Assign a name

```
switch(config)# hostname S1
```

Secure access to console line

```
S1(config)# line console 0
S1(config-line)# password letmein
S1(config-line)# login
S1(config-line)# exit
```

<=or=>

```
S1(Config)# line vty 0 15
S1(Config-line)# password cisco
S1(Config-line)# login
S1(Config-line)# exit
```

Secure privileged mode access (EXEC)

```
S1(config)# enable password c1$co
S1(config)# exit
```

Encrypted password to secure access to privileged mode

```
S1(config)# enable secret itsasecret
S1(config)# exit
```

Note: The `enable password` should be replaced with the newer encrypted secret password using the `enable secret` command.

Encrypt the enable and console password

```
S1(config)# service password-encryption
S1(config)# exit
```

MOTD Banner(Message of the day)

```
S1(config)# banner motd "Authorized Access Only"
S1(config)# exit
```

Save Configuration Files to NVRAM

```
S1# copy running-config startup-config
```

→ to ensure that the changes made are not lost if the system is rebooted or looses power.

Configure switch with an IP address

```
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.253 255.255.255.0
S1(config-if)# no shutdown
```

Show Interfaces

```
Router> show ip interface brief
Router> show interface gi0/0
Router> show interface se0/0/0
```

→ **Note:** Bandwith on a serial interfaces is used by routing processes to determine the best path to a destination.

To get Default Gateway from Console

```
R1# show ip interface brief
```

Configure Default Gateway

```
S1(config)# ip default-gateway 128.107.30.1
```

Enable the router to forward IPv6 packets

R1(config)# ipv6 unicast-routing

Configure IPv6

R1(config)# interface gi0/0

R1(config-if)# ipv6 address 2001:DB8:1:1::1/64

R1(config-if)# ipv6 address FE80::1 link-local

ip nat inside source static tcp 192.168.1.1 80 15.15.15.2 80

sh ip nat translations

* no nat in switch layer 3 → only with vlan

Configure EIGRP

R1(config)# router eigrp 1

R1(config-router)# eigrp router-id 1.1.1.1

R1(config-router)# net 172.16.1.0

R1(config-router)# net 172.16.3.0

R1(config-router)# net 192.68.10.4

R1# sh ip eigrp topology

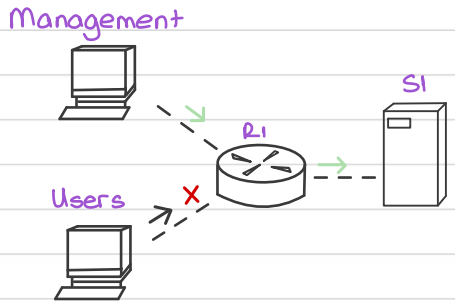
Study Title ACL

WHAT ARE ACLS?

ACLs are set of rules used most commonly to filter network traffic. They are used on network devices with packet filtering capatibilities (e.g. routers or firewalls) ACLs are applied on the interfaces basis to packets leaving or entering an interface.

TYPES OF ACLS

1. Standard access list – you can filter only on the source IP address of a packet. These types of access list are not as powerful as extended access lists, but they are less processor intensive for the router.



Let's say that server S1 holds some important documents that need to be available only to the company's magament. We could configure an access list on R1 to enable access to S1 only to users from the management network All other traffic going to S1 will be blocked. This way, we can ensure that only authorized user can access the sensitive files on S1.

Study Title _____

OSPF

- widely used and supported routing protocol
- every router has the same information about network
  ↳ routers get this information by link-state advertisement (LSAs)

LSA → contains information about the subnet, router and some other network information

LSDB ⟶ Link-state Data Base

LSA	LSA
LSA	LSA
LSA	LSA

* OSPF keeps all the information in LSDB.

Become neighbours – two routers running OSPF on the same link agree to form a neighbour relationship

Exchange database information – the neighbour routers swap their LSBD information with each other

Choose the best routes – each router chooses the best routes to add to its routing table based on the learned LSBD information

Router ID (RID) → format of an IPv4 address
↓
Manually assigned
↓
Highest 'up' status loopback interface IP address
↓
Highest 'up' status non-loopback interface IP address

Loopback – since there is no physical port, it is an open and stable interface as long as the router is running. Even if there is no LAN connected to the router, with Loopback interface, we can test the configurations we pretend to have a LAN here.

Requirements Match?
* Area ID
* Subnet
* Hello and Dead Interval
* Authentication
* Stub area flag
* Unique router ID

Study Title

Requirements

Dead timer is how long a router will wait without hearing a Hello message before it just assumes that a neighbor is dead.

Default Hello timer is every 10 sec for point to point and broadcast networks

- Area ID needs to be the same
- Connecting links between the two routers must be on the same subnet
- Hello and Dead timers must be the same.
- Authentication also must match
- Stub area flag must match as well
- Routers need unique Router IDs.

Stub are flag is used to communicate between adjacent (bitisit) nodes if one of them is in a stub area.

Once the checks have been made and if all is good then Routers can move to Init State

↓

when two routers send Hello message to each other with Router IDs, they enter the 2-way state

Study Title _____

ACL

Rule-based lists that are used by switches and routers to identity traffic bae on characteristics such as IP address or port number. Once identified, the switch or router can filter the traffic.

→ Order of the list is important ❗

10 deny tcp 192.168.10.0 00.0.255 host 192.168.20.50 eq ftp
20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.20.50 eq telnet
30 deny ip host 192.168.10.0 host 192.168.20.50

Standart ACL

→ uses number 1-99
→ expanded number 1300-1999
→ identifies traffic on source address only

ACL No — Source Address

Router (config)# access-list 1 permit 192.168.10.0 0.0.0.255

Action — Wildcard Mask

Extended ACL

→ uses number 100-199
→ expanded number 2000-2699
→ identifies traffic on source address, destination address and protocol

ACL No — Protocol — Src Wildcard — Dest Wildcard — Port

Router(config)# access list 100 deny tcp 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0 eq ftp (21)

Action — Source Address — Destination Address — Operator

gt = greater than
ls = less than
neq = not equal
eq = equal
range = range spec.

Study Title _____

Named ACL

→ allows standard and extended ACLs to be given names instead of numbers making them easier to manage

```
Router(config)# ip access-list extended NAME      (or standard; NAME = name)
Router(config-ext-nacl)# deny tcp 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0 eq ftp
Router(config-ext-nacl)# permit ip 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0
```

Lab 5.1.8 → Configure numbered Standart IPv4 ACLs

```
R2(config)# access-list 1 deny 192.168.11.0 0.0.0.255
R2(config)# access-list 1 permit any

R2(config)# int gi 0/0
R2(config-if)# ip access-group out
```

Lab 5.1.9 → Configure Named Standard IPv4 ACLs

→ Only PC1 and Su_web can access to SU_File. All other traffic to the File Su should be denied

```
R1(config)# ip access list standard File_Server_Rest
R1(config-std-acl)# permit host 192.168.20.4 (PC1)
R1(config-std-acl)# permit host 192.168.100.100 (Su_web)
R1(config-std-acl)# deny any

R1(config)# int fa 0/1 → goes to Su_File
R1(config-if)# ip access-group File_Server_Rest out
```

Lab 5.2.7 → Configure and modify Standard IPv4 ACLs

Part 1 → create a standart numbered ACL that allows traffic from all host on the 192.168.10.0/24 network and all hosts on the 192.168.20.0/24 network to access all hosts on the 192.168.30.0/24 network.

R3 → nereye uygulanacaksa oranın router'ına işlem yapılır

```
R3(config)# access list 1 remark Allow R1 LANs Access
R3(config)# access list 1 permit 192.168.10.0 0.0.0.255
R3(config)# access list 1 permit 192.168.20.0 0.0.0.255
R3(config)# access list 1 deny any

R3(config)# int g0/0/0 →nereye uygulanacaksa
R3(config-if)# ip access-group 1 out
```

Part 2 → create a named standard ACL that allows traffic from all hosts on the 192.168.40.0/24 network access to the 192.168.10.0/24. Also, only allow host PC-C access to the 192.168.10.0/24 network. Name it BRANCH-OFFICE-POLICY

```
R1(config)#ip access-list BRANCH-OFFICE-POLICY
R1(config-std-nacl)# permit host 192.168.30.3 (PC-C)
R1(config-std-nacl)#permit 192.168.40.0 0.0.0.255
R1(config-std-nacl)#end

R1(config)# int g0/0/0 → 192.168.10.0'a giden int (PC-A)
R1(config-if)# ip access-group BRANCH-OFFICE-POLICY out
```

Study Title _____

Lab 5.4.12 → Configure Extended IPv4 ACLs → Scenario 1
→ Two employees need access to services provided by the server. PC1 only needs FTP access while PC2 only needs web access. Both computers need to be able the server, but not each other.

/27 → 255.255.255.224 → 0.0.0.31
/28 → 255.255.255.240 → 0.0.0.15

Part 1 → Configure an ACL to permit FTP and ICMP from PC1 LAN

net to PC1 — File Transfer Protocol

R1(config)# access list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp

Transmission Control Protocol — server

net to PC1

R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62

internet control message protocol — server

R1# sh access-lists

R1(config)# int gi0/0
R1(config-if)# ip access-group 100 in

Part 2 → Configure an ACL to permit HTTP access and ICMP from PC2 LAN

server

R1(config)# ip access-list extended HTTP_Only
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www

net to PC2

R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62

R1(config)# int gi0/1
R1(config-if)# ip access-group HTTP_Only in

Lab 5.4.13 → Configure Extended IPv4 ACLs → Scenario 2
→ Specific devices on the LAN are allowed to various services on servers located on the internet.

when you write "host" → no need wildcard

Part 1 → Block HTTPS and HTTP access from PC1 to Server1 and Server2. The servers are inside the cloud and only know their IP addresses.
→ Block FTP access from PC2 to Server1 and Server 2
→ Block ICMP access from PC3 to Server 1 and Server 2

```
R1(config)# ip access-list extended Limited Access
R1(config-ext-nacl)# deny tcp host 172.31.1.101 host 64.101.255.254 eq 80
                                  PC1            Server 1          HTTP
R1(config-ext-nacl)# deny tcp host 172.31.1.101 host 64.101.255.254 eq 443
                                  PC1            Server 1          HTTPS
R1(config-ext-nacl)# deny tcp host 172.31.1.101 host 64.103.255.254 eq 80
                                  PC1            Server 2          HTTP
R1(config-ext-nacl)# deny tcp host 172.31.1.101 host 64.103.255.254 eq 443
                                  PC1            Server 2          HTTPS
R1(config-ext-nacl)# deny tcp host 172.31.1.102 host 64.101.255.254 eq 21
                                  PC2            Server 1          FTP
R1(config-ext-nacl)# deny tcp host 172.31.1.102 host 64.103.255.254 eq 21
                                  PC2            Server 2          FTP
R1(config-ext-nacl)# deny icmp host 172.31.1.103 host 64.101.255.254
                                   PC3             Server 1
R1(config-ext-nacl)# deny icmp host 172.31.1.103 host 64.103.255.254
                                   PC3             Server 2
R1(config-ext-nacl)# permit any any

R1(config)# int gi0/0 ⇒ PC tarafı
R1(config-if)# ip access-group LimitedAccess in
```

NAT

Lab 6.4.5 → Configure Static NAT

outside

R1(config)# ip nat inside source static 172.16.16.1 64.100.50.1

inside

R1(config)# int gi 0/0 → server'a giden net
R1(config-if)# ip nat inside

R1(config)# int se 0/1/0 → internet'e giden taraf
R1(config)# ip nat outside

Lab 6.5.6 → Configure Dynamic NAT

Part 1 → Configure Dynamic NAT

- configure one statement for ACL 1 to permit any address belonging to the 172.16.0.0/16 network

R2(config)# access-list permit 172.16.0.0 0.0.255.255

- Configure R2 with a NAT pool that uses two addresses in the 209.165.200.228/30 address space.

R1(config)# ip nat pool myPool 209.165.200.229 209.165.200.230 netmask 255.255.255.252

- Associate ACL 1 with the NAT pool

R1(config)# ip nat inside source list 1 pool myPool

- Configure R2 interfaces

R1(config)# int se0/1/0
R1(config-if)# ip nat outside

R1(config)# int se0/1/1
R1(config-if)# ip nat outside

Study Title _____

Lab 6.6.7 → Configure PAT

Part 1 → Configure Dynamic NAT with Overload

R1(config)# access-list 1 permit 172.16.0.0 0.0.255.255 → permit any address belonging

R1(config)# ip nat pool NAME 209.165.200.233 209.165.200.234 netmask 255.255.255.252

two usable address in the 209.165.200.232/30

R1(config)# ip nat inside source list 1 pool NAME overload → associate ACL 1 with the NAT pool

allow adresses to be reused

```
R1(config)# int se0/1/0
R1(config-if)# ip nat outside
R1(config-if)# int gi0/0/0
R1(config-if)# ip nat inside
R1(config-if)# int gi0/0/1
R1(config-if)# ip nat inside
```

Part 2 → Configure PAT using an Interface

R1(config)# access-list 2 permit 172.17.0.0 0.0.255.255

0.0 çünkü her neti yazmaya gerek yok

ex: 172.17.10.0 172.17.20.0

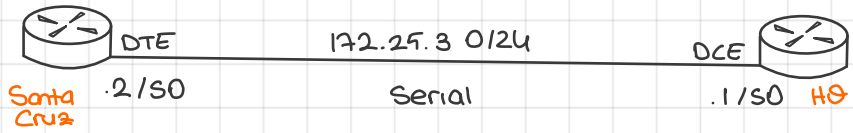
R1(config)# ip nat inside source list 2 interface se0/1/1 overload → use the interface connected to the internet and provide translations for all internal devices

```
R1(config)# int gi0/0/0
R1(config-if)# ip nat inside
R1(config)# int gi 0/0/1
R1(config-if)# ip nat inside
R1(config)# int se0/1/1
R1(config-if)# ip nat outside
```

Study Title _____

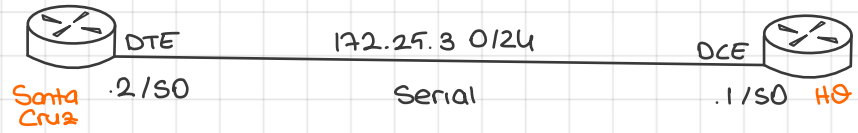
PPP AUTHENTICATION

PAP ⇒ no encryption




```
SantaCruz(config)# username HQ password HQpass
SantaCruz(config)# int serial0
SantaCruz(config-if)# ip address 172.25.3.2 255.255.255.0
SantaCruz(config-if)# encapsulation ppp
SantaCruz(config-if)# ppp authentication pap
SantaCruz(config-if)# ppp pap sent-username SantaCruz password SantaCruzPass

HQ(config)# username SantaCruz password SantaCruzPass
HQ(config)# int serial0
HQ(config-if)# ip address 172.25.3.1 255.255.255.252
HQ(config-if)# encapsulation ppp
HQ(config-if)# ppp authentication pap
HQ(config-if)# ppp pap sent-username HQ password HQPass
```

CHAP ⇒ same shared password

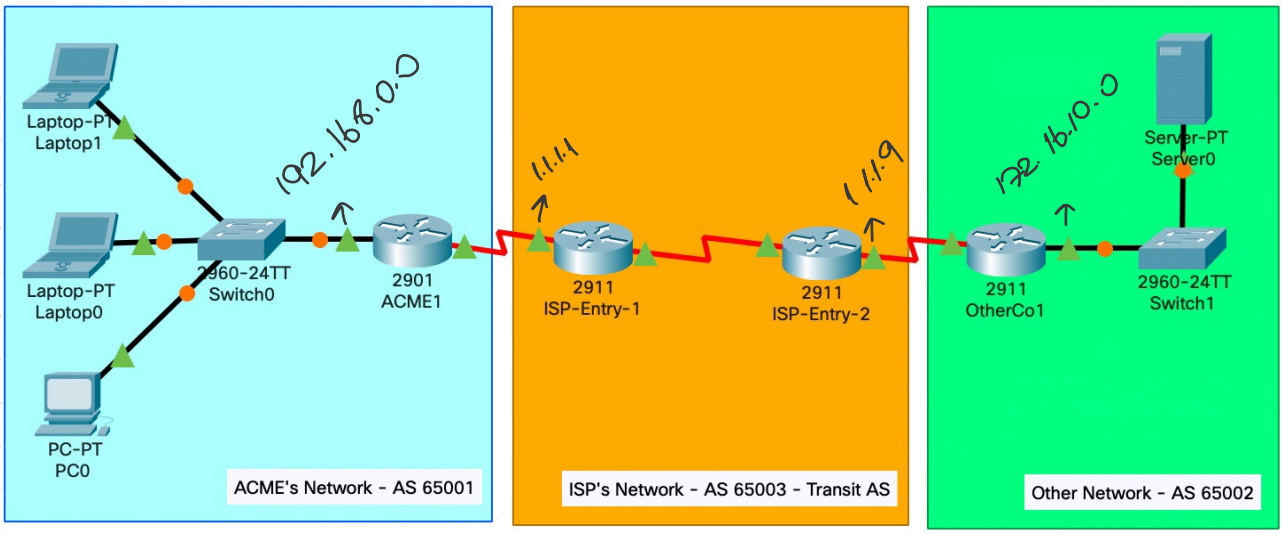


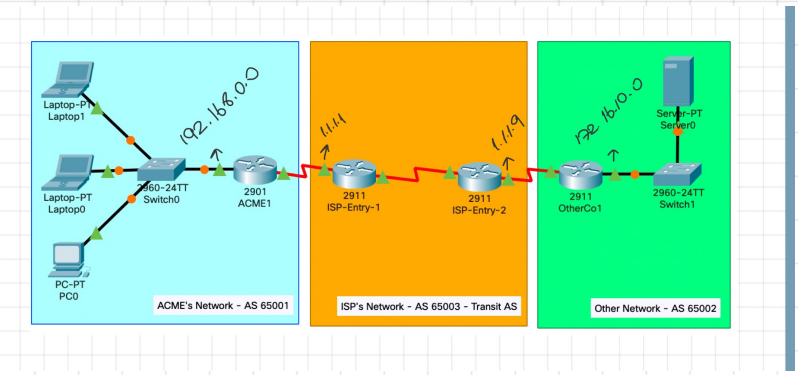

```
SantaCruz(config)# username HQ paseword boardwalk
SantaCruz(config)# int serial0
SantaCruz(config-if)# ip address 172.25.3.2 255.255.255.252
SantaCruz(config-if)# encapsulation ppp
SantaCruz(config-if)# ppp authentication chap

HQ(config)# username SantaCruz password boardwalk
HQ(config)# int serial0
HQ(config-if)# ip address 172.25.3.1 255.255.255.252
HQ(config-if)# encapsulation ppp
HQ(config-if)# ppp authentication chap
```

BGP

Lab Configure and verify eBGP

```
ACME1(config)# router bgp 65001
ACME1(config-router)# neighbor 1.1.1.1 remote-as 65003
ACME1(config-router)# network 192.168.0.0 mask 255.255.255.252

OtherCo1(config)# router bgp 65002
OtherCo1(config-router)# neighbor 1.1.1.9 remote-as 65003
OtherCo1(config-router)# network 172.16.10.0 mask 255.255.255.252
```





GRE

```
RA(config)# int tunnel 0        IP of tunnel on RA
RA(config-if)# ip address 10.10.10.1  255.255.255 252
RA(config-if)# tunnel source s0/0/0
RA(config-if)# tunnel destination 209.165.122.2 → (IP of RB on se0/0/0
RA(config-if)# tunnel mode gre ip
RA(config-if)# no sh

RA(Config)# ip route 192.168.2.0  255.255.255.0  10.10.10.2

                    RB'nin PCB'ye giden IP

RB(config)# int tunnel 0        ip of tunnel on RB
RB(config-if)# ip address 10.10.10.2  255.255.255 252
RB(config-if)# tunnel source s0/0/0
RB(config-if)# tunnel destination 64.103.211.2 → (IP of RA on se0/0/0 )
RB(config-if)# tunnel mode gre ip
RB(config-if)# no sh

RA(Config)# ip route 192.168.1.0  255.255.255.0  10.10.10.1

                    RA'nin PCA'ya giden tarafı
```