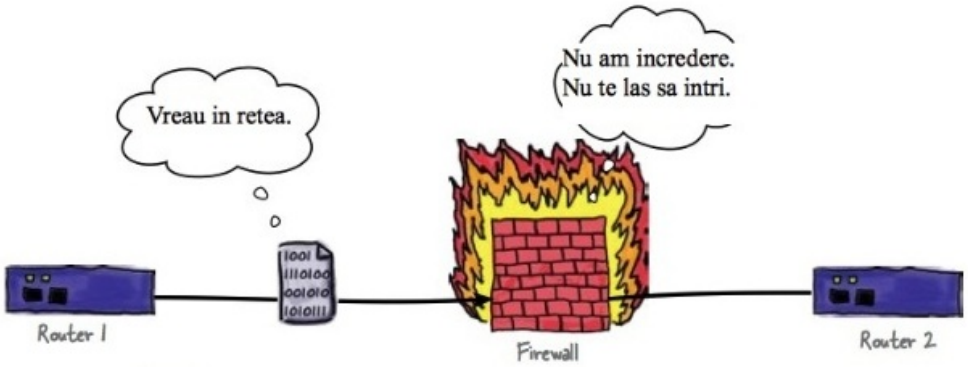
Pentru a spori securitatea retelei trebuie intarita si cea a ruterelor, lucru realizabil folosind lista de control a accesului. ACL-ul este de fapt o tabela in care ruterul patreaza o lista a adreselor IP care au drept sa treaca de ruter. Puteti configura aceasta tabela astfel incat sa permiteti accesul numai anumitor adrese IP.



4 Firewall

Firewall-ul intrerupe libera comunicare intre retele de incredere si cele un–trusted. Filtrarea pachetelor care circula prin retele este facuta prin intermediul unor reguli de acces. Firewall-ul poate fi un dispozitiv harware sau un soft care ruleaza pe un dispozitiv.

Un ruter normal, cat si calculatoarele pe care ruleaza Linux, pot fi setate ca firewall. Exista o multime de aplicatii software care pot oferi o solutie soft de firewall. Scopul acestui laboartor este de a explica modul in care un firewall functioneaza si proteaza reteaua.



Un firewall filtreaza pachetele prin aplicarea unor reguli asemanatoare cu ACL-urile ruterelor. Singura mare diferenta este ca in ACL regula se eplica unor dispozitive IP, pe cand regurile de filtrare a pachetelor ale unui firewall se aplica ... pachetelor.

Allow?	Source	Destination	Protocol
Yes	10.0.1.101	10.0.1.212	TCP

Un firewall utilizeaza filtrare de pachete statica si dinamica.

4.1 Filtrarea statica

1) **Analiza header-ului pachetului** Toata informatia de care are nevoie un firewall pentru a-si pune in aplicare regulile se gaseste in header-ul pachetului. Firewall-ul, la fel ca un politist de frontiera, aduna informatii despre calatorii care for sa ajunga de o parte sau alta a granitei.

2) **Acceptarea sau refuzarea accesului pe baza unei reguli.**

4.2 Filtrarea dinamica

1) **Analiza header-ului pachetului SI administrarea unei tabele de stari.** Si in cazul filtrarii dinamice avem de a face cu o analiza a header-ului pachetului. Totusi filtrarea dinamica este mai inteligenta deoarece se foloseste o tabela de stari care permite sa se tina evidenta pachetelor care au venit si momentul de timp la care acestea au sosit.

2) **Acceptarea sau refuzarea accesului pe baza unei reguli si a starii.**
Exemplu : un pachet adresat hostului local este permis daca acesta reprezinta raspuns la un pachet generat din interior (de hostul local).

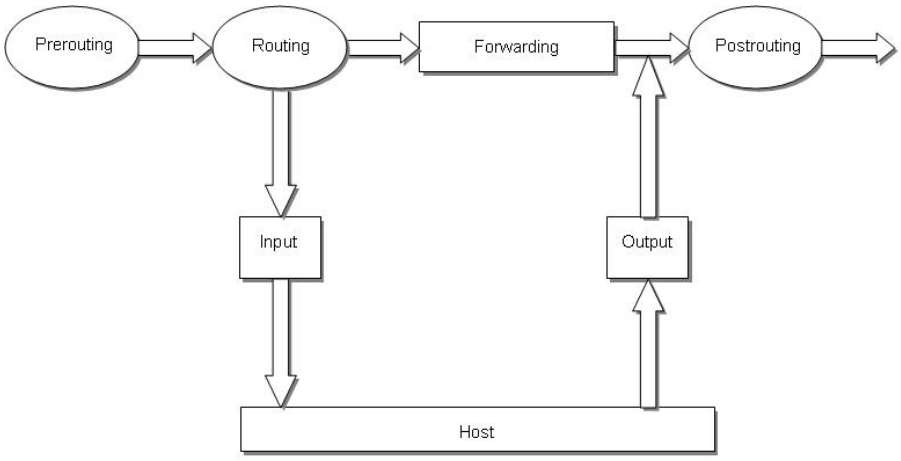
4.3 NETFILTER

In Linux singura modalitatea de a crea un firewall este reprezentata de arhitectura NETFILTER. Aceasta arhitectura foloseste comanda iptables (user space tool) pentru a filtra pachetele de date.
NETFILTER este o tehnologie foarte avansata care permite crearea unui firewall de la zero extrem de eficient care poate fi folosit cu incredere pe servere din mediul Enterprise.

4.4 Structura NETFILTER

4.4.1 Chains

NETFILTER foloseste default 5 chain-uri numite si hooks (carlige) prin care pachetele trec si in care NETFILTER poate interactiona cu acestea. Acestea sunt de fapt puncte de interactiune cu pachetele.



Cele 5 chain-uri predefinite sunt :
1. PREROUTING Acest chain este atins de pachete inaintea procesului de routare, imediat ce driverul placii de retea a acceptat pachetul.
Este folosit pentru :
modificarea headerelor pachetelor (mangling) inaintea procesului de rutare ; Exemplu : modificare TOS (Type of service) sau modificare TTL (Time to live) pentru a influenta procesul de rutare DNAT (Destination NAT sau Port Forwarding) ;

In acest chain trebuie evitata filtrarea fiindca nu toate pachetele trec prin el.

2. INPUT Prin acest chain trec pachetele destinate calculatorului local. Orice pachet care ajunge la statia locala trece prin acesta indiferent pe ce interfata intra sau de unde vine.

Este folosit pentru : modificarea pachetelor (mangling) dupa rutare, dar inainte sa fie trimise procesului local ; filtrarea pachetelor ;

3. OUTPUT Prin acest hook trec pachetele generate de calculatorul local.

Este folosit pentru : filtrare sau manipulare pachete generate de hostul local ;

4. FORWARD Prin acest chain trec pachetele care tranziteaza hostul (acesta a devenit router, leaga minim 2 retele). Pachetele care trec prin chain-ul FORWARD nu sunt destinate hostului local si nici nu sunt generate de acesta.

5. POSTROUTING Ultimul chain prin care trec pachetele, dupa procesul de rutare. Prin acest chain trec atat pachetele care transiteaza hostul daca acesta este Router cat si cele generate de host.

Este folosit pentru : modificarea pachetelor (mangling) dupa procesul de rutare, dar inca pe hostul local ; SNAD (Source NAT & Masquerading) ;

Fiecare din aceste 5 hook-uri/chain-uri (puncte de interactiune cu pachetele) se foloseste pentru a interveni intr-un anume mod asupra pachetelor.

Exista 3 destinatii posibile pentru un pachet functie de care acesta trece prin anumite chain-uri din cele 5 : a) vine din retea si este destinat hostului local ; b) este generat de hostul local ; c) trece prin hostul local ;

Nota : Iptables ofera posibilitatea definirii de chainuri de catre utilizatori pe langa cele 5 chainuri default.

4.4.2 Tabele

Fiecarui hook/chain NETFILTER ii este asociat un set de reguli definite intr-un tabel. In momentul in care un pachet "loveste" un chain acesta este verificat de fiecare regula din tabel. O regula contine criterii care trebuie satisfacute de pachet si un target precum ACCEPT, DROP sau SNAT. Targetul este actiunea intreprinsa daca pachetul satisface regula din tabel. Fiecare regula are un target.

Exemplu :

Cerinta : Dorim sa blocam/dropam toate pachetele care vin catre serverul SSH ce ruleaza pe hostul local si asculta pe portul TCP/22.

Mod realizare : Orice pachet destinat hostului local va trece prin chain-ul INPUT. Acesta va fi si chain-ul in care intervenim pentru droparea/blocarea pachetelor ssh. Intr-un tabel (numit filter) atasat chainului INPUT vom adauga o regula compusa din criterii precum : pachetul este destinat hostului local, iar portul destinatie este 22. Target-ul va fi DROP.

In mod implicit NETFILTER ofera 4 tabele ce contin reguli pentru "prinderea" pachetelor si care se ataseaza de cele 5 chain-uri.

Tabele NETFILTER :

1. filter

Este folosit doar pentru filtrarea pachetelor (ACCEPT sau DROP) si se foloseste doar pe chainurile FORWARD, INPUT sau OUTPUT.

2. nat

Este folosit doar pentru NAT (SNAT si DNAT). Doar primul pachet dintr-un stream va fi procesat de regulile din acest tabel. Asupra celorlalte pachete se va actiona identic. Se poate atasa de chainurile PREROUTING in cazul DNAT (port forwarding) si POSTROUTING in cazul SNAT.

3. mangle

Este folosit pentru manipularea/modificarea pachetelor si anume modificarea headerelor de Layer3 si Layer4 (modificare tos, ttl etc). Acest tabel poate fi atasat de orice chain.

4. raw

Se foloseste doar pentru marcarea pachetelor care nu trebuie sa fie procesate de "connection tracking system". Tabelul se poate folosi doar pentru chainurile PREROUTING si/sau OUTPUT. Mecanismul de "connection tracking" este consumator de resurse, astfel pentru un anumit tip de trafic se poate opri connection tracking system.

Exemplu : excluderea traficul generat pentru localhost.

Nota : Cele 4 tabele descrise mai sus sunt implicite. Acestea nu pot fi sterse si nici alte tabele nu pot fi create.

4.4.3 The state machine

Connection tracking este componenta NETFILTER care ofera acestuia statutul de firewall stateful. Acesta poate lua decizii de filtrare a pachetelor nu in functie de headerul Layer3 (IP) si Layer4 (TCP/UDP) ci in functie de relatia pachetului cu celelalte pachete.

Connection tracking este realizat de un framework din kernel care se numeste conntrack. Acesta poate fi incarcat ca modul sau poate fi parte integranta a kernelului.

conntrack reprezinta o parte din NETFILTER care identifica pachetele ca aflandu-se intr-o anume stare in functie de relatia cu celelalte pachete din acelasi stream.

NETFILTER defineste 4 stari pentru fiecare pachet :

1. NEW

Primul pachet dintr-o conexiune generat de hostul local se gaseste in starea NEW.

2. ESTABLISHED

Pachetul destinat hostului local ca raspuns la pachetul trimis anterior isi schimba starea in ESTABLISHED in momentul in care intra in PREROUTING. Sunt toate pachetele dintr-o conexiune mai putin primul care a initiat conexiunea si care se afla in starea NEW.

3. RELATED

In starea RELATED se gasesc acele pachete legate de un alt flux de date. **Exemplu :** in cazul FTP activ, conexiunea de date de pe portul 20 ca raspuns la conexiunea de control initiata catre portul 21

4. INVALID

Sunt acele pachete ale caror header contine informatii neconcordante. **Exemplu :** un pachet al carui header TCP contine atat flag-ul syn cat si fin

Informatiile pe care modulul conntrack le foloseste pentru a sti in ce stare se gaseste un pachet, pot fi vizualizate in $/proc/net/nf_conntrack(FC9)$

tcp 6 117 SYN_SENT src=192.168.1.6 dst=192.168.1.9 sport=32775
dport=22 [UNREPLIED] src=192.168.1.9 dst=192.168.1.6 sport=22
dport=32775 [ASSURED] use=2

Detalii :
tcp - protocolul de transport ;
6 - valoarea campului protocolului din headerul IP ;
117 - nr. de secunde in care aceasta intrare este valida. Timpul este decrementat continuu pana cand apare trafic legat de aceasta conexiune. Apoi timpul este resetat cu valoarea default ;
SYN_SENT- trafic doar intr-o directie ;
src - ip source ;
dst - ip destinatie ;
sport - port sursa ;
dport - port destinatie ;
UNREPLIED - nu a existat trafic in ambele directii. In momentul in care apare trafic in ambele directii UNREPLIED se inlocuieste cu ASSURED ;

ASSURED (la final) - informatii despre aceasta conexiune nu vor fi sterse cand se atinge nr. maxim de conexiuni ;

4.4.4 Structura iptables

Commanda iptables (user space tool) se foloseste pentru a comunica cu NETFILTER.

Important

1. Scopul comenzii iptables este de a adauga, sterge, inlocui, lista, vizualiza etc reguli din cele 4 tabele standard care sunt atasate de cele 5 chainuri.

2. In mod default nu exista nicio regula in tabele, acestea fiind goale. Implicit nu exista firewall.

3. Un pachet traverseaza in mod secvential regulile din tabelele atasate chainurilor pana in momentul in care o regula "prinde" pachetul, caz in care se executa TARGET-ul regulii. Restul regulilor din tabel nu se mai verifica ulterior.

4. Daca pachetul nu este prins de nicio regula din tabel se executa politica default (-P POLICY) care este implicit ACCEPT.

Structura comenzii iptables este :

iptables -t *nume_tabel* -*OPERATIE_ASUPRA_CHAIN**NUME_CHAIN* -criterii -j TARGET

unde :

Nume tabel :

filter
nat
raw
mangle

Nota : Numele tabelului in care se adauga regula trebuie scris cu litera mica. Daca se omite numele tabelului acesta este default filter.

Exemplu :

iptables -t filter -A INPUT -p tcp –dport 80 -j ACCEPT
este echivalent cu :
iptables -A INPUT -p tcp –dport 80 -j ACCEPT

Operatii asupra unui chain

-A -¿ adaugare regula la sfarsitul tabelului atasat chainului ;
-I -¿ adaugare regula pe prima pozitie in tabelul atasat chainului ;
-L -¿ listare reguli ;
-P -¿ policy, actiunea default care se executa daca nicio regula nu prinde pachetul ;
-N -¿ creare chain nou definit de utilizator ;
-X -¿ sterge chain definit de utilizator ;
-F -¿ flush, goleste regulile din tabelul atasat chainului ;
-Z -¿ zero, reset counters ;

Nota : Operatiile asupra regulilor din tabelele atasate chainurilor trebuie scrise cu litera mare.

Exemplu :

Adauga o regula la sfarsitul tabelului filter (default daca nu se specifica) pentru chain-ul OUTPUT care permite trimiterea de pachete catre IP-ul din spatele domeniului www.google.ro
1. iptables -A OUTPUT -d www.google.ro -j ACCEPT

Adauga policy DROP pentru INPUT. Orice pachet destinat hostului local care nu este acceptat de nicio regula din tabelul filter de pe chainul INPUT este dropat
2. iptables -P INPUT DROP

Sterge toate regulile din tabelul filter (default daca nu se specifica) de pe chainul FORWARDING
3. iptables -F FORWARDING

Nume Chain :

PREROUTING
INPUT
OUTPUT
FORWARD
POSTROUTING

Exemplu :
Dropeaza toate pachetele generate de hostul local catre orice server http care daca acesta asculta pe portul 80
iptables -A OUTPUT -p tcp –dport 80 -j DROP

Criterii
-s IP_sursa
Exemplu : -s 80.0.0.1 sau -s 192.168.0.0/24 sau -s 0/0. Specifica IP-ul sursa din pachet. 0/0 inseamna orice IP

-d IP_dest
Exemplu : -d 182.0.10.1 sau -d 10.10.0.0/26 sau -d 0/0 -¿ specifica IP-ul destinatie din pachet. 0/0 inseamna orice IP

-p protocol
Exemplu : -p tcp sau -p udp sau -p icmp

–sport $port\_sursa$
Exemplu : iptables -I INPUT -p udp –sport 53 -j DROP -¿ dropeaza toatea pachetele UDP care sunt destinante hostului local si vin de la un server DNS (port 53)

–dport $port\_dest$
Exemplu : iptables -A FORWARD -p tcp –dport 8080 -j DROP -¿ dropeaza toata pachetele catre portul tcp 8080 care tranziteaza ruterul linux

-i $interfata\_in$
Exemplu : iptables -A INPUT -i eth0 -j ACCEPT -¿ accepta toate pachetele destinate hostului local care intra pe interfata eth0

-o $interfata\_out$
Exemplu : iptables -t mangle -A OUTPUT -o eth1 -j TTL –ttl-set 67 -¿ modifica TTL-ul din headerul IP setand valoarea 67 pentru toate pachetele generate de hostul local care ies pe interfata eth1

Target

Specifica actiunea intreprinsa asupra pachetului daca criteriile sunt indeplinite.
ACCEPT -¿ pachetul este acceptat ;
DROP -¿ pachetul este dropat ;
REJECT -¿ pachetul este rejectat si hostul raspunde cu un mesaj de eroare sursei ;
LOG -¿ logheaza/salveaza informatii despre pachet intr-un fisier ;

LIMIT -¿ limiteaza nr. de pachete pe unitatea de timp ;
SNAT -¿ realizeaza source nat ;
MASQUERADE -¿ realizeaza source nat ;
DNAT -¿ realizeaza destination nat/port forwarding ;
TTL -¿ modifica TTL din pachet (headerul IP) ;

4.4.5 Exemple

Pentru a crea un firewall folosind NETFILTER, singura modalitate in Linux, exista mai multe posibilitati :

1. Folosind scripturi deja create precum firestarter Aceste scripturi contin o sectiune de configurare unde adminul seteaza modul in care firewall-ul va opera (alege porturile pe care asculta severele locale si catre care se pot conecta utilizatorii, seteaza IP-urile de la care se accepta conexiuni etc).

Scriptul odata rulat genereaza in spate comenzi iptables. Modalitatea este recomandata pentru adminii incepatori care nu doresc un control total al modului in care firewall-ul opereaza.

2. Folosind aplicatii grafice disponibile in GNOME sau KDE. Scopul cursului este de a forma administratori de sistem pentru administrarea de servere. Serverele Linux ruleaza in runlevel 3 fara mod grafic, iar intreaga administrare se face remote folosind ssh din consola in mod text.

3. Creand propriul firewall de la zero. Aceasta este modalitatea cea mai recomandata. Chiar daca la un moment dat vom folosi din anumite considerente sau politici stabile un script general gata facut, pentru o buna intelegere a modului in care NETFILTER functioneaza orice admin trebuie sa creeze si sa testeze un firewall de la zero.

Firewall basic

In Linux exista o interfata virtuala numita Loopback (prescurtata lo) care poate fi vizualizata folosind ifconfig. Este foarte important ca traficul care paraseste interfata de loopback precum si traficul destinat interfetei de loopback sa fie permis de firewall. Altfel sistemul devine instabil, iar anumite procese nu mai pot functiona corect. Exemplu : serverul grafic X. Interfata de loopback are ip-ul 127.0.0.1 si numele localhost. Este folosita de catre procesele client-server care ruleaza pe acelasi host.

Exemplu permitere trafic pentru loopback : iptables -A INPUT -i lo -j ACCEPT iptables -A OUTPUT -o lo -j ACCEPT

1. Se doreste crearea unui firewall stateful pentru sistem Linux folosit ca Desktop. Pe acesta nu ruleaza servere, iar utilizatorul poate comunica cu orice serviciu extern.
Cerinte :
hostul poate genera orice fel de trafic TCP, UDP sau ICMP catre orice IP extern ;
pachetele destinate hostului sunt acceptate doar daca reprezinta raspuns la traficul generat din interior ;
pachetele care reprezinta initializarea unei conexiuni din exterior catre interior sunt filtrate ;

2. Filtrarea dupa MAC
Cerinte :
se doreste acceptarea de pachete doar de la un singur MAC
scenariu poate fi util cand se doreste limitarea hosturilor cu care poate comunica un server in LAN, sau comunicarea doar cu default gateway si deci doar pe Internet

3. Listarea unui firewall sau verificare firewall care ruleaza. Comanda se executa direct in consola.
iptables -vnL
Outputul acestei comenzi ne indica :
policy de pe fiecare CHAIN

regulile din tabelele atasate chain-urilor
nr de pachete prinse de fiecare regula

Comanda de mai sus listeaza toate regulile din tabelul filter (este default) de pe toate chainurile

Daca se doreste listarea regulilor din tabelul nat sau mangle se foloseste :

iptables -t nat -vnL

Tema :

1. Blocarea conexiunilor catre host cu exceptia SSH. Traficul outbound si raspunsul la acesta este permis.
2. Logarea de pachete

Cerinta :

se doreste logarea tuturor pachetele de tip HTTP care sunt generate de host in vederea analizarii ulterioare a site-urilor vizitate sau pentru analiza continutului headerelor acestora.

3. Se doreste crearea unui firewall pentru un server din LAN.

Cerinte :

- pe host ruleaza server ssh (tcp/22), http (tcp/80), https (tcp/443), smtp (tcp/25), pop (tcp/110), imap (tcp/143) si dns (dns/53). Toate aceste servicii si doar acestea trebuie sa fie accesibile userilor din LAN
- firewall-ul trebuie sa dropeze si sa logheze pachetele invalide
- serverul poate accesa pe internet doar servere web (pentru update) si dns (pentru query iterative sau pentru a folosi un forwarder)