

Established under sub section (3) of Section 1 of Garden City University Act, 2013 (Karnataka Act No. 47 of 2013)

**Dr. A P J Abdul Kalam School of Engineering**

**B. Tech in Computer Science and Engineering**

**(AI & ML, CS and DS)**

**LAB MANUAL**

**COURSE CODE: 10ABTEC22213 COURSE INSTRUCTOR: T. SRINIVAS PRASAD**

**COURSE NAME: COMPUTER NETWORK SEMESTER: II**

**AND TROUBLE SHOOTING (LAB)**

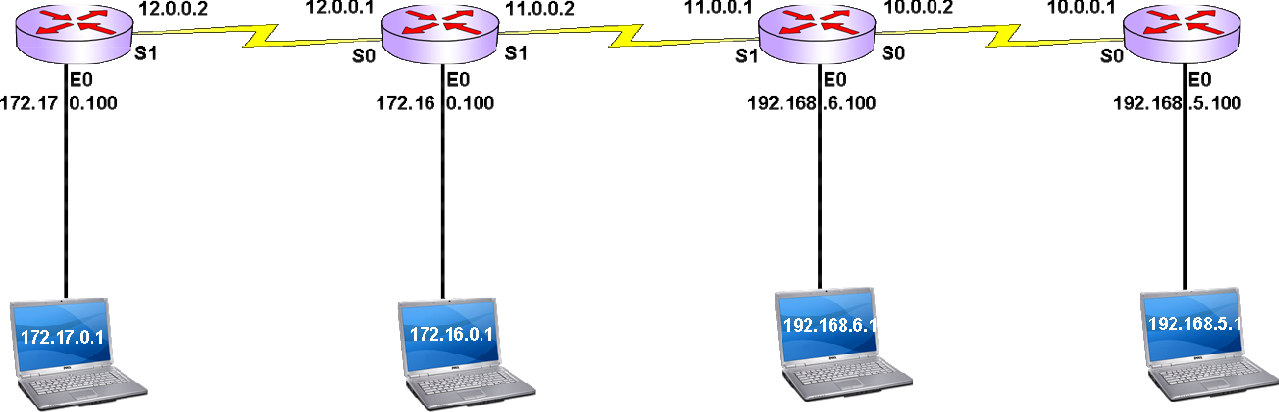
|  |  |
| --- | --- |
| 1 | **Configuring Static Routing** |
| 2 | **Configuring Routing Information Protocol (RIP)** |
| 3 | **Configuring Interior Gateway Routing Protocol (IGRP)** |
| 4 | **Configuring Open Shortest Path First (OSPF)** |
| 5 | **Configuring Network Address Translation (NAT)** |
| 6 | **Configuring Access Control List (ACL)** |
| 7 | **Configuring Secure Shell (SSH)** |
| 8 | **Configuring Privilege Levels** |
| 9 | **Configuring Zone Based Firewall (ZBF)** |
| 10 | **Configuring Layer 2 Security** |
| 11 | **Virtual Private Network (VPN)** |

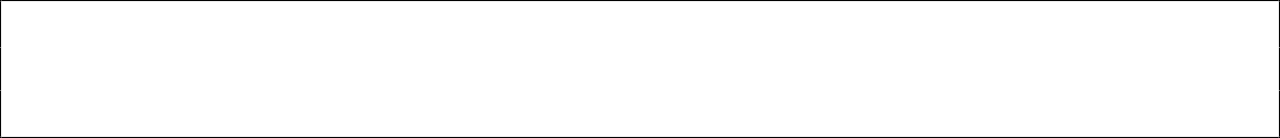
**Computer Network and Troubleshooting**

**List of Programs**

**LAB 1: Configuring Static Routing**



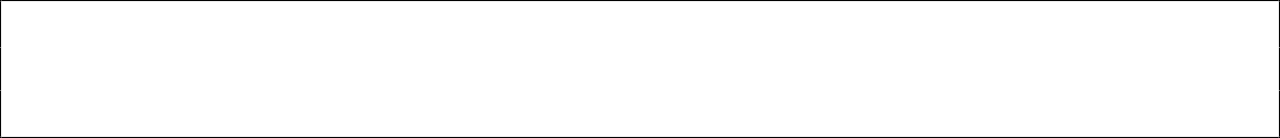




Syntax:

Router(Config)# ip route <network> <subnetmask> <exit int>

##### (Or)



Syntax:

Router(Config)# ip route <network> <subnetmask> <nexthop>

Exit Interface:

Interface on the home router which forwards the data to the next router

Next hop IP address:

Interface IP Address of next immediate router towards the destination

### **Static Routing configuration**

##### Bang:

Chen:

##### Hyd:

Bang> enable

Bang# show ip route Bang# config t

Bang(config)# no ip routing Bang(config)# ip routing

Bang(config)# ip route 172.16.0.0 255.255.0.0 s 1

Bang(config)# ip route 192.168.6.0 255.255.255.0 s 1

Bang(config)# ip route 192.168.5.0 255.255.255.0 s 1 Bang(config)# exit

Bang # show ip route

Chen> enable

Chen# show ip route Chen# config t

Chen(config)# no ip routing Chen(config)# ip routing

Chen(config)# ip route 172.17.0.0 255.255.0.0 s 0

Chen(config)# ip route 192.168.6.0 255.255.255.0 s 1

Chen(config)# ip route 192.168.5.0 255.255.255.0 s 1 Chen(config)# exit

Chen # show ip route

Hyd> enable

Hyd# show ip route Hyd# config t

Hyd(config)# no ip routing Hyd(config)# ip routing

Hyd(config)# ip route 172.17.0.0 255.255.0.0 s 1

Hyd(config)# ip route 172.16.0.0 255.255.0.0 s 1

Hyd(config)# ip route 192.168.5.0 255.255.255.0 s 0 Hyd(config)# exit

Hyd # show ip route

##### Del:

Del> enable

Del# show ip route Del# config t

Del(config)# no ip routing Del(config)# ip routing

Del(config)# ip route 172.17.0.0 255.255.0.0 s 0

Del(config)# ip route 172.16.0.0 255.255.0.0 s 0

Del(config)# ip route 192.168.6.0 255.255.255.0 s 0 Del(config)# exit

Del # show ip route

### **Static default Routing configuration**

* It is a form of static routing
* Used when destination information is not available
* Used as the last option
* Configured at “End points” /stub network (with one exit interface)
* In default routing destination network address is 0.0.0.0
* Used in Internet configuration

Syntax:

Router(Config)# ip route

0.0.0.0

0.0.0.0

<exit int>

* + - 1. Network with 0.0.0.0 subnet mask value represents all ip addresses from 0.0.0.0 to 255.255.255.255

Bang, Del routers are the end points/stub routers in the LAB network

##### Bang:

Bang> enable

Bang# show ip route Bang# config t

Bang(config)# no ip routing Bang(config)# ip routing

Bang(config)# ip route 0.0.0.0 0.0.0.0 s 1 Bang(config)# exit

Bang # show ip route

##### Del:

Del> enable

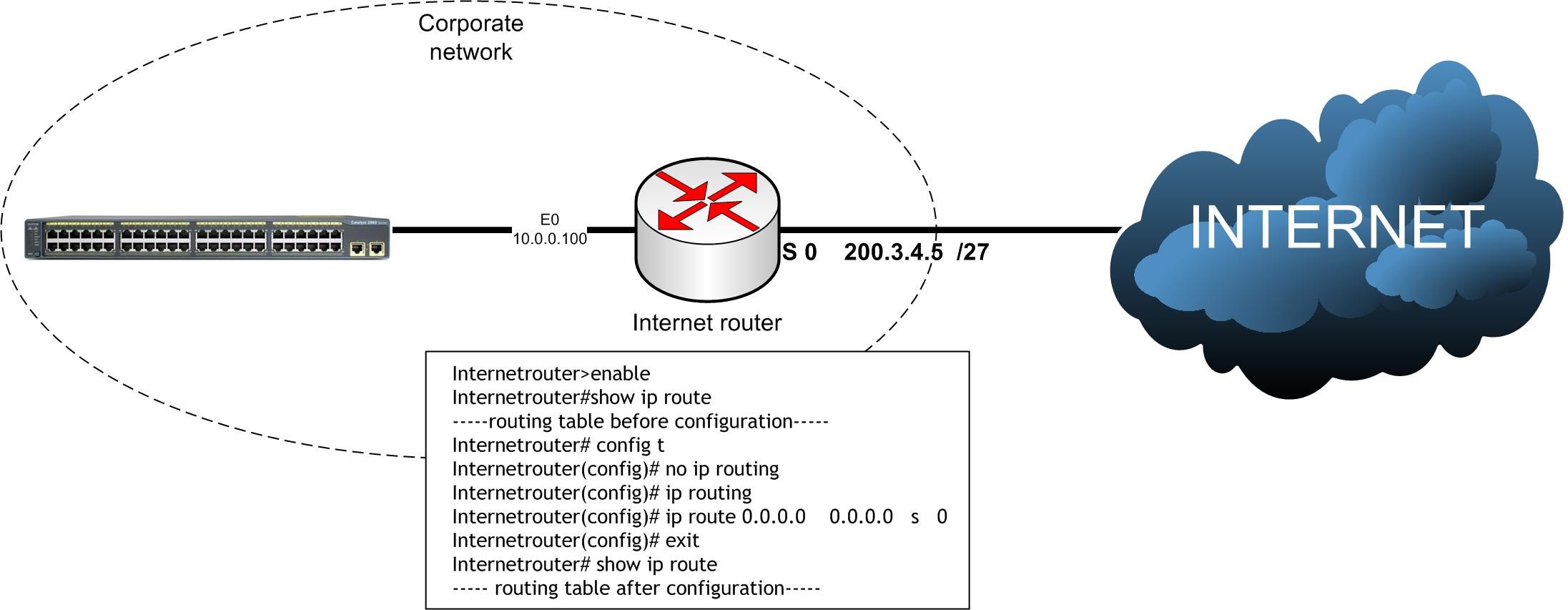
Del# show ip route Del# config t

Del(config)# no ip routing Del(config)# ip routing

Del(config)# ip route 0.0.0.0 0.0.0.0 s 0 Del(config)# exit

Del # show ip route

### **Internet Router Configuration (Static default Routing)**



# **Lab 2 : Configuring RIPv1**

### **Routing Information Protocol (v1)**

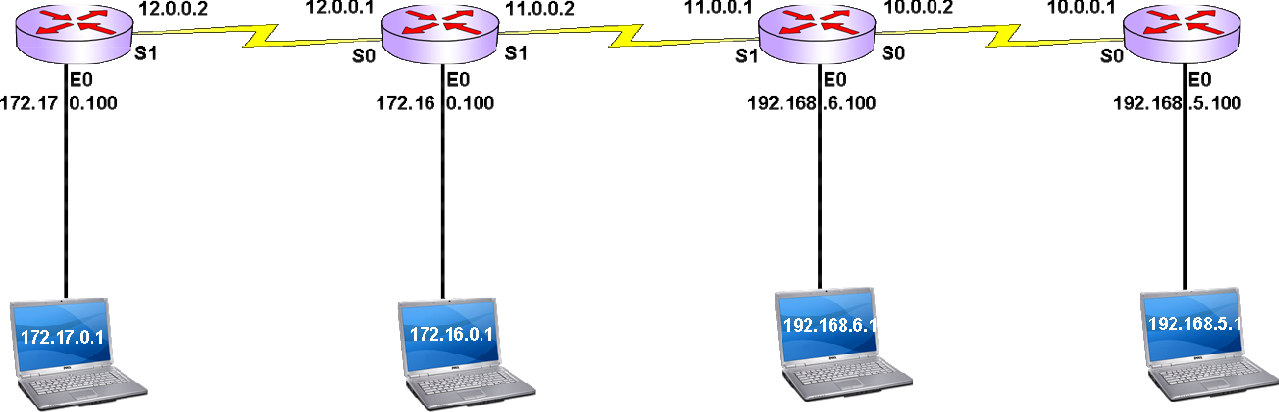
* AD=120 (Administrative distance)
* Metric=hop count (15=max, 16=invalid)
* Algorithm= bellman ford
* Update timer =30 sec
* Invalid timer =180 sec
* Hold down timer =180 sec
* Flush timer =240 sec
* Load balancing =6 equal paths
* Classful routing (subnetting “not “supported)
* Open Standard

Syntax:

Router(config)# router rip

Router(config-router)# network <network Address>





### **RIPv1 configuration**

##### Bang:

Chen:

##### Hyd:

Del:

Bang> enable

Bang# config t

Bang(config)# no ip routing

Bang(config)# ip routing

Bang(config)# router rip

Bang(config-router)# version 1

Bang(config-router)# network 172.17.0.0

Bang(config-router)# network 12.0.0.0

Bang(config-router)# end

Bang # show ip route

Chen> enable

Chen# config t

Chen(config)# no ip routing

Chen(config)# ip routing

Chen(config)# router rip

Chen(config-router)# version 1

Chen(config-router)# network 172.16.0.0

Chen(config-router)# network 12.0.0.0

Chen(config-router)# network 11.0.0.0

Chen(config-router)# end

Chen # show ip route

Hyd> enable Hyd# config t

Hyd(config)# no ip routing Hyd(config)# ip routing Hyd(config)# router rip Hyd(config-router)# version 1

Hyd(config-router)# network 192.168.6.0

Hyd(config-router)# network 11.0.0.0

Hyd(config-router)# network 10.0.0.0 Hyd(config-router)# end

Hyd # show ip route

Del> enable Del# config t

Del(config)# no ip routing Del(config)# ip routing Del(config)# router rip Del(config-router)# version 1

Del(config-router)# network 192.168.5.0

Del(config-router)# network 10.0.0.0 Del(config-router)# end

Del # show ip route

# **RIPv2**

### **Routing Information Protocol (v2)**

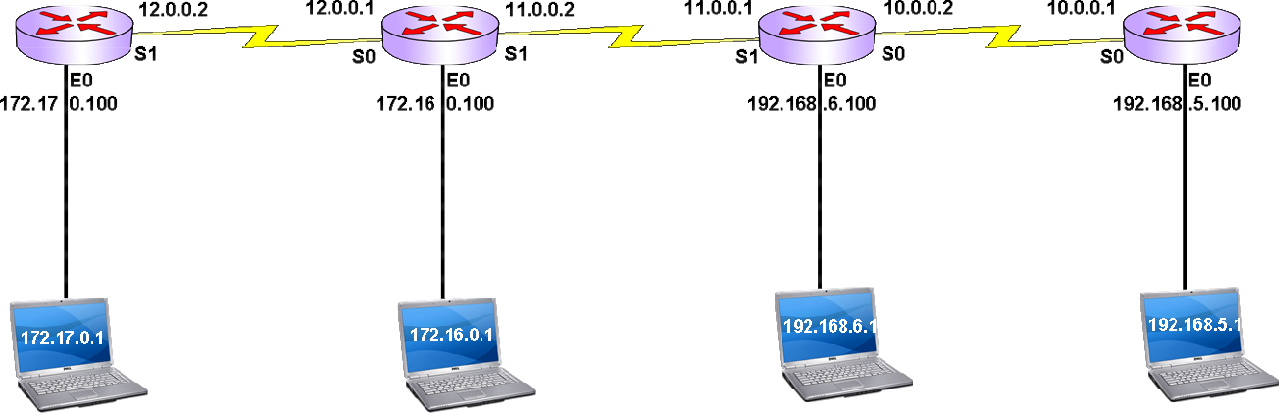
* AD=120 (Administrative distance)
* Metric=hop count (15=max, 16=invalid)
* Algorithm= bellman ford
* Triggered updates
* Multicast updates on 224.0.0.9
* Load balancing =6 equal paths
* Classless routing (subnetting supported)
* Open Standard

Syntax:

Router(config)# router rip Router(config-router)# version 2

Router(config-router)# network <network Address>





### **RIPv2 configuration**

##### Bang:

Bang> enable

Bang# config t

Bang(config)# no ip routing

Bang(config)# ip routing

Bang(config)# router rip

Bang(config-router)# version 2

Bang(config-router)# network 172.17.0.0

Bang(config-router)# network 12.0.0.0

Bang(config-router)# end

Bang # show ip route

Chen:

Chen> enable

Chen# config t

Chen(config)# no ip routing

Chen(config)# ip routing

Chen(config)# router rip

Chen(config-router)# version 2

Chen(config-router)# network 172.16.0.0

Chen(config-router)# network 12.0.0.0

Chen(config-router)# network 11.0.0.0

Chen(config-router)# end

Chen # show ip route

##### Hyd:

Hyd> enable

Hyd# config t

Hyd(config)# no ip routing

Hyd(config)# ip routing

Hyd(config)# router rip

Hyd(config-router)# version 2

Hyd(config-router)# network 192.168.6.0

Hyd(config-router)# network 11.0.0.0

Hyd(config-router)# network 10.0.0.0

Hyd(config-router)# end

Hyd # show ip route

Del:

Del> enable

Del# config t

Del(config)# no ip routing

Del(config)# ip routing

Del(config)# router rip

Del(config-router)# version 2

Del(config-router)# network 192.168.5.0

Del(config-router)# network 10.0.0.0

Del(config-router)# end

Del # show ip route

# **Lab 3 :Configuring IGRP**

### **Interior Gateway Routing Protocol**

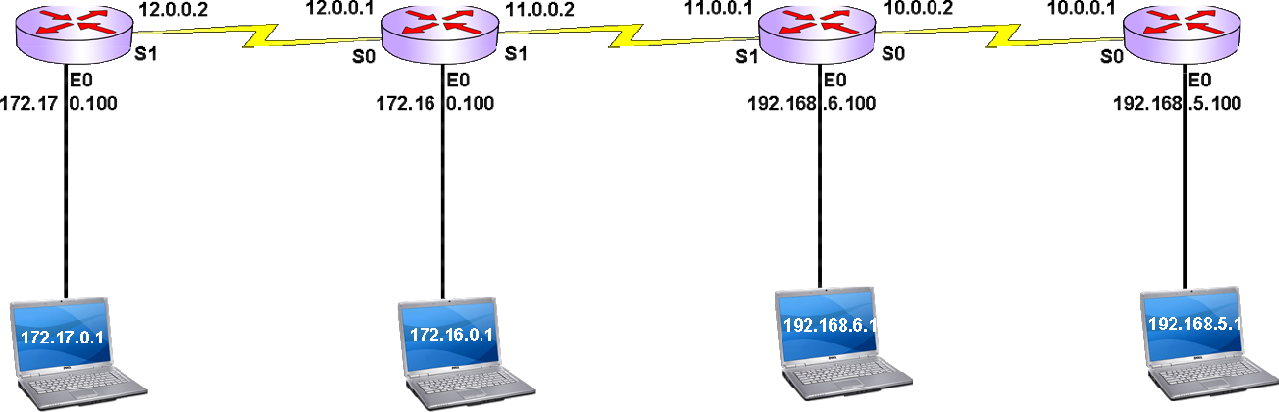
* AD=100
* Metric = 24 Bit Composite (Bandwidth+Delay+Load+Reliablity+MTU)
* Algorithm = Bellman Ford
* Update timer = 90 Sec
* Invalid timer = 270 Sec
* Hold On timer = 280 Sec
* Flush timer = 630 Sec
* Load balancing = 4-6 equal /unequal paths
* Classful routing (subnetting “not” supported)
* Cisco proprietary

Syntax:

Router(config)# router igrp <AS No>

Router(config-router)# network <network Address>





### **IGRP configuration**

##### Configure all routers in the same Autonomous system

IGRP communicates within AS only

##### Bang:

Bang> enable

Bang# config t

Bang(config)# no ip routing

Bang(config)# ip routing

Bang(config)# router igrp 87

Bang(config-router)# network 172.17.0.0

Bang(config-router)# network 12.0.0.0

Bang(config-router)# end

Bang # show ip route

Chen:

Chen> enable

Chen# config t

Chen(config)# no ip routing

Chen(config)# ip routing

Chen(config)# router igrp 87

Chen(config-router)# network 172.16.0.0

Chen(config-router)# network 12.0.0.0

Chen(config-router)# network 11.0.0.0

Chen(config-router)# end

Chen # show ip route

##### Hyd:

Hyd> enable

Hyd# config t

Hyd(config)# no ip routing

Hyd(config)# ip routing

Hyd(config)# router igrp 87

Hyd(config-router)# network 192.168.6.0

Hyd(config-router)# network 11.0.0.0

Hyd(config-router)# network 10.0.0.0

Hyd(config-router)# end

Hyd # show ip route

Del:

Del> enable

Del# config t

Del(config)# no ip routing Del(config)# ip routing

Del(config)# router igrp 87

Del(config-router)# network 192.168.5.0

Del(config-router)# network 10.0.0.0

Del(config-router)# end

Del # show ip route

##### 

# **LAB 4: Configuring OSPF**

### **Open Shortest Path First**

* AD=110
* Metric = cost (108/bandwidth in bps)
* Algorithm = DIJKSTRA or SPF
* Hello timer = 10 sec
* Dead timer = 40 sec
* Flush timer = 30 min
* Multicast updates on 224.0.0.5, 224.0.0.6
* It is Classless ( Subnetting supported )
* Open Standard

Syntax:

Router(config)# router ospf <Process id No>

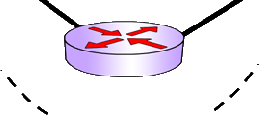
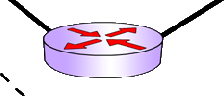
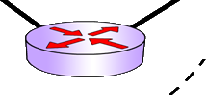
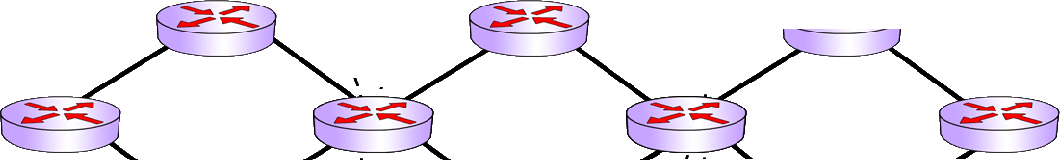
Router(config-router)# network <network> <WCM> area <area id>

##### What is WCM?

* + - * + Wild Card Mask
        + Inverse of Subnet Mask Value
        + Class A WCM : 0.255.255.255
        + Class B WCM : 0.0.255.255
        + Class C WCM : 0.0.0.255

##### What is Area?

* + - * + Ospf maintains Link state information of every router to run SPF algorithm
        + Router consumes more resources if more routers present in the network
        + Areas are used to limit the link state database handled by router
        + Area is a logical boundary for OSPF routers
        + OSPF routers handle the link state information of all routers belong to same area
        + Area Border Routers(ABR) route the data between different areas



##### What is Process id?

* + - * + OSPF can be configured as multiple instances on the same router
        + Process id is used to identify the instance of OSPF
        + It need not be the same on all routers
        + Process id is 16 bit value
        + Range : 1- 65535

##### OSPF areas:

* + - * + OSPF areas are basically two types

Backbone area

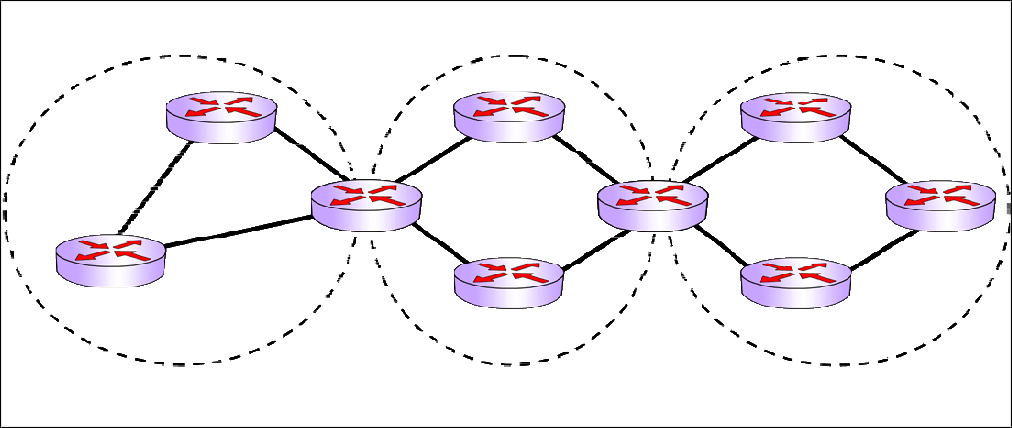
Area 0 is called as backbone area

Transit area between different areas

Non Backbone area

Areas other than Backbone area

All non backbone areas must be directly connected to area 0



##### OSPF Router Types:

Backbone routers

Internal routers

ABR

ASBR

Backbone routers

Routers in Back bone area (area 0).

Internal routers

Routers belong to same area (backbone or non back bone). Back bone routers are internal routers

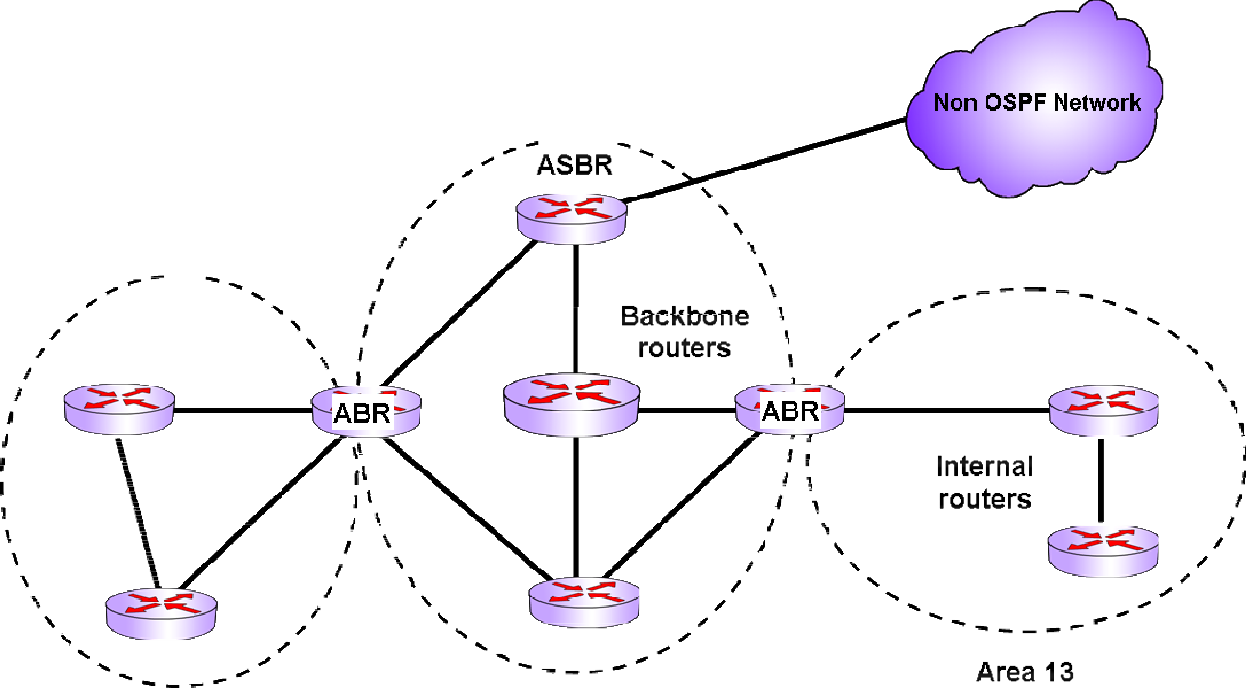
ABR – Area Border Router

Router belongs to multiple areas.

For ABR, at least one interface must be in Back bone area

ASBR – Autonomous system boundary router

OSPF router that is connected to non OSPF network ASBR is generally placed in area 0



##### OSPF area Types:

Backbone area

Stub Area

Totally stub area

NSSA (Not so stubby area)

##### LSA Types:

* LSA – Link state Advertisement
* It contains Link state information and networks available on that link

|  |  |
| --- | --- |
| LSA Types | Name |
| 1 | Router LSA |
| 2 | Network LSA |
| 3 | Summary LSA |
| 4 | ASBR summary |
| 5 | AS external LSA |
| 6 | Multicast OSPF |
| 7 | NSSA LSA |

### **OSPF configuration**

##### Configure all routers in single area (Area 0)

Process id can be different from router to router

Bang:

Bang> enable

Bang# config t

Bang(config)# no ip routing

Bang(config)# ip routing

Bang(config)# router ospf 14

Bang(config-router)# network 172.17.0.0 0.0.255.255 area 0

Bang(config-router)# network 12.0.0.0 0.255.255.255 area 0

Bang(config-router)# end

Bang # show ip route

Chen:

Chen> enable

Chen# config t

Chen(config)# no ip routing

Chen(config)# ip routing

Chen(config)# router ospf 1456

Chen(config-router)# network 172.16.0.0 0.0.255.255 area 0

Chen(config-router)# network 12.0.0.0 0.255.255.255 area 0

Chen(config-router)# network 11.0.0.0 0.255.255.255 area 0

Chen(config-router)# end

Chen # show ip route

Hyd:

Hyd> enable

Hyd# config t

Hyd(config)# no ip routing

Hyd(config)# ip routing

Hyd(config)# router ospf 258

Hyd(config-router)# network 192.168.6.0 0.0.0.255 area 0

Hyd(config-router)# network 11.0.0.0 0.255.255.255 area 0

Hyd(config-router)# network 10.0.0.0 0.255.255.255 area 0

Hyd(config-router)# end

Hyd # show ip route

Del:

Del> enable

Del# config t

Del(config)# no ip routing

Del(config)# ip routing

Del(config)# router ospf 25696

Del(config-router)# network 192.168.5.0 0.0.0.255 area 0

Del(config-router)# network 10.0.0.0 0.255.255.255 area 0

Del(config-router)# end

Del # show ip route

## **LAB 5 : Configuring Network Address Translation**

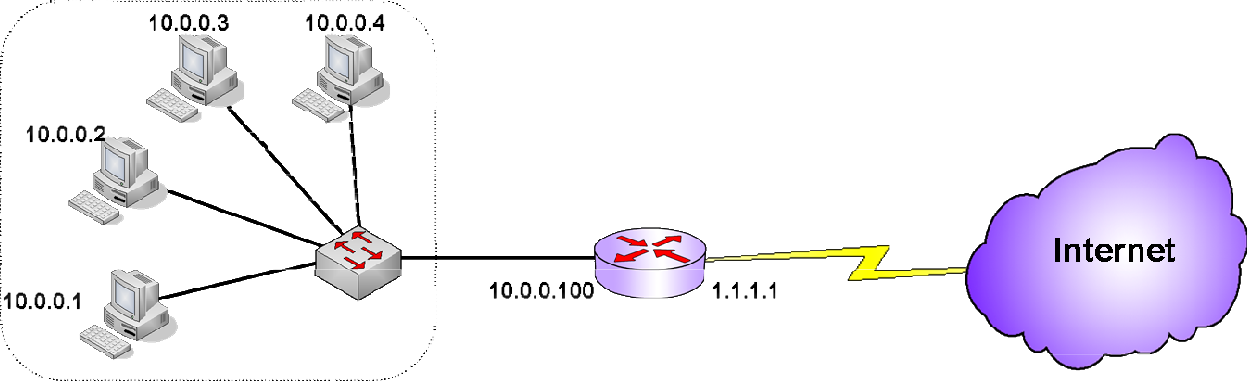
##### **What is NAT?**

* + - Network Address Translation
    - All the Local Area Networks use private IP addressing scheme
    - Private IP addresses are not routable in public network
    - To access public Network public IP address is required
    - Systems within the LAN communicate with private IP addresses

These private IP addresses need to be translated into public IP addresses while accessing public network (internet)

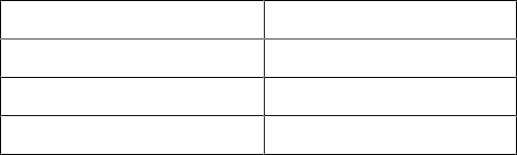
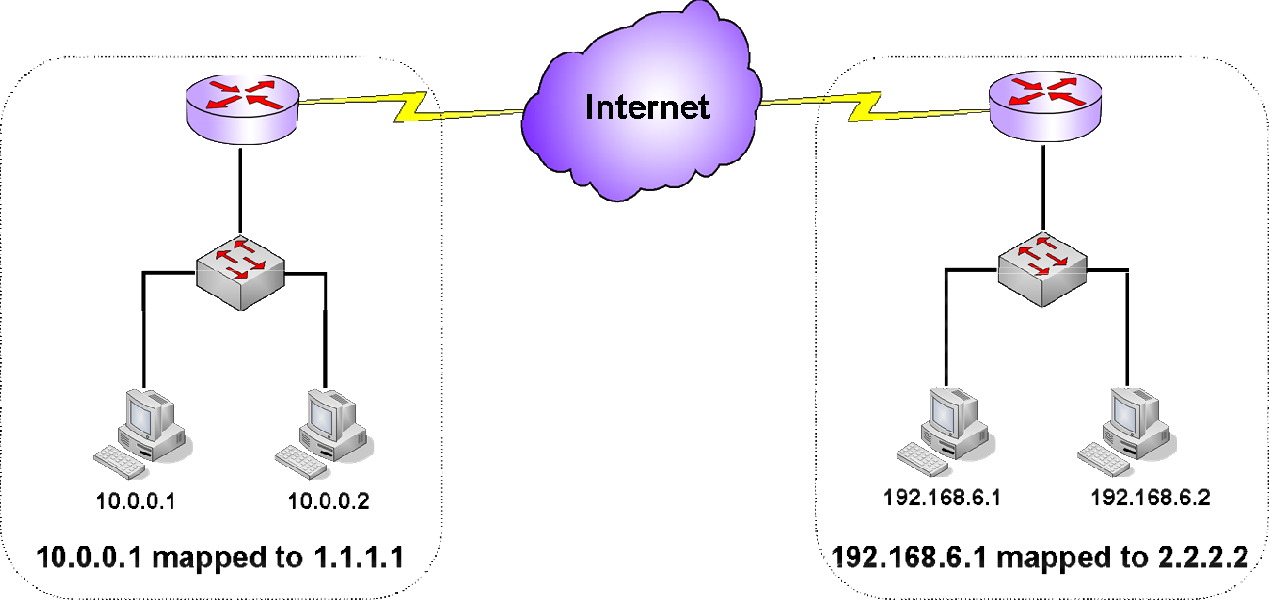
* + - When reply comes back, Public IP addresses are translated back to private IP addresses before forwarding data to system
    - Private to Public and Public to Private IP translation is called NAT
    - Generally NAT operations are taken care by router

##### **NAT Operations**:



**NAT terms**:

|  |  |
| --- | --- |
| Inside local | An inside device with an assigned private IP address |
| Inside global | An inside device with a mapped public IP address |
| Outside local | An outside device with an assigned private IP address |
| Outside global | An outside device with a mapped public IP address |



Inside local Inside global Outside local Outside global

10.0.0.1

1.1.1.1

192.168.6.1

2.2.2.2

##### **Types of NAT**:

* + - NAT is basically two types
      * Static NAT
      * Dynamic NAT

##### **Static NAT:**

* + - In static NAT one private IP address is mapped to one Public IP
    - Also called as 1 to 1 NAT
    - It is not possible to map every private IP to a public IP
    - Generally static NAT is used for public servers



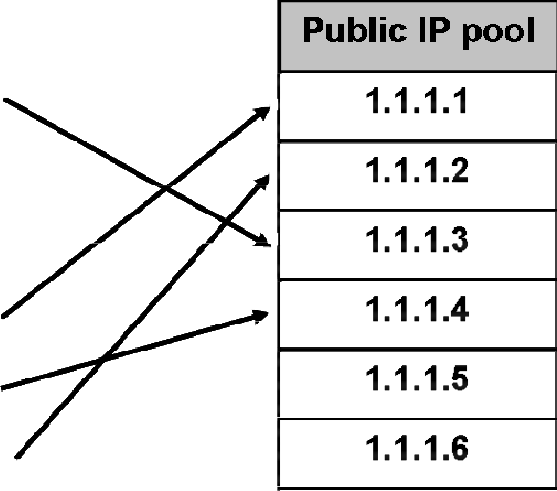




##### **Dynamic NAT**:

* + - In dynamic NAT a group of private ip addresses are mapped to a pool of public IP addresses
    - NAT happens dynamically on First come First serve basis
    - Access-list is created to specify a group of private ip addresses
    - A pool is created with public IP addresses
    - Access-list is mapped with the pool





**Static NAT configuration**

##### Syntax:

IP address Mapping:

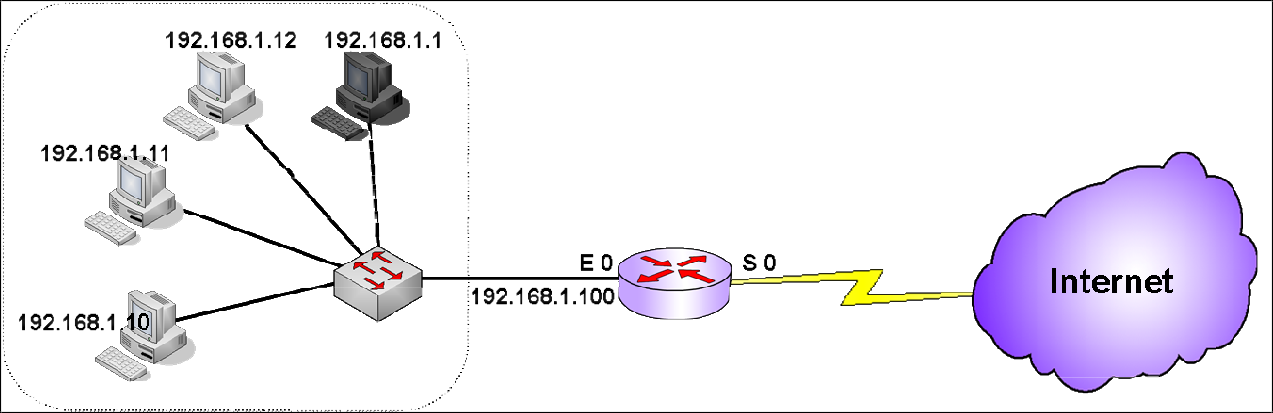
Router(config)# ip nat inside source static <inside local> <inside global>

Apply NAT on the interface:

Router(config)# interface <s0/e0/s1> Router(config-if)# ip nat inside | outside

Example:

NAT 192.168.1.1 to 200.200.200.1



##### Configuration:

IP address Mapping:

Router(config)# ip nat inside source static 192.168.1.1 200.200.200.1

Apply NAT on the interface:

Router(config)# interface e 0 Router(config-if)# ip nat inside

Router(config)# interface s 0 Router(config-if)# ip nat outside

Checking NAT results:

Router# show ip nat translations

Router# show ip nat statistics

#### **Dynamic NAT configuration**



Create ACL:

Router(config)# access-list <1-99> permit

<sourceip>

<match>

Create NAT Pool:

Router(config)# ip nat pool <poolname> <starting ip> <ending ip> netmask <mask>

Map ACL to NAT Pool:

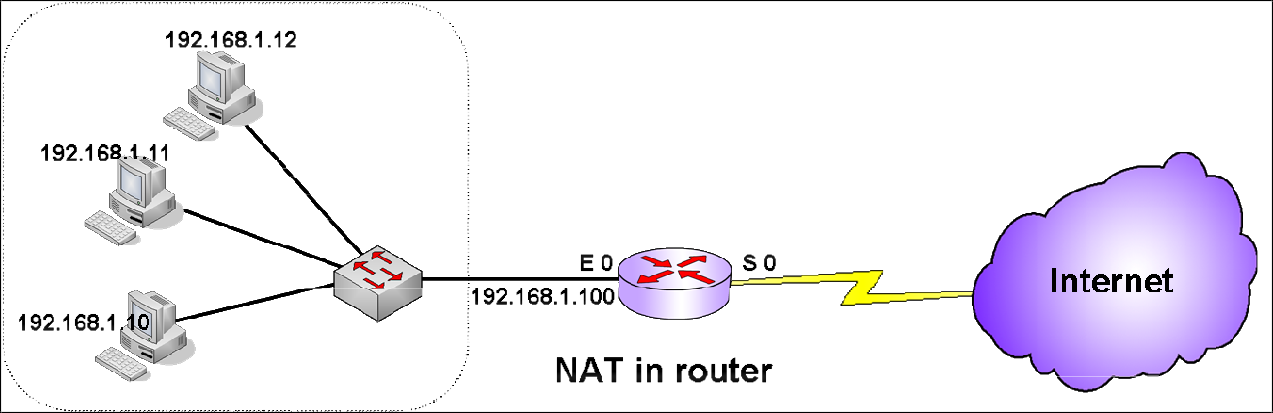
Router(config)# ip nat inside source list <1-99> pool <poolname>

Apply NAT on the interface:

Router(config)# interface <s0/e0/s1> Router(config-if)# ip nat inside | outside

##### Example:

NAT 192.168.1.10, 192.168.1.11, 192.168.1.12 to public IP pool dynamically



##### Configuration:

Create ACL:

Router(config)# access-list 73 permit 192.168.1.10 0.0.0.0

Router(config)# access-list 73 permit 192.168.1.11 0.0.0.0

Router(config)# access-list 73 permit 192.168.1.12 0.0.0.0

Create NAT Pool:

Router(config)# ip nat pool naga 200.200.200.1 200.200.200.6 netmask 255.255.255.240

Map ACL to NAT Pool:

Router(config)# ip nat inside source list 73 pool naga

Apply NAT on the interface: Router(config)# interface e 0 Router(config-if)# ip nat inside

Router(config)# interface s 0 Router(config-if)# ip nat outside

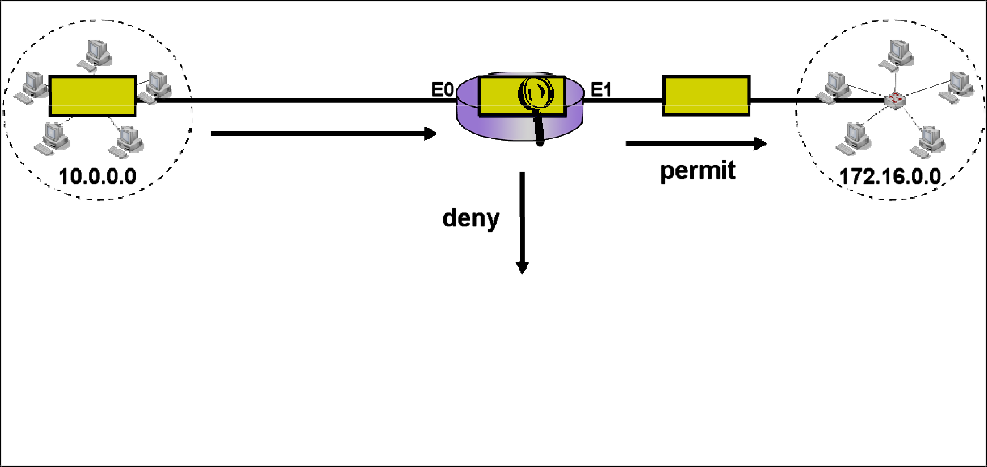
##### **LAB 6: ConfigringAccess Control List (ACL)**

##### What is ACL ?

* + - * Access Control List
      * Security implementation feature
      * It is used to filter the network traffic that crosses routers
      * ACL is the list of statements that allows or denies the predefined traffic
      * With ACL, router works as packet filtering firewall
      * Router takes filtering decisions based on L3 Header and L4 Header

L3 Header contains Source IP, destination IP, Protocol Number L4 Header contains Source Port, destination Port numbers

##### ACL Basic idea:



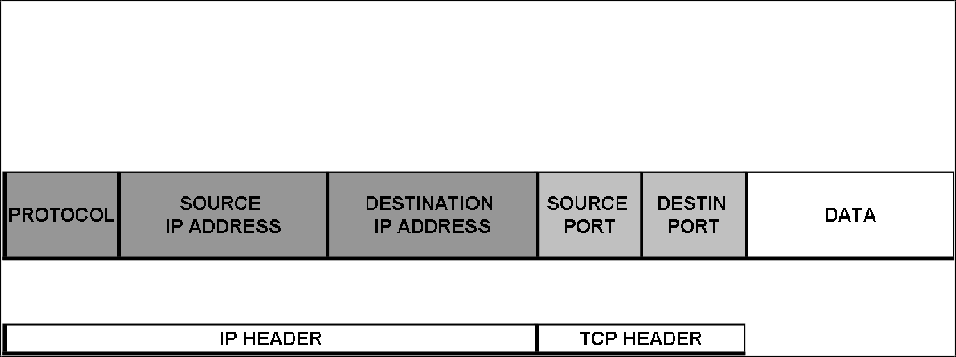
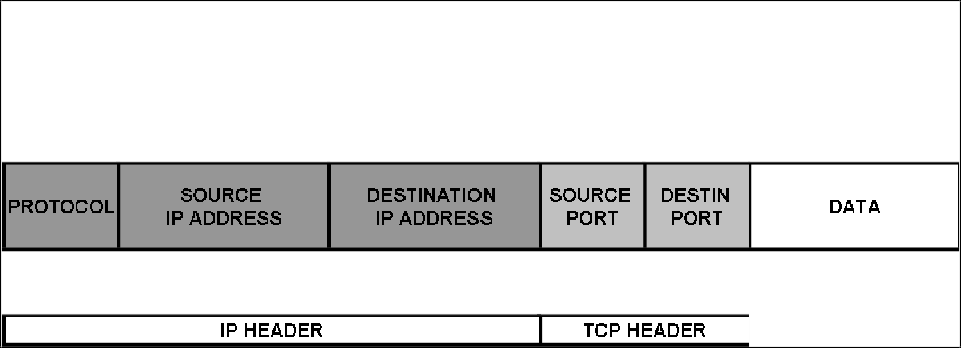
ACL Types:

Basically ACL’s are two types

* + - * + Standard ACL
        + Extended ACL

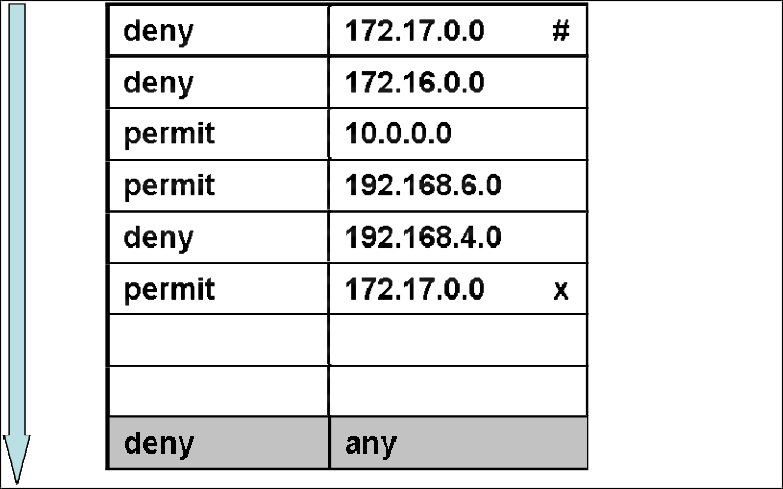
##### Standard – Extended ACL differences:

|  |  |
| --- | --- |
| Standard ACL | Extended ACL |
| It takes decisions based on Source IP | It can take decisions based on Source IP - Destination IP Protocol  Source Port - Destination Port |
| Implemented close to destination | Implemented close to source |
| ACL creation Number: 1-99 | ACL creation Number: 100-199 |
| Works on both direction | Works on single direction |
| Works on all services | Can work on single service |



ACL statements order:

* + - * Router checks ACL statements from top to bottom to find a match
      * If a match is found, router will not check further statements
      * “deny any” statement presents as a last statement in the ACL list, which is called implicit deny. So by default router blocks everything with ACL
      * Implicit deny can be overridden by “permit any” statement

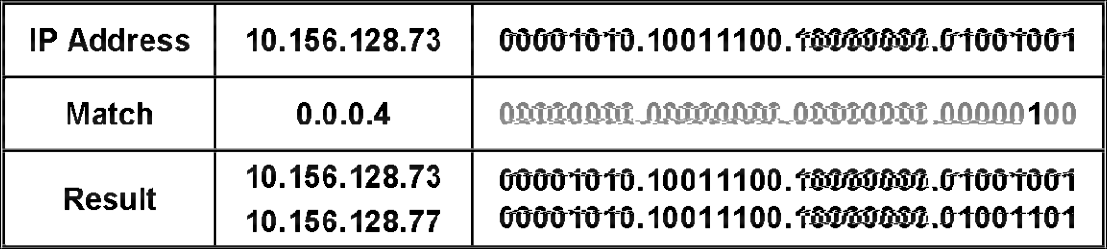


##### What is Match?

* + - * + Match is 32 bit value that defines the scope of IP address
        + It is similar to Wild card Mask value
        + 0 is must match: 1 is ignore
        + It indicates on what range IP addresses action should be taken

Examples:

|  |  |  |
| --- | --- | --- |
| IP Address | Match | Action taken on |
| 172.16.5.145 | 0.0.0.0 | 172.16.5.145 |
| 172.16.5.145 | 0.0.0.255 | 172.16.5.0 to 172.16.5.255 |
| 172.16.5.145 | 0.0.255.255 | 172.16.0.0 to 172.16.255.255 |
| 172.16.5.145 | 0.255.255.255 | 172.0.0.0 to 172.255.255.255 |
| 172.16.5.145 | 255.255.255.255 | All ip addresses |
| 10.0.0.123 | 255.255.255.255 | All ip addresses |
| 10.156.128.73 | 0.255.255.255 | 10.0.0.0 to 10.255.255.255 |
| 10.156.128.73 | 0.0.0.4 | 10.156.128.73 & 10.156.128.77 |



In the Match 3rd Bit indicates ignore. So 0 or 1 can be taken in the ip address 3rd bit That results two IP addresses

IP addressing, Binary operations knowledge is required to understand this concept

## **ACL implementation**

* + - * First understand the requirement
      * Identify source ip, destination ip, protocol, source port, destination port
      * Select the type of ACL ( Standard / Extended) to implement
      * Identify the traffic flow (in bound, out bound)
      * Select the router as a filtering point
      * Create ACL on the router and implement ACL on appropriate interface

First understand the requirement:

* + - * First understand need to implement ACL
      * Which traffic should be denied and which traffic should be allowed

Identify source ip, destination ip, protocol, source port, destination port:

* + - * Identify IP addresses from which to which the traffic should be filtered
      * Identify IP protocol to filter the traffic
* TCP
* UDP
* IP
* Eigrp
* Icmp
* Ospf

* + - * Identify TCP/UDP ports that should be filtered
* http
* ftp
* dns
* smtp
* telnet

Select the type of ACL ( Standard / Extended) to implement:

* + - * Select Standard or Extended ACL which is best suitable for the task
      * Standard ACL is a subset of Extended ACL
      * Extended ACL can be implemented for all types of scenarios

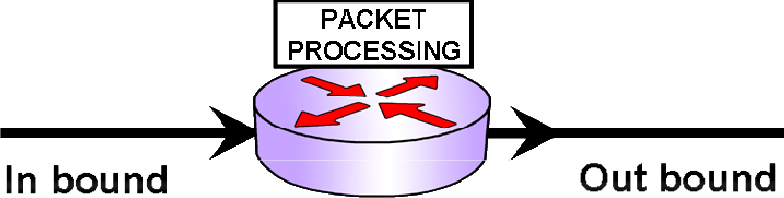
Identify traffic flow (in bound, out bound):

* + - * In bound : The traffic entering in to the router

It filters the traffic before the packet is processed, does not consume router resources

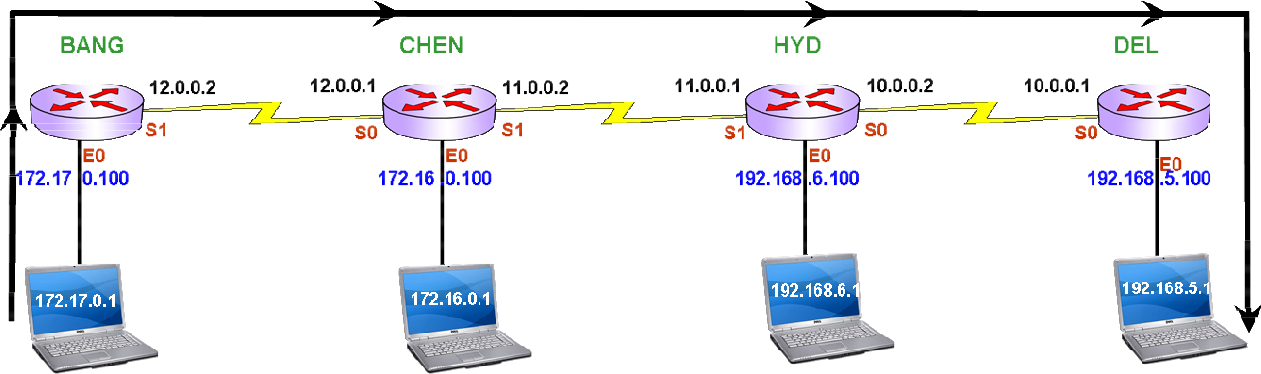
* + - * Out bound : The traffic leaving from router

It filters traffic after the packet is processed, consumes router resources



Select a router as a filtering point:

* + - * There may be number of routers appear in the traffic flow
      * Select one of the router as a best filtering point
      * Generally ACL is configured nearer to source or destination



Create ACL on the router and implement ACL on appropriate interface :

* + - * Create ACL in global configuration mode
      * Implement ACL on one of the interface, in interface mode
      * Maximum two ACL’s can be applied on one interface
* One as inbound
* Second as outbound

## **ACL Syntax**

##### **Standard ACL**

Creation:

Router(config)# access-list Router(config)# access-list

Router(config)# access-list

Implementation:

<1-99>

<1-99>

<1-99>

<permit/deny>

<permit/deny>

<deny/permit>

<sourceip>

<sourceip>

any

<match>

<match>

Router(config)# interface <s0/e0/s1>

Router(config-if)# ip access-group <1-99> <in/out>

**Extended ACL**

Creation:

Router(config)# access-list <100-199> <permit/deny> <protocol> <sourceip> <match> <destination ip> <match> <eq> <port> Router(config)# access-list <100-199> <permit/deny> <protocol> <sourceip> <match> <destination ip> <match> <eq> <port>

Router(config)# access-list <100-199> <deny/permit> <protocol>

Implementation:

Router(config)# interface <s0/e0/s1>

any

any

Router(config-if)# ip access-group <100-199> <in/out>

Checking ACL:

Router # show access-list

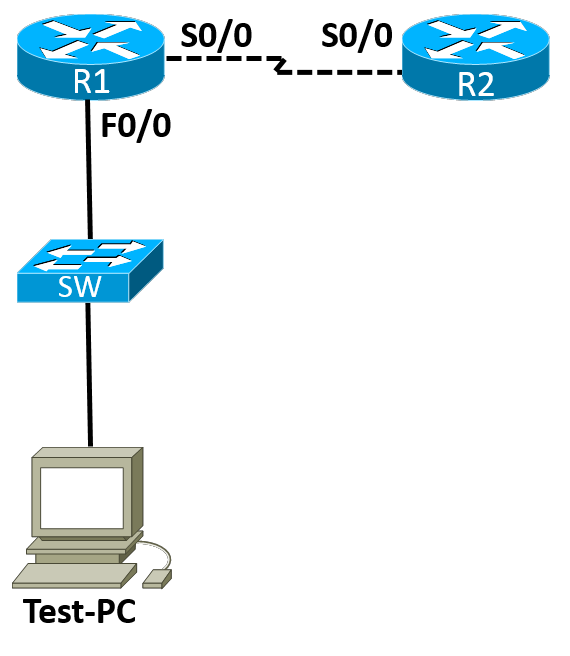
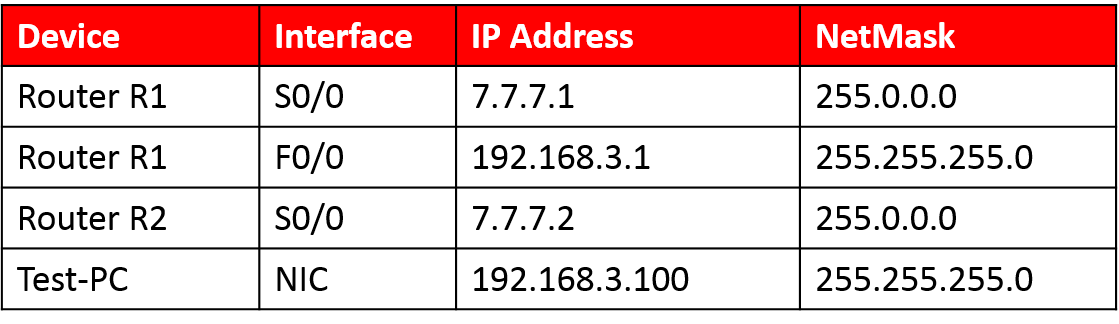
Deleting ACL:

Router(config)# no access-list <ACL No>

**LAB 7: Configuring Secure Shell (SSH)**

**Lab** **Topology**

**Steps** **to** **Configure** **SSH:**



1. Configure hostname

2. Configure domainname 3Generate Keys

4. Create user

5. Apply Authentication on line vty

**Step** **1:** **Configuring** **Hostname**

R2(config)#hostname R\_2

**Step** **2:** **Configuring** **Domain** **name**

R\_2(config)#ip domain-name abc.com

**Step** **3:** **Generate** **keys**

R\_2(config)#crypto key generate rsa

The name for the keys will be: R\_2.abc.com

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R\_2(config)#

\*Mar 1 00:10:11.495: %SSH-5-ENABLED: SSH 1.99 has been enabled

**Step 4: Create a User**

R\_2(config)#username user1 privilege 15 password cisco123

**Step** **5:** **Apply** **authentication** **method** **on** **line** **vty**

R\_2(config)#line vty 0 4 R\_2(config-line)#login local R\_2(config-line)#exit

LAB 1: CONFIGURING SECURE SHELL (SSH)

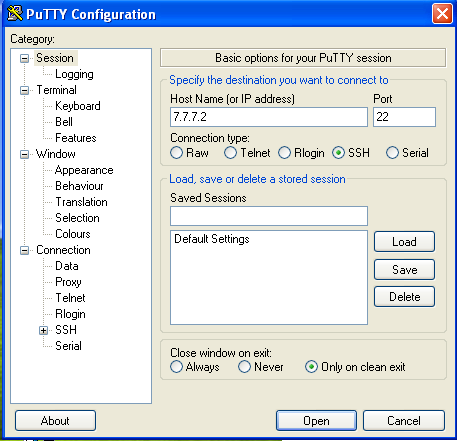
**Verification:** SSH using a router R1#ssh -l user1 7.7.7.2

Password: R\_2#

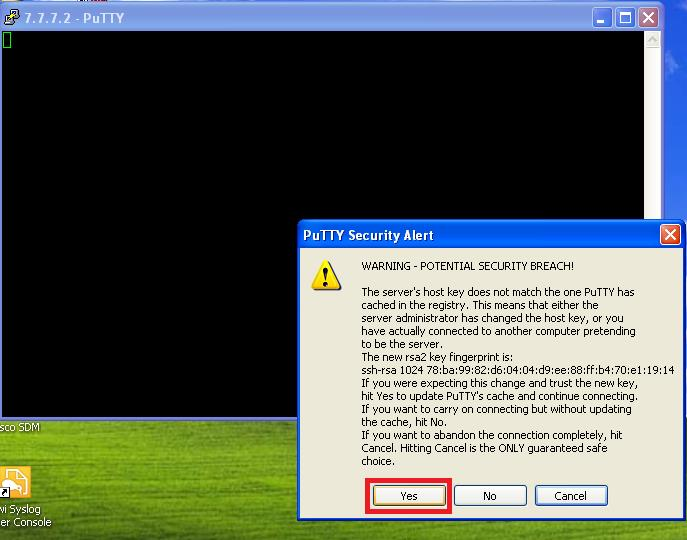
R\_2#

**Verification:** SSH from Test-PC using Putty

**Step** **1:** Enter the Router R2 IP address, select SSH and click on Open.



**Step** **2:** By clicking on YES accept the keys

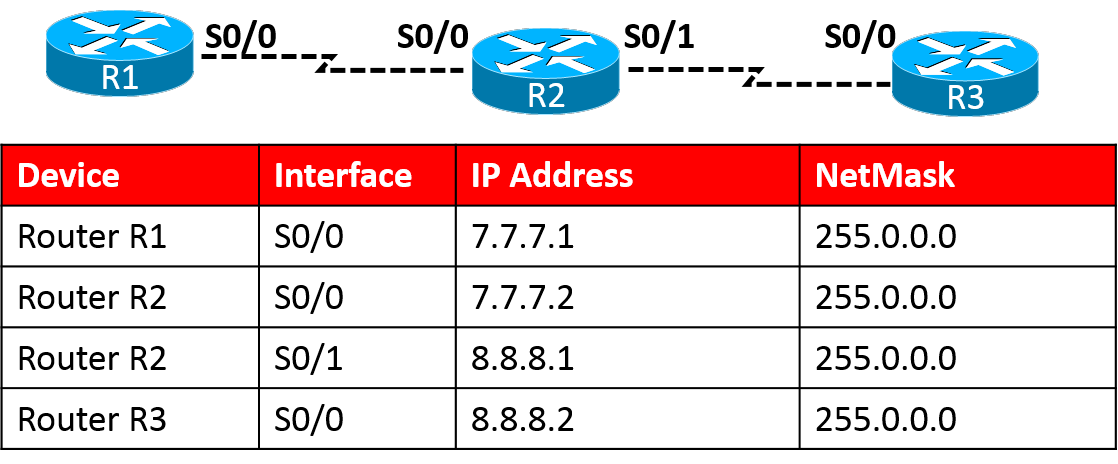
**Step** **3:** Enter username, password and then press Enter.



LAB 7: CONFIGURING PRIVILEGE LEVELS

**Lab 8: Configuring Privilege levels**

**Lab** **Topology**

**Steps:**

1. Create users according privilege levels

2. Assigns commands to users according to their privileges

3. Login from other router and verify

**Step** **1:** **Create** **users** **according** **to** **privilege** **level**

LAB 7: CONFIGURING PRIVILEGE LEVELS

R2(Config)#username user4 privilege 4 password 0 cisco R2(Config)#username user5 privilege 5 password 0 cisco R2(Config)#username user6 privilege 6 password 0 cisco

**Step** **2:** **Define** **commands** **according** **to** **user** **privilege** **levels**

**Level** **4** **user** **is** **only** **for** **monitoring** **purpose** **hence** **giving** **him** **all** **show** **commands** **authorization**

R2(Config)#privilege exec level 4 show all

**Level** **5** **user** **is** **going** **to** **deal** **with** **all** **routing** **task** **hence** **giving** **him** **all** **route** **commands** **authorization**

R2(Config)#privilege exec level 5 configure terminal R2(Config)#privilege configure level 5 router all R2(Config)#privilege configure level 5 ip route all

**Level** **6** **user** **is** **going** **to** **deal** **with** **security** **services** **hence** **giving** **him** **authorization** **for** **access-list** **and** **inspection** **commands**

R2(Config)#privilege configure level 6 access-list all R2(Config)#privilege configure level 6 ip access-list all R2(Config)#privilege configure level 6 ip inspect all

**Applying** **Authentication** **method** **on** **line** **vty**

R2(config)#line vty 0 4 R2(config-line)#login local R2(config-line)#exit

B 7: CONFIGURING PRIVILEGE LEVELS

**Verification:** **Login** **from** **Router** **R1** **into** **Router** **R2** **and** **verify** **privilege** **level**

R1#7.7.7.2

Trying 7.7.7.2 ... Open

User Access Verification

Username: user6 Password:

R2#

R2#show privilege

Current privilege level is 6

**Verification:** **Login** **from** **Router** **R1** **into** **Router** **R2** **and** **verify** **privilege** **level**

R1#7.7.7.2

Trying 7.7.7.2 ... Open

User Access Verification

Username: user4 Password:

R2#

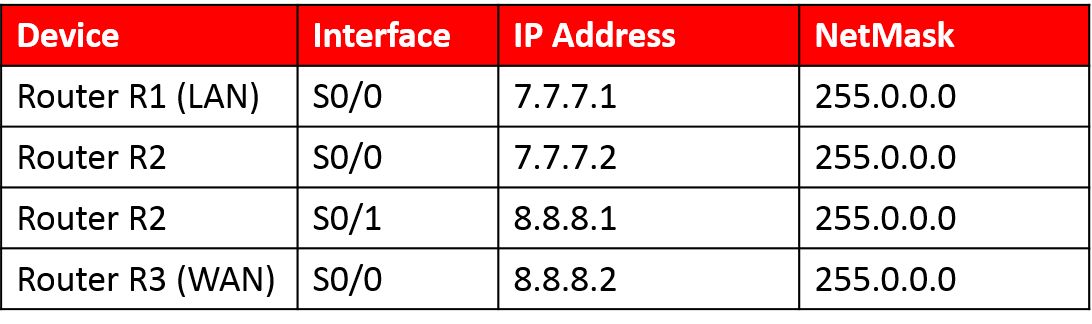
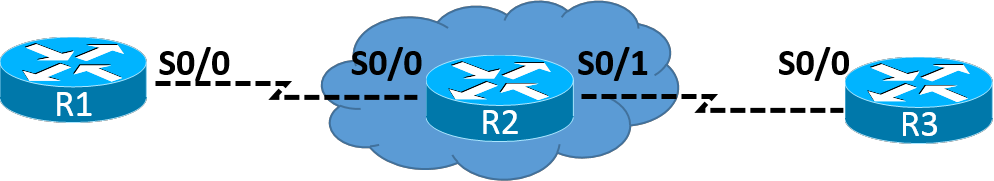
R2#show privilege

Current privilege level is 4 !

LAB 11: CONFIGURING ZONE BASED FIREWALL (ZBF

**Lab 9: Configuring Zone Based Firewall (ZBF)**

Lab Topology

Steps:

1. Create Zones

2. Associate Zones with the interfaces

3. Define interesting traffic using Class- map

4. Define Action over the interesting Traffic

5. Pair the Zones

6. Apply the policy over the Pair

LAB 11: CONFIGURING ZONE BASED FIREWALL (ZB

**Task** **1:** Create zones for Inside and outside network

R2(config)#zone security LAN R2(config-sec-zone)#exit

R2(config)#zone security WAN R2(config-sec-zone)#exit

**Task** **2:** Associate Zones with the interfaces

R2(config)#Interface Serial0/0 R2(config-if)#zone-member security LAN R2(config-if)#exit

R2(config)#Interface Serial0/1 R2(config-if)#zone-member security WAN R2(config-if)#exit

**Task** **3:** Define the interesting Traffic using Class-map

R2(config)#class-map type inspect match-any fls-cmap R2(config-cmap)#match protocol icmp

R2(config-cmap)#match protocol telnet R2(config-cmap)#match protocol http R2(config-cmap)#match protocol ssh R2(config-cmap)#exit

**Task** **4:** Define action over the interesting Traffic using Policy-map

R2(config)#policy-map type inspect fls-pmap R2(config-pmap)#class fls-cmap

R2(config-pmap-c)#inspect R2(config-pmap-c)#exit R2(config-pmap)#exit

LAB 11: CONFIGURING ZONE BASED FIREWALL (ZBF

**Task** **5:** Pair the zones and apply the policy over the zone-pair

R2(config)#zone-pair security zpair source LAN destination WAN R2(config-sec-zone-pair)#service-policy type inspect fls-pmap R2(config-sec-zone-pair)#exit

**Verification:**

LAN(config)#do ping 8.8.8.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 8.8.8.2, timeout is 2 seconds: !!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/86/180 ms

WAN(config)#do ping 7.7.7.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 7.7.7.1, timeout is 2 seconds: .....

Success rate is 0 percent (0/5)

LAB 12: CONFIGURING LAYER 2 SECURITY

**Lab 10: Configuring Layer 2 Security**

**VLAN** **Hopping** **Attack**

1. Always use a dedicated VLAN ID for all trunk ports

2. Disable unused ports and put them in an unused VLAN

3. Do not use VLAN 1 for anything

4. Disable auto-trunking on user facing ports (DTP)

5. Explicitly configure trunking on infrastructure ports

6. Use all tagged mode for the Native VLAN on trunks

7. Use 802.1q tag all on the trunk port

**MAC** **Attacks**

**CatOS**

set port security 5/1 enable

set port security 5/1 port max 3

set port security 5/1 violation restrict set port security 5/1 age 2

set port security 5/1 timer-type inactivity

**IOS**

switchport port-security switchport port-security maximum 3

switchport port-security violation restrict switchport port-security aging time 2 switchport port-security aging type inactivity

LAB 12: CONFIGURING LAYER 2 SECURITY

**IOS** **New** **Commands**

switchport port-security

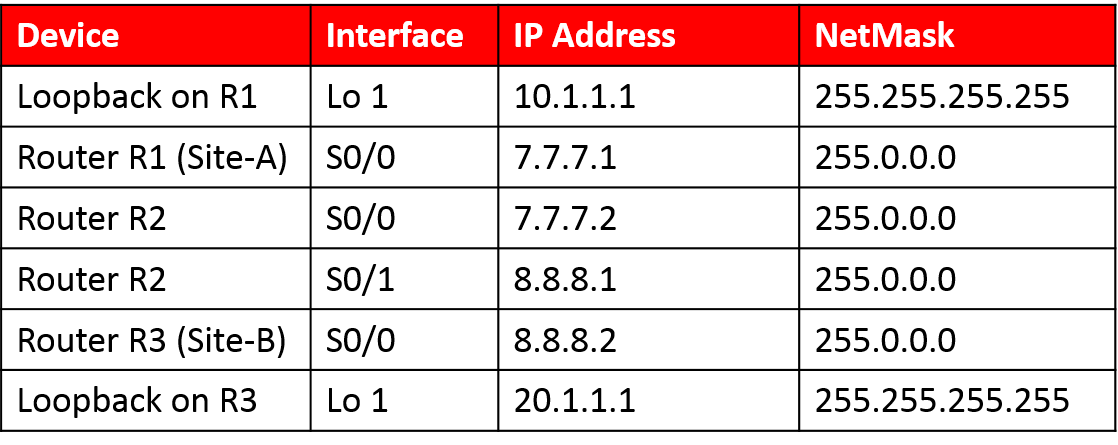
switchport port-security maximum 1 vlan voice switchport port-security maximum 1 vlan access switchport port-security violation restrict switchport port-security aging time 2 switchport port-security aging type inactivity

snmp-server enable traps port-security trap-rate 5

LAB 13: VIRTUAL PRIVATE NETWORK (VPN)

**Lab 11: Virtual Private Network (VPN)**

Lab Topology

**Steps**

1. Configure VPN on Site-A

2. Configure VPN on Site-B

LAB 13: VIRTUAL PRIVATE NETWORK (VPN)

**Task** **1:** Configuring VPN on Site-A i.e., on Router R1

R1(config)#hostname Site-A

**Defining** **ISAKMP** **Policy**

Site-A(config)#crypto isakmp policy 26 Site-A(config-isakmp)#encryption 3des Site-A(config-isakmp)#group 2

Site-A(config-isakmp)#hash sha

Site-A(config-isakmp)#authentication pre-share Site-A(config-isakmp)#exit

**Define** **Shared** **Secret** **and** **peer** **address**

Site-A(config)#crypto isakmp key flsvpn address 8.8.8.2

**Define** **IPSec** **attributes**

Site-A(config)#crypto ipsec transform-set seta esp-3des esp-sha-hmac Site-A(cfg-crypto-trans)#exit

**Define** **the** **interesting** **traffic**

Site-A(config)#access-list 165 permit ip host 10.1.1.1 host 20.1.1.1

**Bind** **all** **the** **attributes** **to** **be** **applied** **on** **interface** **using** **crypto** **map**

Site-A(config)#crypto map sitea-map 16 ipsec-isakmp

% NOTE: This new crypto map will remain disabled until a peer and a valid access list have been configured.

Site-A(config-crypto-map)#match address 165 Site-A(config-crypto-map)#set peer 8.8.8.2

Site-A(config-crypto-map)#set transform-set seta Site-A(config-crypto-map)#exit

**Apply** **the** **Crypto** **map** **on** **the** **respected** **interface**

Site-A(config)#int s0/0

Site-A(config-if)#crypto map sitea-map

\*Mar 1 00:32:56.579: %CRYPTO-6-ISAKMP\_ON\_OFF: ISAKMP is ON Site-A(config-if)#end

**Task** **2:** Configure VPN on Site-B i.e., on Router R3

R3(config)#hostname Site-B

**Defining** **ISAKMP** **Policy**

Site-B(config)#crypto isakmp policy 25 Site-B(config-isakmp)#encryption 3des Site-B(config-isakmp)#hash sha

Site-B(config-isakmp)#group 2

Site-B(config-isakmp)#authentication pre-share Site-B(config-isakmp)#exit

**Define** **Shared** **Secret** **and** **peer** **address**

Site-B(config)#crypto isakmp key flsvpn address 7.7.7.1

**Define** **IPSec** **attributes**

Site-B(config)#crypto ipsec transform-set setb esp-3des esp-sha-hmac Site-B(cfg-crypto-trans)#exit

**Define** **the** **interesting** **traffic**

Site-B(config)#access-list 155 permit ip host 20.1.1.1 host 10.1.1.1

**Bind** **all** **the** **attributes** **to** **be** **applied** **on** **interface** **using** **crypto** **map**

Site-B(config)#crypto map siteb-map 11 ipsec-isakmp

% NOTE: This new crypto map will remain disabled until a peer and a valid access list have been configured.

Site-B(config-crypto-map)#match address 155

Site-B(config-crypto-map)#set peer 7.7.7.1

Site-B(config-crypto-map)#set transform-set setb Site-B(config-crypto-map)#exit

**Apply** **the** **Crypto** **map** **on** **the** **respected** **interface**

Site-B(config)#int s0/0

Site-B(config-if)#crypto map siteb-map

\*Mar 1 00:27:41.479: %CRYPTO-6-ISAKMP\_ON\_OFF: ISAKMP is ON Site-B(config-if)#exit

**Verification** **Commands**

Site-A#ping 20.1.1.1 source loopback 1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds: Packet sent with a source address of 10.1.1.1

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 172/196/208 ms

Site-B#ping 10.1.1.1 source loopback 1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds: Packet sent with a source address of 20.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 152/175/196 ms

Site-B#show crypto ipsec sa interface: Serial0/0

Crypto map tag: siteb-map, local addr 8.8.8.2

protected vrf: (none)

local ident (addr/mask/prot/port): (20.1.1.1/255.255.255.255/0/0) remote ident (addr/mask/prot/port): (10.1.1.1/255.255.255.255/0/0)

current\_peer 7.7.7.1 port 500 PERMIT, flags={origin\_is\_acl,}

#pkts encaps: 9, #pkts

#pkts decaps: 9, #pkts

encrypt: 9, decrypt: 9,

#pkts digest: 9

#pkts verify: 9

#pkts compressed: 0, #pkts decompressed: 0 #pkts not compressed: 0, #pkts compr. failed: 0

#pkts not decompressed: 0, #pkts decompress failed: 0 #send errors 0, #recv errors 0

local crypto endpt.: 8.8.8.2, remote crypto endpt.: 7.7.7.1 path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0

current outbound spi: 0x10E4A8A6(283420838)

inbound esp sas:

spi: 0x594F23E9(1498358761) transform: esp-3des esp-sha-hmac , in use settings ={Tunnel, }

conn id: 1, flow\_id: SW:1, crypto map: siteb-map

sa timing: remaining key lifetime (k/sec): (4469160/3489) IV size: 8 bytes

replay detection support: Y Status: ACTIVE

inbound ah sas:

inbound pcp sas:

outbound esp sas:

spi: 0x10E4A8A6(283420838) transform: esp-3des esp-sha-hmac , in use settings ={Tunnel, }

conn id: 2, flow\_id: SW:2, crypto map: siteb-map

sa timing: remaining key lifetime (k/sec): (4469160/3483) IV size: 8 bytes

replay detection support: Y Status: ACTIVE

outbound ah sas:VIRTUAL PRIVATE NETWORK (VPN)

outbound pcp sas:

Site-B#show crypto isakmp sa IPv4 Crypto ISAKMP SA

Dst src state conn-id slot status

8.8.8.2 7.7.7.1 QM\_IDLE 1001 0 ACTIVE

IPv6 Crypto ISAKMP SA