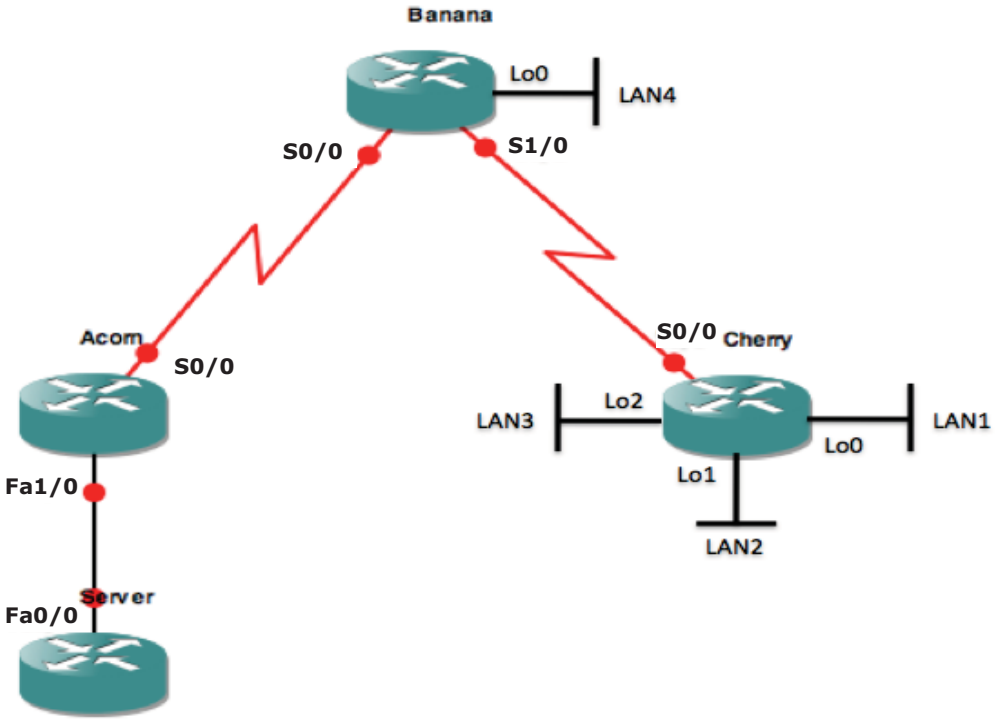
Laborator ACL

Topologie:



Schema de adresare:

Subnetati optim:

10.1.1.0/24	–	conexiunile *seriale* dintre routere -> numar minim de adrese
80.11.1.0/24	–	reteaua Acorn-Server -> 250 hosturi
192.168.1.0/23	–	LAN1, LAN2, LAN3 -> 100 hosturi fiecare
172.16.15.192/27	–	LAN4 -> 15 hosturi

Taskuri:

1. Configurari initiale:
   – hostname-uri si adrese IP conform topologiei date;
   – rute statice necesare sau un protocol de rutare la alegere astfel incat sa existe conectivitate end-to-end

2. Politici de securitate:

a) Creati un ACL standard ce opreste tot traficul provenit din LAN2 catre server. Asigurati-va ca routerul va afisa un mesaj de fiecare data cand un pachet este blocat.
Hint: Testati utilizand un ping de pe interfata de loopback a lui Cherry.

b) Asigurati-va ca nici una din retelele locale de pe Cherry nu pot comunica cu LAN4 de pe Banana.

c) Utilizati una sau mai multe ACL-uri pentru a opri accesul router-ului Cherry prin telnet la Acorn si Banana.

d) Rulati serviciul de server web pe Acorn. Testati-l utilizand telnet de pe oricare alt router din topologie (trimiteti un GET request). Blocati tot traficul de HTTP si HTTPS trimis din LAN3 catre server-ul de web.
Hint: Testati utilizand telnet de pe o anumita interfata catre un anumit port.

e) Creati un ACL astfel incat LAN4 sa nu accepte conexiuni TCP initiate din exterior, dar sa poata accesa in mod normal exteriorul.

f) Creati un ACL care sa contina o singura intrare pentru liniile VTY ale router-ului Acorn astfel incat *doar* pachetele cu IP sursa impar sa il poata accesa print telnet.

g) Creati un ACL care sa blocheze explicit cele doua mesaje implicate in procesul de testare al conectivitatii dintre Acorn si Banana, astfel incat *ping*-ul intre aceste doua echipamente sa nu functioneze.
Hint: Aveti in vedere ca si Banana si Acorn au mai multe interfete active.

h) Creati un ACL reflexiv pentru tot traficul ICMP dintre Acorn si Cherry.

Observatie: Orice trafic care nu este blocat explicit prin aceste cerinte va fi permis!

Rezolvari:

1. Configurari initiale:
   – hostname-uri si adrese IP conform topologiei date;
   – rute statice necesare sau un protocol de rutare la alegere astfel incat sa existe conectivitate end-to-end

```
Cherry(config)# interface Loopback0
Cherry(config-if)# ip address 192.168.0.1 255.255.255.128
Cherry(config-if)# interface Loopback1
Cherry(config-if)# ip address 192.168.0.129 255.255.255.128
Cherry(config-if)# interface Loopback2
Cherry(config-if)# ip address 192.168.1.1 255.255.255.128
Cherry(config-if)# interface Serial0/0
Cherry(config-if)# ip address 10.1.1.6 255.255.255.252
Cherry(config-if)# no shut
Cherry(config)# ip route 0.0.0.0 0.0.0.0 se0/0
Banana(config)# interface Loopback0
Banana(config-if)# ip address 172.16.15.193 255.255.255.224
Banana(config-if)# interface Serial0/0
Banana(config-if)# ip address 10.1.1.2 255.255.255.252
Banana(config-if)# no shut
Banana(config-if)# interface Serial0/1
Banana(config-if)# ip address 10.1.1.5 255.255.255.252
Banana(config-if)# no shut
Banana(config)# ip route 80.11.1.0 255.255.255.0 Serial0/0
Banana(config)# ip route 192.168.0.0 255.255.254.0 Serial0/1
Acorn(config)# interface Serial0/0
Acorn(config-if)# ip address 10.1.1.1 255.255.255.252
Acorn(config-if)# no shut
Acorn(config-if)# interface FastEthernet1/0
Acorn(config-if)# ip address 80.11.1.1 255.255.255.0
Acorn(config-if)# no shut
Acorn(config)# ip route 0.0.0.0 0.0.0.0 Se0/0
Server(config)# interface FastEthernet0/0
Server(config)# ip address 80.11.1.2 255.255.255.0
Server(config)# ip route 0.0.0.0 0.0.0.0 80.11.1.1
```

2. Politici de securitate

a) Creati un ACL standard ce opreste tot traficul provenit din LAN2 catre server. Asigurati-va ca routerul va afisa un mesaj de fiecare data cand un pachet este blocat.

Hint: Testati utilizand un ping de pe interfata de loopback a lui Cherry.

```
Cherry#ping 80.11.1.2 source 192.168.0.129
Packet sent with a source address of 192.168.0.129
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 16/41/76 ms

Server(config)# access-list 10 deny 192.168.0.128 0.0.0.127 log
Server(config)# access-list 10 permit any
Server(config)# int Fa0/0
Server(config-if)# ip access-group 10 in

Cherry#ping 80.11.1.2 source 192.168.0.129
Packet sent with a source address of 192.168.0.129
U.U.U
Success rate is 0 percent (0/5)

Server#
*Mar  1 00:26:09.847: %SEC-6-IPACCESSLOGNP: list 10 denied 0 192.168.0.129 -> 80.11.1.2,
1 packet
```

b) Asigurati-va ca nici una din retelele locale de pe Cherry nu pot comunica cu LAN4 de pe Banana.

```
Banana(config)# ip access-list extended BsiD
Banana(config-ext-nacl)# deny ip 192.168.0.0 0.0.1.255 172.16.15.192 0.0.0.31
Banana(config-ext-nacl)# permit ip any any
Banana(config)# int s0/1
Banana(config-if)# ip access BsiD in

Cherry#ping 172.16.15.193 source 192.168.0.1
U.U.U
Cherry#ping 172.16.15.193 source 192.168.0.129
U.U.U
Cherry#ping 172.16.15.193 source 192.168.1.1
U.U.U
```

c) Utilizati una sau mai multe ACL-uri pentru a opri accesul router-ului Cherry prin telnet la Acorn si Banana.

Se rezolva identic pentru Acorn si pentru Banana:

```
Banana(config)# access-list 10 deny   10.1.1.6
Banana(config)# access-list 10 deny   192.168.1.1
Banana(config)# access-list 10 deny   192.168.0.1
Banana(config)# access-list 10 deny   192.168.0.129
Banana(config)# access-list 10 permit any
Banana(config)# line vty 0 4
Banana(config)# access-class 10 in
Cherry#telnet 10.1.1.2 /source-interface lo2
Trying 10.1.1.2 ...
% Connection refused by remote host
```

d) Rulati serviciul de server web pe Acorn. Testati-l utilizand telnet de pe oricare alt router din topologie (trimiteti un GET request). Blocati tot traficul de HTTP si HTTPS trimis din LAN3 catre server-ul de web.
Hint: Testati utilizand telnet de pe o anumita interfata catre un anumit port.
Avem deja un access-list pe interfata s0/1 pe directia in pe Banana. Trebuie sa il editam (datorita regulii celor 3P).

```
Cherry#telnet 10.1.1.1 80 /source-interface loopback2
Trying 10.1.1.1, 80 ... Open
HTTP/1.1 400 Bad Request
Banana(config)# do sh access-list
......
Extended IP access list BsiD
    10 deny ip 192.168.0.0 0.0.1.255 172.16.15.192 0.0.0.31 (33 matches)
    20 permit ip any any
......
Banana(config)# ip access-list extended BsiD
Banana(config-ext-nacl)# 3 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq 80
Banana(config-ext-nacl)# 4 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq 443
Banana(config-ext-nacl)# 5 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq 80
Banana(config-ext-nacl)# 6 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq 443
Cherry# telnet 10.1.1.1 80 /source-interface Lo2
Trying 10.1.1.1, 80 ...
% Destination unreachable; gateway or host down
Cherry# telnet 10.1.1.1 80
Trying 10.1.1.1, 80 ... Open
```

e) Creati un ACL astfel incat LAN4 sa nu accepte conexiuni TCP initiate din exterior, dar sa poata accesa in mod normal exteriorul.

```
Banana(config)#ip access-list extended ESTABL
Banana(config-ext-nacl)# permit tcp any 172.16.15.192 0.0.0.31 established
Banana(config-ext-nacl)# deny   tcp any 172.16.15.192 0.0.0.31
Banana(config-ext-nacl)# permit ip any any
Banana(config-if)#int s0/0
Banana(config-if)#ip access ESTABL in
Banana(config-ext-nacl)# do telnet 10.1.1.1 /source-interface lo0
Trying 10.1.1.1 ... Open
Acorn# telnet 172.16.15.193
Trying 172.16.15.193 ...
% Destination unreachable; gateway or host down


Banana(config)# ip access-list extended BsiD
Banana(config-ext-nacl)# 7 permit tcp any 172.16.15.192 0.0.0.31 established
Banana(config-ext-nacl)# 8 deny tcp any 172.16.15.192 0.0.0.31
Banana(config-ext-nacl)# do sh ip access-list
Standard IP access list 10
    ......
Extended IP access list BsiD
    3 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq www (3 matches)
    4 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq 443
    5 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq www
    6 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq 443
    7 permit tcp any 172.16.15.192 0.0.0.31 established
    8 deny tcp any 172.16.15.192 0.0.0.31
    10 deny ip 192.168.0.0 0.0.1.255 172.16.15.192 0.0.0.31 (33 matches)
    20 permit ip any any (7 matches)
Cherry# telnet 172.16.15.193
Trying 172.16.15.193 ...
% Destination unreachable; gateway or host down
Banana(config-ext-nacl)# do sh ip access-list
.....
  7 permit tcp any 172.16.15.192 0.0.0.31 established
  8 deny tcp any 172.16.15.192 0.0.0.31 (3 matches)
```

f) Creati un ACL care sa contina o singura intrare pentru liniile VTY ale router-ului Acorn astfel incat *doar* pachetele cu IP sursa impar sa il poata accesa print telnet.

```
Acorn(config)# ip access-list standard 10
Acorn(config-std-nacl)# 50 deny 0.0.0.0 255.255.255.254
Acorn(config-std-nacl)# 60 permit any
Banana(config)#do telnet 80.11.1.1 /source-interface se0/0
Trying 80.11.1.1 ...
% Connection refused by remote host

Banana(config)#do telnet 80.11.1.1 /source-interface se0/1
Trying 80.11.1.1 ... Open
```

g) Creati un ACL care sa blocheze explicit cele doua mesaje implicate in procesul de testare al conectivitatii dintre Acorn si Banana, astfel incat *ping*-ul intre aceste doua echipamente sa nu functioneze.
Hint: Aveti in vedere ca si Banana si Acorn au mai multe interfete active.

```
Acorn(config)#ip access extended NOBANANAPING
Acorn(config-ext-nacl)#deny icmp host 10.1.1.2 host 10.1.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 10.1.1.2 host 10.1.1.1 echo-reply
Acorn(config-ext-nacl)#deny icmp host 10.1.1.2 host 80.11.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 10.1.1.2 host 80.11.1.1 echo-reply
Acorn(config-ext-nacl)#deny icmp host 10.1.1.5 host 10.1.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 10.1.1.5 host 10.1.1.1 echo-reply
Acorn(config-ext-nacl)#deny icmp host 10.1.1.5 host 80.11.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 10.1.1.5 host 80.11.1.1 echo-reply
Acorn(config-ext-nacl)#deny icmp host 172.16.15.193 host 10.1.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 172.16.15.193 host 10.1.1.1 echo-reply
Acorn(config-ext-nacl)#deny icmp host 172.16.15.193 host 80.11.1.1 echo
Acorn(config-ext-nacl)#deny icmp host 172.16.15.193 host 80.11.1.1 echo-reply
Acorn(config-ext-nalc)#permit ip any any
Acorn(config-ext-nacl)#int s0/0
Acorn(config-if)#ip access NOBANANAPING in

Banana# ping 80.11.1.1 source se0/1
U.U.U
Banana# ping 80.11.1.1 source se0/0
U.U.U
Banana# ping 80.11.1.1 source lo0
U.U.U
Banana# ping 10.1.1.1 source se0/1
U.U.U
Banana# ping 10.1.1.1 source se0/0
U.U.U
Banana# ping 10.1.1.1 source lo0
U.U.U

Acorn#ping 10.1.1.2 source fa1/0
.....
Acorn#ping 10.1.1.2 source se0/0
.....
Acorn#ping 10.1.1.5 source fa1/0
.....
Acorn#ping 10.1.1.5 source se0/0
.....
Acorn#ping 172.16.15.193 source fa1/0
.....
Acorn#ping 172.16.15.193 source fa1/0
.....
```

h) Creati un ACL reflexiv pentru tot traficul ICMP dintre Acorn si Cherry.

```
Banana(config)#do sh access-list
Standard IP access list 10
    ....
Extended IP access list BsiD
    3 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq www (6 matches)
    4 deny tcp 192.168.1.0 0.0.0.127 host 10.1.1.1 eq 443
    5 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq www
    6 deny tcp 192.168.1.0 0.0.0.127 host 80.11.1.1 eq 443
    7 permit tcp any 172.16.15.192 0.0.0.31 established (18 matches)
    8 deny tcp any 172.16.15.192 0.0.0.31 (3 matches)
    10 deny ip 192.168.0.0 0.0.1.255 172.16.15.192 0.0.0.31 (44 matches)
    20 permit ip any any (24 matches)
```

ACL-ul trebuie pus pe Banana fiindca Cherry si Acorn nu se pot autocenzura. Am ales sa punem noile regulile in ACL-ul de pe interfata dinspre Cherry si nu pe cea dinspre Acorn. De ce? Deoarece pachetele trimise de Cherry care nu sunt raspunsuri la cereri ale lui Acorn pot fi aruncate la intrarea pe Banana, inainte de a fi rutate catre Acorn.

Trebuie sa mutam regula 10 a ACL BsiD la un numar de ordine mai mare pentru a face loc regulilor de evaluate. Putem sa punem regulile de evaluare icmp si inainte de tcp, doar ca ar insemna sa mutam mai multe reguli la numere de ordine mai mari. Din acest motiv, este bine sa scrieti orice ACL in Notepad – este mai usor si mai rapid de sters un ACL in totalitate, de modificat in Notepad si de copiat inapoi in terminal (atunci cand sunt mai multe reguli de inserat). Alternativa este sa va ganditi de la bun inceput ce aveti de facut ptr toate cerintele si sa implementati integral un ACL necesar.

```
Banana(config)#ip access-list extended REFLECTICMP
Banana(config-ext-nacl)#permit icmp host 10.1.1.1 host 10.1.1.6 reflect SER1
Banana(config-ext-nacl)# permit icmp host 10.1.1.1 host 10.1.1.6 reflect SER1
Banana(config-ext-nacl)# permit icmp host 10.1.1.1 host 192.168.0.1 reflect LAN11
Banana(config-ext-nacl)# permit icmp host 10.1.1.1 host 192.168.0.129 reflect LAN12
Banana(config-ext-nacl)# permit icmp host 10.1.1.1 host 192.168.1.1 reflect LAN14
Banana(config-ext-nacl)# permit icmp host 80.11.1.1 host 10.1.1.6 reflect SER2
Banana(config-ext-nacl)# permit icmp host 80.11.1.1 host 192.168.0.1 reflect LAN21
Banana(config-ext-nacl)# permit icmp host 80.11.1.1 host 192.168.0.129 reflect LAN22
Banana(config-ext-nacl)# permit icmp host 80.11.1.1 host 192.168.1.1 reflect LAN23
Banana(config-ext-nacl)#permit ip any any

Banana(config)#ip access-list extended BsiD
Banana(config-ext-nacl)#no 10
Banana(config-ext-nacl)#18 deny ip 192.168.0.0 0.0.1.255 172.16.15.192 0.0.0.31
Banana(config-ext-nacl)#9 evaluate SER1
Banana(config-ext-nacl)#10 evaluate LAN11
Banana(config-ext-nacl)#11 eval LAN12
Banana(config-ext-nacl)#12 eval LAN13
Banana(config-ext-nacl)#13 eval SER2
Banana(config-ext-nacl)#14 eval LAN21
Banana(config-ext-nacl)#15 eval LAN22
Banana(config-ext-nacl)#16 eval LAN23

Banana(config)# int se0/1
Banana(config-if)# ip access-group BsiD in
Banana(cinfig-if)# ip access-group REFLECTICMP out
```

La sfarsitul configurarilor, este recomandat sa rulati din nou comenzile de verificare date pe parcursul rezolvarii. Scopul este de a testa daca ati respectat cerintele si daca ordinea regulilor in cadrul unui ACL la care ati adaugat reguli pe parcurs nu a afectat functionarea lor.