

**REPORT OF DESIGNING A NETWORK STRUCTURE**

**CREATING A NETWORK TOPOLOY**

**WITH**

**PROVIDING SECURITY**

**OWNER**

**Trust-in-depth**

**DATE: 30/05/2025**

**Table of Contents**

**1.Contents 2**

**2.Network Topology Overview4**

**3.Hostname Configurations 5**

HQ Side5

Branch Side5

**Internet (Remote, ISP, Unrelated)** **5**

**4.VLAN Configuration6**

**1.HQ SIDE6**

Inter-VLAN Routing6

Sub-Interfaces6

Native VLAN7

Management VLAN8

**2.BRANCH SIDE9**

Inter-VLAN Routing9

Sub-Interfaces9

Native VLAN10

Management VLAN11

**3. Explanation of command VLANS12**

Vtp(Rute-on-a-stick) 12

Native13

Management13

**5.IP Addressing14**

**1.DHCP Configuration14**

HQ DHCP Router14

ISP\_DHCP SERVER15

**2.Manual IP 16**

Manually Configured PC’S 16

Manually Configured Servers17

Manually Configured Router Interfaces18

**6.Routing (EIGRP)** **20**

Configuration Per Router20

Summarization and Wildcard Masks21

**7.NAT Configuration**22

HQ Side22

Branch Side23

**8.Server Configurations25**

Web Server25

DNS Server25

FTP Server25

Mail Server26

DHCP Server26

**9.Security Measures**27

ACL Rules and Lists27

SSH and Password Policies29

MAC Address Port Security32

**10.WAN Configuration (PPP)** 34

PPP Encapsulation34

IP Address Assignments34

**11.VPN Configuration35**

ISAKMP & IPSec Setup35

ACL for VPN Traffic36

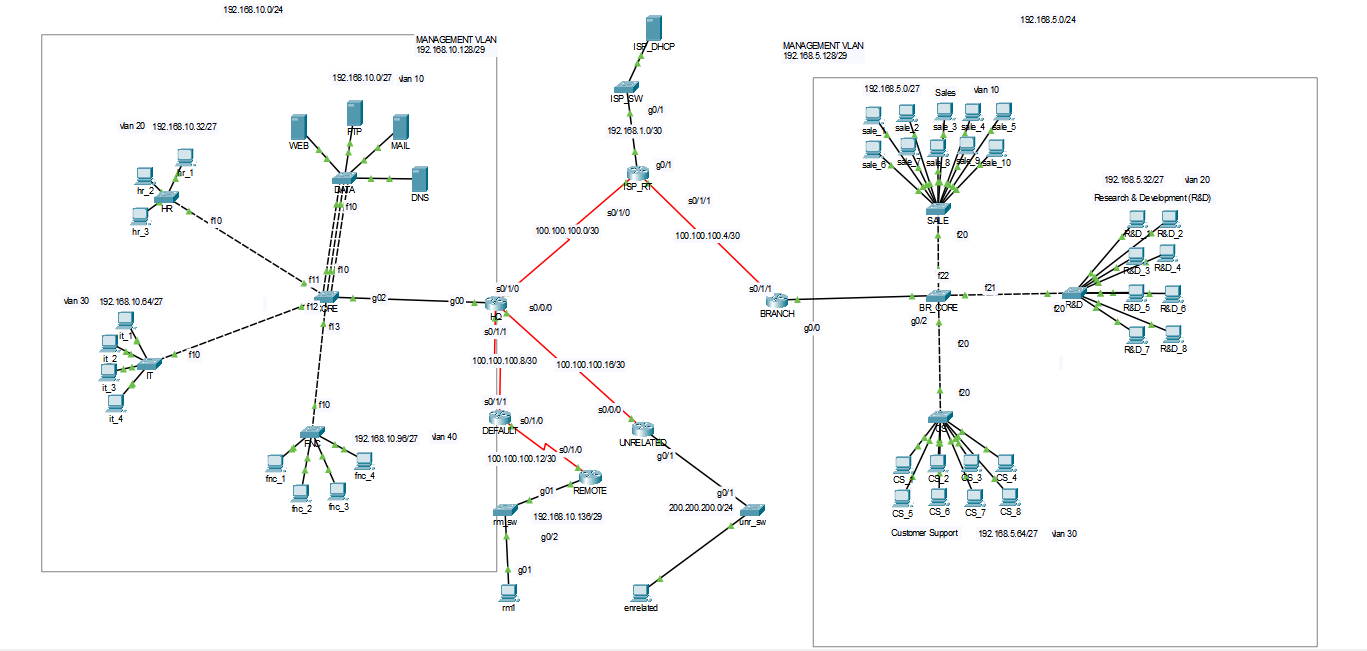
Crypto Map Binding36

**12.Etherchannel37**

**13.STP(Spanning Tree Protocol)** **38**

**14.The Problems I Faced**39

**TOPOLOGY**

****

**Hostname Configurations**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| hq\_router | core\_switch | data\_switch | hr\_switch | it\_switch | finance\_switch |
| Switch#en  Switch#conf t  Switch(config)#host HQ  HQ(config)# | Switch#en  Switch#conf t  Switch(config)#host  CORE  CORE(config)# | Switch>en  Switch#conf t  Switch(config)#host DATA  DATA(config)# | Switch>en  Switch#conf t  Switch(config)#host HR  HR(config)# | Switch>en  Switch#conf t  Switch(config)#host IT  IT(config)# | Switch>en  Switch#conf t  Switch(config)#host FNC  FNC(config)# |

hq\_side

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| branch\_router | branch\_core\_switch | sale\_switch | research\_switch | costumer\_sup\_switch |
| Router>en  Router#conf t  Router(config)#host BRANCH  BRANCH(config)# | Switch>en  Switch#conf t  Switch(config)#host BR\_CORE  BR\_CORE(config)# | Switch>en  Switch#conf t  Switch(config)#host SALE  SALE(config)# | Switch>en  Switch#conf t  Switch(config)#host R&D  R&D(config)# | Switch>en  Switch#conf t  Switch(config)#host CS  CS(config)# |

branch\_side  
  
  
Internet\_side(REMOTE)

|  |  |  |
| --- | --- | --- |
| DEFAULT(router) | REMOTE(router) | Rm\_sw(switch) |
| Router>en  Router#conf t  Router(config)#host DEFAULT  DEFAULT (config)# | Router>en  Router#conf t  Router(config)#host REMOTE  REMOTE (config)# | Switch>en  Switch#conf t  Switch(config)#host rm\_sw  rm\_sw(config)# |

Internet\_side(ISP)

|  |  |
| --- | --- |
| ISP\_RT(Router) | ISP\_SW(Switch) |
| Router>en  Router#conf t  Router(config)#host ISP\_RT  ISP\_RT(config)# | Switch>en  Switch#conf t  Switch(config)#host ISP\_SW  ISP\_SW(config)# |

Internet\_side(UNRELATED)

|  |  |
| --- | --- |
| UNRELATED(Router) | unr\_sw(switch) |
| Router>en  Router#conf t  Router(config)#host UNRELATED  UNRELATED(config)# | Switch>en  Switch#conf t  Switch(config)#host unr\_sw  unr\_sw(config)# |

**What do those commands of hostname configuration do?**

Router>en #en means “enable”, it turn device enable mode to be full connected

Router#conf t # it is “configure terminal”. We can change settings

Router(config)#host ------- #Host is “hostname”. With this we give a name to our device

UNRELATED(config)# #We can see the changes -Router is converted to UNRELATED

**Summary:**

* **We are using it to give a name each our devices**

**VLANS**

**HQ SIDE**

**-- Inter-VLAN Routing(Router-on-a-Stick)--**

***Sub-interfaces***

|  |  |
| --- | --- |
| HQ (config-if)#int g0/0.10  HQ (config-subif)#encapsulation dot1Q 10  HQconfig-subif)#ip address 192.168.10.1 255.255.255.224  HQ (config-subif)#int g0/0.20  HQ(config-subif)#encapsulation dot1Q 20  HQ (config-subif)#ip address 192.168.10.33 255.255.255.224  HQ (config-subif)#int g0/0.30  HQ (config-subif)#encapsulation dot1Q 30  HQ (config-subif)#ip address 192.168.10.65 255.255.255.224  HQ (config-subif)#int g0/0.40  HQ (config-subif)#encapsulation dot1Q 40  HQ (config-subif)#ip address 192.168.10.97 255.255.255.224 |  |
| -----No shut down  HQ(config)#int g0/0  HQ(config-if)#no sh |

**What those commands above(sub-interface) do?**

HQ (config-subif)#int g0/0.20 #making sub interface in this there is g0/0 interface and we split that interfaces as a sub-interface

HQ(config-subif)#encapsulation dot1Q 20 #we are tagging the data to demonstrate that which data is going where

HQ (config-subif)#ip address 192.168.10.33 255.255.255.224 #giving IP address to sub-interface

HQ(config)#int g0/0 #going real interface

HQ(config-if)#no sh #prentent to be down

\*WE do this because we want to vlans to can communicate outside device so there will be each vlan gateway address

***VTP Configuration (VLAN Trunking Protocol)***

Step 1:

CREATE VTP NETWORK   
-Create a domain for vtp

-Set the core\_sw as a vtp server and give a password

-Created vlans

Step 2:

DEFINE RELATIONS( SERVER-CLIENT)

On each switch excluded core;

-Create their vlans( it is important

-Give same domain name for each

-Make them a client

- Give them a password

|  |  |
| --- | --- |
| CORE Switch | The structure |
| CORE>en  CORE#conf t  CORE(config)#vtp domain VTP\_NETWORK  CORE(config)#vtp mode server  Device mode already VTP SERVER.  CORE(config)#vtp password **core**  CORE(config)#vlan 10  CORE(config-vlan)#name DATA  CORE(config-vlan)#vlan 20  CORE(config-vlan)#name HR  CORE(config-vlan)#vlan 30  CORE(config-vlan)#name IT  CORE(config-vlan)#vlan 40  CORE(config-vlan)#name FNC |  |

|  |  |  |  |
| --- | --- | --- | --- |
| DATA\_SWITCH | HR\_SWITCH | IT\_SWITCH | FNC\_SWITCH |
| DATA#conf t  DATA(config)#vtp domain VTP\_NETWORK  DATA(config)#vtp mode client  DATA(config)#vtp password **core** | HR#en  HR#conf t  HR(config)#vtp domain VTP\_NETWORK  HR(config)#vtp mode client  HR(config)#vtp password **core** | IT>en  IT#conf t  IT(config)#vtp domain VTP\_NETWORK  IT(config)#vtp mode client  IT(config)#vtp password **core** | FNC>en  FNC#conf t  FNC(config)#vtp domain VTP\_NETWORK  FNC(config)#vtp mode client  FNC(config)#vtp password **core** |

Step 3:

DETERMINE PORT WHICH WILL HAS TRUNK, WHICH WILL HAS ACCESS MODE

--Organise the ports either trunk or access

-Configure each end\_device interface connected to switch as access mode

-make device connect to vlan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATA\_SWITCH | HR\_SWITCH | IT\_SWITCH | FNC\_SWITCH |  |
| DATA>en  DATA#conf t  DATA(config)#int range f0/1-4  DATA(config-if-range)#switchport mode access  DATA(config-if-range)#switchport access vlan 10  DATA(config-if-range)#ex  DATA(config)#int f0/10  DATA(config-if)#switchport mode trunk | HR#en  HR#conf t  HR(config)#int range f0/1-3  HR(config-if-range)#switchport mode access  HR(config-if-range)#switchport access vlan 20  HR(config-if-range)#ex  HR(config)#int f0/10  HR(config-if)#switchport mode trunk | IT>en  IT#conf t  IT(config)#int range f0/1-4  IT(config-if-range)#switchport mode access  IT(config-if-range)#switchport access vlan 30  IT(config-if-range)#ex  IT(config)#int f0/10  IT(config-if)#switchport mode trunk | FNC#en  FNC#conf t  FNC(config)#int range f0/1-4  FNC(config-if-range)#switchport mode access  FNC(config-if-range)#switchport access vlan 40  FNC(config-if-range)#ex  FNC(config)#int f0/10  FNC(config-if)#switchport mode trunk | CORE>en  CORE#conf t  CORE(config)#int range f0/2-5  CORE(config-if-range)#switchport mode trunk |

**Native Vlan:**

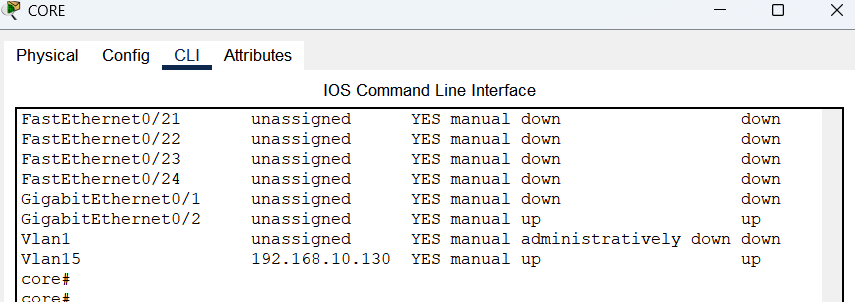
**The native VLAN is used for untagged traffic on a trunk port**

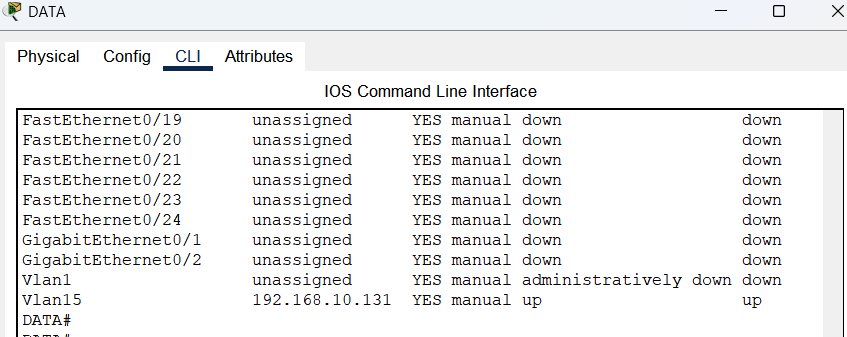
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **core** | **data** | **hr** | **it** | **fnc** |
| core>en  core#conf t  core(config)#vlan 99  core(config-vlan)#name Native\_VLAN  core(config-vlan)#int range f0/10-13  core(config-if-range)#switchport trunk native vlan 99  core(config-if-range)#int g0/2  core(config-if)#sw trunk native vlan 99 | DATA(config)#int f0/10  DATA(config-if)#switchport access vlan 99  DATA(config-if)#switchport trunk native vlan 99 | HR#conf t  HR(config)#int f0/10  HR(config-if)#switchport access vlan 99  HR(config-if)#switchport trunk native vlan 99 | IT#conf t  IT(config)#int f0/10  IT(config-if)#sw ac vlan 99  IT(config-if)#sw tr nat vlan 99 | FNC#conf t  FNC(config)#int f0/10  FNC(config-if)#sw acc vlan 99  FNC(config-if)#sw tr nat vlan 99 |

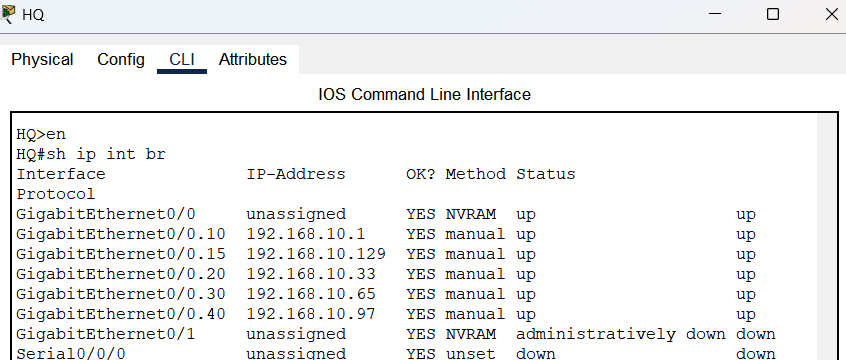
**Management VLAN**

**The management VLAN is used to manage the switch (SSH, Telnet, web, etc.)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **core** | **data** | **hr** | **it** | **fnc** | **hq** |
| core#conf t  core(config)#vlan 15  core(config-vlan)#name MANAGEMENT  core(config-vlan)#int vlan 15  core(config-if)#ip address 192.168.10.130 255.255.255.248  core(config)#ip default-gateway 192.168.10.129 | DATA(config)#int f0/10  DATA(config-if)#sw access vlan 15  DATA(config)#ex  DATA(config)#int vlan 15  DATA(config-if)#ip address 192.168.10.131 255.255.255.248  DATA(config-if)#no sh  DATA(config-if)#ex  DATA(config)#ip default-gateway 192.168.10.129 | HR#conf t  HR(config)#int f0/10  HR(config-if)#sw acc vlan 15  HR(config-if)#int vlan 15  HR(config-if)#ip address 192.168.10.132 255.255.255.248  HR(config-if)#no sh  HR(config-if)#ex  HR(config)#ip default-gateway 192.168.10.129 | IT#conf t  IT(config)#int f0/10  IT(config-if)#sw acc vlan 15  IT(config-if)#int vlan 15  IT(config-if)#ip add 192.168.10.133 255.255.255.248  IT(config-if)#no sh  IT(config-if)#ex  IT(config)#ip default-gateway 192.168.10.129 | FNC#conf t  FNC(config)#int f0/10  FNC(config-if)#sw acc vlan 15  FNC(config-if)#int vlan 15  FNC(config-if)#ip add 192.168.10.134 255.255.255.248  FNC(config-if)#no sh  FNC(config-if)#ex  FNC(config)#ip default-gateway 192.168.10.129 | HQ(config)#int g0/0.15  HQ(config-subif)#encapsulation dot1Q 15  HQ(config-subif)#ip address 192.168.10.129 255.255.255.248  Note:router olduğundan vlan gibi bir yapıdan çok sub inerface şeklinde yapılandırıldı. |

****

****

****

**BRANCH SIDE**

**-- Inter-VLAN Routing(Router-on-a-Stick)--**

**Sub-interfaces**

|  |  |
| --- | --- |
| BRANCH(config)#int g0/0.10  BRANCH(config-subif)#encapsulation dot1Q 10  BRANCH(config-subif)#ip address 192.168.5.1 255.255.255.224  BRANCH(config-subif)#int g0/0.20  BRANCH(config-subif)#encapsulation dot1Q 20  BRANCH(config-subif)#ip address 192.168.5.33 255.255.255.224  BRANCH(config-subif)#int g0/0.30  BRANCH(config-subif)#encapsulation dot1Q 30  BRANCH(config-subif)#ip address 192.168.5.65 255.255.255.224 |  |
| ---for no sut down  BRANCH(config)#int g0/0  BRANCH(config-if)#no sh |

***VTP Configuration (VLAN Trunking Protocol)***

|  |  |
| --- | --- |
| BR\_CORE Switch | The structure |
| BR\_CORE>en  BR\_CORE #conf t  BR\_CORE (config)#vtp domain BR\_NETWORK  BR\_CORE (config)#vtp mode server  Device mode already VTP SERVER.  BR\_CORE (config)#vtp password **br\_core**  BR\_CORE (config)#vlan 10  BR\_CORE (config-vlan)#name SALE  BR\_CORE (config-vlan)#vlan 20  BR\_CORE (config-vlan)#name R&D  BR\_CORE (config-vlan)#vlan 30  BR\_CORE (config-vlan)#name CS |  |

|  |  |  |
| --- | --- | --- |
| SALE(Switch) | R&D(Switch) | CS (Switch) |
| SALE#en  SALE#conf t  SALE (config)#vtp domain VTP\_NETWORK  SALE (config)#vtp mode client  SALE (config)#vtp password **br\_core** | R&D#en  R&D #conf t  R&D (config)#vtp domain VTP\_NETWORK  R&D (config)#vtp mode client  R&D (config)#vtp password **br\_core** | CS>en  CS#conf t  CS(config)#vtp domain VTP\_NETWORK  CS(config)#vtp mode client  CS(config)#vtp password **br\_core** |

|  |  |  |  |
| --- | --- | --- | --- |
| Br\_core | SALE(Switch) | R&D(Switch) | CS (Switch) |
| BR\_CORE(config)#int range f0/20-22  BR\_CORE(config-if-range)#switchport mode trunk  BR\_CORE(config)#int g0/2  BR\_CORE(config-if)#switchport mode trunk | SALE(config)#int range f0/1-10  SALE(config-if-range)#switchport mode access  SALE(config-if-range)#switchport access vlan 10  SALE(config-if-range)#ex  SALE(config)#int f0/20  SALE(config-if)#switchport mode trunk | R&D(config)#int range f0/1-8  R&D(config-if-range)#switchport mode access  R&D(config-if-range)#sw access vlan 20  R&D(config-if-range)#ex  R&D(config)#int f0/20  R&D(config-if)#switchport mode trunk | CS(config)#int range f0/1-8  CS(config-if-range)#switchport mode access  CS(config-if-range)#switchport access vlan 30  CS(config-if-range)#ex  CS(config)#int f0/20  CS(config-if)#switchport mode trunk |

**Interface Configuration(Trunk-Acces)**

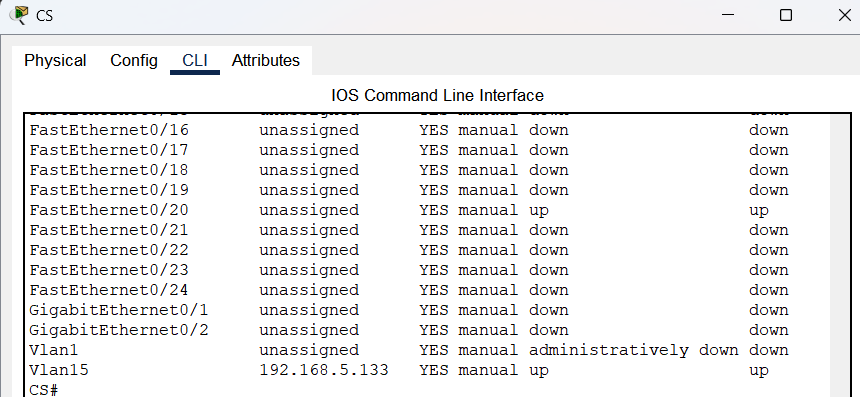
**Native Vlan:**

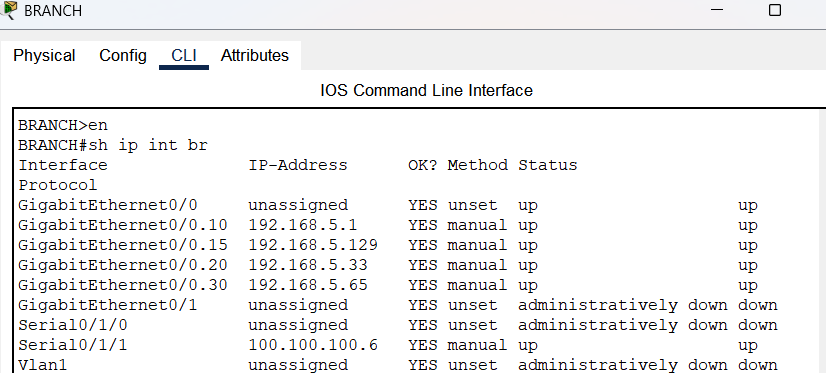
|  |  |  |  |
| --- | --- | --- | --- |
| **br\_core** | **sale** | **r&d** | **cs** |
| BR\_CORE#conf t  Enter configuration  BR\_CORE(config)#vlan 99  BR\_CORE(config-vlan)#name Native\_Vlan  BR\_CORE(config-vlan)#int range f0/20-22  BR\_CORE(config-if-range)#switchport trunk native vlan 99  BR\_CORE(config-if-range)#int g0/2  BR\_CORE (config-if)#sw trunk native vlan 99 | SALE#conf t  SALE(config)#int f0/20  SALE(config-if)#sw acc vlan 99  SALE(config-if)#sw tr nat vlan 99 | R&D#conf t  R&D(config)#int f0/20  R&D(config-if)#sw acc vlan 99  R&D(config-if)#sw tr nat vlan 99 | CS#conf t  CS(config)#int f0/20  CS(config-if)#sw acc vlan 99  CS(config-if)#sw tr nat vlan 99 |

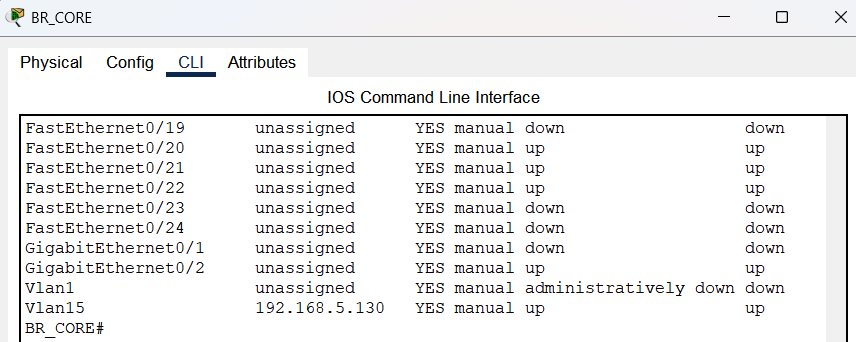
|  |  |  |
| --- | --- | --- |
| We can see the result of : br\_core,  Hr,  R&D  Beside them for the other native the structure of vlans will be like that |  |  |

**Management VLAN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Br\_core** | **sale** | **R&d** | **cs** | **branch** |
| BR\_CORE#conf t  BR\_CORE(config)#vlan 15  BR\_CORE(config-vlan)#name **MANAGEMENT**  BR\_CORE(config-vlan)#int vlan 15  BR\_CORE(config-if)#ip address 192.168.5.130 255.255.255.248  BR\_CORE(config-if)#ex  BR\_CORE(config)#ip default-gateway 192.168.10.129 | SALE#conf t  SALE(config)#int f0/20  SALE(config-if)#sw ac vlan 15  SALE(config-if)#int vlan 15  SALE(config-if)#ip address 192.168.5.131 255.255.255.248  SALE(config-if)#ex  SALE(config)#ip default-gateway 192.168.5.129 | R&D#conf t  R&D(config)#int f0/20  R&D(config-if)#sw ac vlan 15  R&D(config-if)#int vlan 15  R&D(config-if)#ip address 192.168.5.132 255.255.255.248  R&D(config-if)#ex  R&D(config)#ip default-gateway 192.168.5.129 | CS#conf t  CS(config)#int f0/20  CS(config-if)#sw ac vlan 15  CS(config-if)#int vlan 15  CS(config-if)#  CS(config-if)#ip address 192.168.5.133 255.255.255.248  CS(config-if)#ex  CS(config)#ip default-gateway 192.168.5.129 | BRANCH#conf t  BRANCH(config)#int g0/0.15  BRANCH(config-subif)#encapsulation dot1Q 15  BRANCH(config-subif)#ip address 192.168.5.129 255.255.255.248 |

****

****

****

**After Configuring vlans:**

|  |  |
| --- | --- |
|  | After configuration like beside,  T defines Trunk ports   A defines Access ports |

**Explanation of command VLANS**

**1-VTP (route-on-a stick)**

**ALL 3 Step commands. What do those command do?**

---Creating vlan network and determining server and client relationship---

CORE>en # Access to device

CORE#conf t #give permission to change setting which is terminal

CORE(config)#vtp domain VTP\_NETWORK # We are creating a domain network which is named VTP\_NETWORK

CORE(config)#vtp mode server

# We are saying that device will be server so clients cannot create their own vlan but server can do that and share it with clients

Device mode already VTP SERVER. #We get this because they are server as a default

CORE(config)#vtp password **core #**We have already made our network and now we are giving a password for it

CORE(config)#vlan 10 #We are creating vlans

CORE(config-vlan)#name DATA # naming the vlan

DATA#conf t

DATA(config)#vtp domain VTP\_NETWORK # We wanna get vlan name from VTP\_SERVER so we shoul be same network

DATA(config)#vtp mode client

#That command says we are client so we cannot create any vlan just can get them from VTR\_SERVER

DATA(config)#vtp password **core #**For that network we should have same password to reach network

---Defining which interface will be in trunk or access mode---

DATA(config)#int range f0/1-4 #to conf a range of interface at the same time

DATA(config-if-range)#switchport mode access #it is default mode to connect a device

DATA(config-if-range)#switchport access vlan 10 #it provides to connect to vlan 10 for all interface in range

DATA(config-if-range)#ex #it is “exit” to exit conf-if-range

DATA(config)#int f0/10 #we can go and configure f0/10 interface via this command

DATA(config-if)#switchport mode trunk #we turned mode to trunk to make access ports like a just one port

**Sum:**

* **Firstly, we are making a vlan network and then on switch side define who will be trunk and who will be access, also on router side we are making sub-interfaces to give gateway to vlans**

**2-Native Vlan**

core>en #to reach device

core#conf t #to make some changes on setting

core(config)#vlan 99 # creating vlan

core(config-vlan)#name Native\_VLAN # naming the vlan

core(config-vlan)#int range f0/10-13 # to go some interface range

core(config-if-range)#switchport trunk native vlan 99 #defining vlan 99 on range, as a native vlan

core(config-if-range)#int g0/2 #going to interface g0/2

core(config-if)#sw trunk native vlan 99 #creating native vlan on interface g0/2

**Sum:**

* **If we remember, at the begining we tagged traffic with dot1Q, in case of untagged traffic we will be using native vlan to route the untagged packet.**

**3-Management Vlan**

HR#conf t # being able to configuring

HR(config)#int f0/10 # to go interface

HR(config-if)#sw acc vlan 15

# it is actually “switchport access vlan 15”. In vtp system that means the server created vlan 15 and now client try to access it.

HR(config-if)#int vlan 15 #we accessed vlan 15 ant going interface of vlan 15

HR(config-if)#ip address 192.168.10.132 255.255.255.248 #giving management vlan ip address

HR(config-if)#no sh #to prevent it from being down

HR(config-if)#ex #exiting from the last mode

HR(config)#ip default-gateway 192.168.10.129 #determining gateway ip address to reach outside devices

**Sum:**

* **we are using management vlan to manage switches like sh,telnet, etc.**
* **We was split our IP mask /24 to two part: 1.one is 192.168.10.0/25 and another one is 192.168.128.0/25 so that we created them because that in any case we can use it. It that state, we use 2. part IPs to give vlan**

***IP ADDRESSING***

***DHCP CONFIGURATION***

***We have 2 system to serve us ips:***

***1-HQ (Router) =it provides ips to HQ(headquarter)***

***2-ISP\_DHCP(Server) = provides ips to all network under an ISP and of course for branch office.***

***HQ ROUTER SİDE***

-Creating pools

|  |
| --- |
| HQ (config)#ip dhcp pool vlan10  HQ (dhcp-config)#network 192.168.10.0 255.255.255.224  HQ(dhcp-config)#default-router 192.168.10.1  HQ (dhcp-config)#dns-server 192.168.10.8  HQ (dhcp-config)#ip dhcp pool vlan20  HQ (dhcp-config)#network 192.168.10.32 255.255.255.224  HQ (dhcp-config)#default-router 192.168.10.33  HQ (dhcp-config)#dns-server 192.168.10.8  HQ (dhcp-config)#ip dhcp pool vlan30  HQ (dhcp-config)#network 192.168.10.64 255.255.255.224  HQ (dhcp-config)#default-router 192.168.10.65  HQ (dhcp-config)#dns-server 192.168.10.8  HQ (dhcp-config)#ip dhcp pool vlan40  HQ (dhcp-config)#network 192.168.10.96 255.255.255.224  HQ (dhcp-config)#default-router 192.168.10.97  HQ (dhcp-config)#dns-server 192.168.10.8 |

-excluding gateway addresses

|  |
| --- |
| HQ (config)#ip dhcp excluded-address 192.168.10.1  HQ (config)#ip dhcp excluded-address 192.168.10.33  HQ (config)#ip dhcp excluded-address 192.168.10.65  HQ (config)#ip dhcp excluded-address 192.168.10.97 |

Explanation of commands:

HQ (config)#ip dhcp pool vlan10 #creating pool and naming

HQ (dhcp-config)#network 192.168.10.0 255.255.255.224 #giving a network address with subnetmask

HQ(dhcp-config)#default-router 192.168.10.1 #giving IP address of gateway

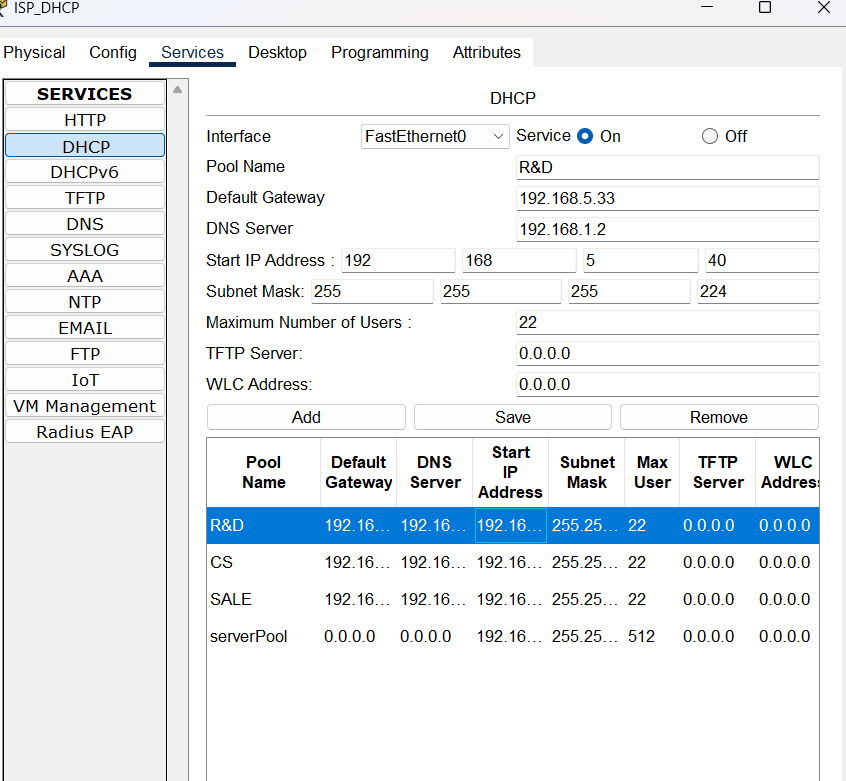
HQ (dhcp-config)#dns-server 192.168.10.8 #giving DNS server IP address

HQ (config)#ip dhcp excluded-address 192.168.10.1 #Excluding IP address so not to give that IP any device

**SUM:**

* **Before everything we created pool, which has a scale of IP address in it, to share IP address to other devices.**
* **The reason not to include gateway IP of each vlans is that we don’t want to give that IP address other devices and make gateway IP stabile to block some conflict.**
* **And also give IP address of DNS to can reverse IP address when it is needed**

**DHCP SERVER SIDE (ISP\_DHCP)**

****

|  |
| --- |
| branch |
| BRANCH#conf t  BRANCH(config)#int g0/0.10  BRANCH(config-subif)#ip helper-address 192.168.1.2  BRANCH(config-subif)#int g0/0.20  BRANCH(config-subif)#ip helper-address 192.168.1.2  BRANCH(config-subif)#int g0/0.30  BRANCH(config-subif)#ip helper-address 192.168.1.2 |

***Meaning of them?***BRANCH(config)#int g0/0.10

**#**Going to sun-interface

BRANCH(config-subif)#ip helper-address 192.168.1.2

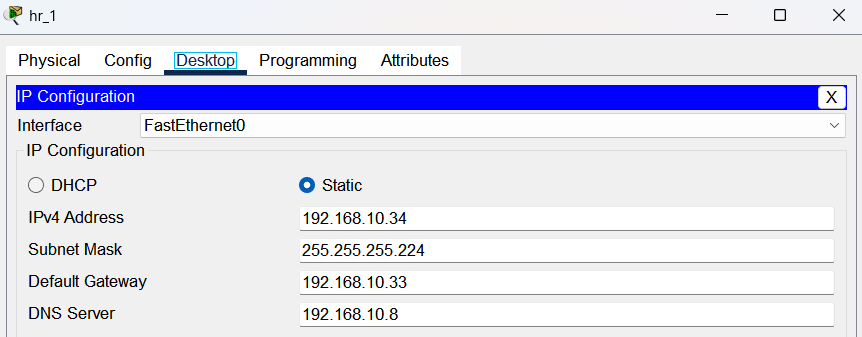
**#**Introducing IP address of DHCP as a helper because We decided use DHCP to get IP address not to give IP address manually

***Sum:***

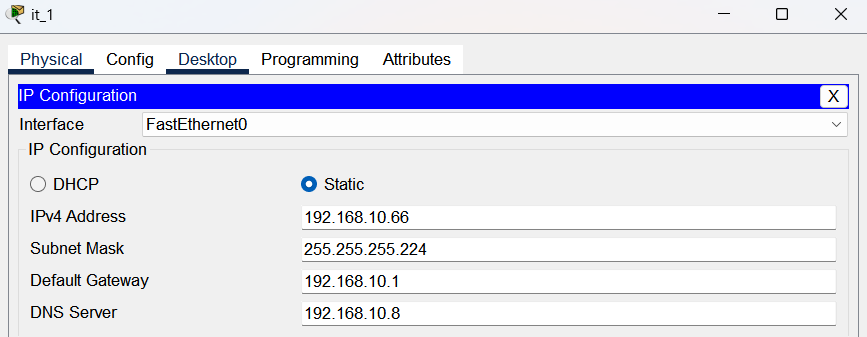
* ***we give each sub-interface ip of helper-address to get IP address and can communicate with other devices***

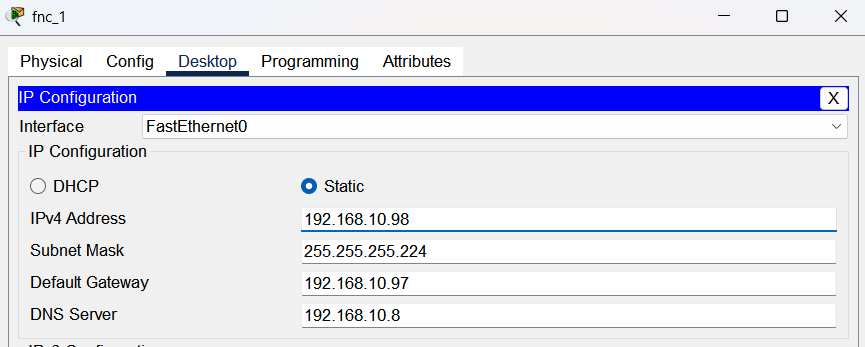
***MANUALLY***

I am doing some part manually due to being able to write restriction like some ACL list or rules and also giving each servers an IP address manually because that pc’s should be able to reach them every time and prevent some conflict. Thus, servers should have stable IP addresses.

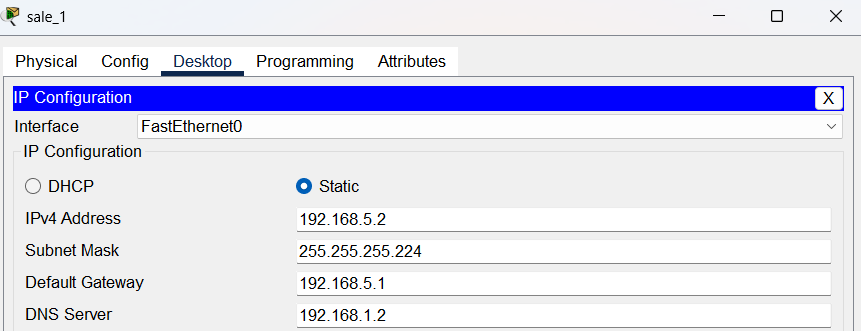
**Manually Configured PC’S**  
**hr\_1:**  


**it\_1:**

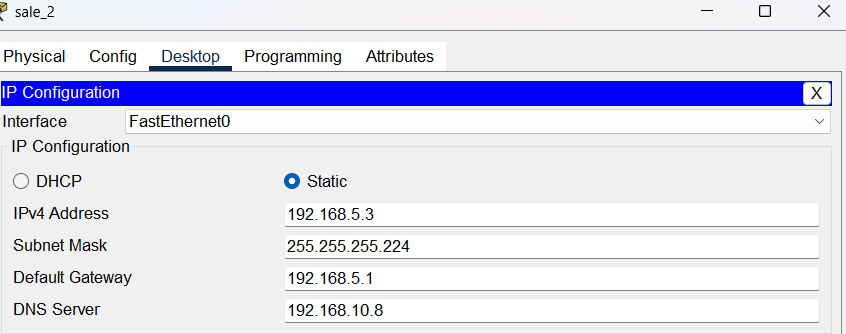


**Fnc\_1:**  
****

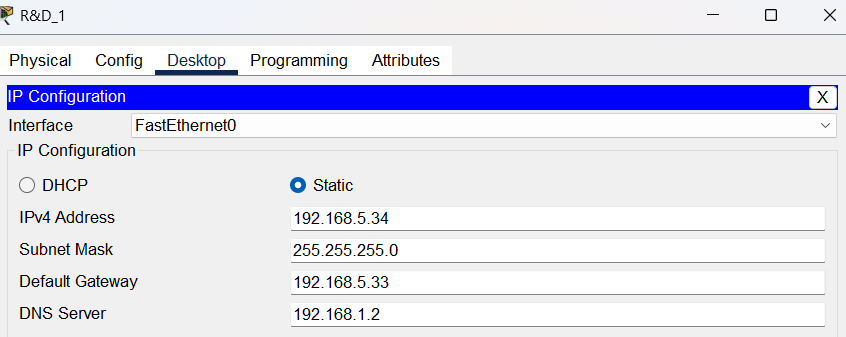
**Sale\_1:**

****

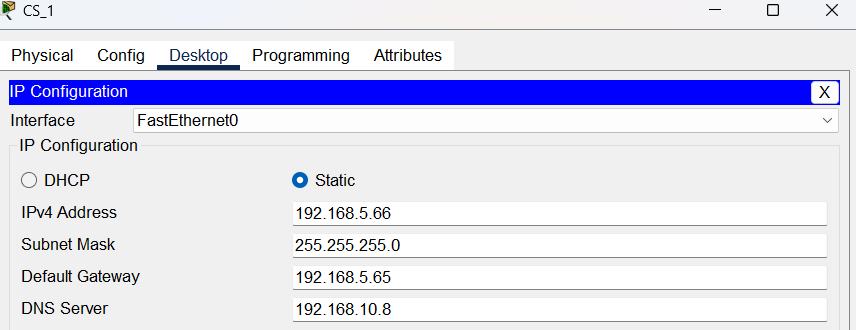
**Sale\_2:**

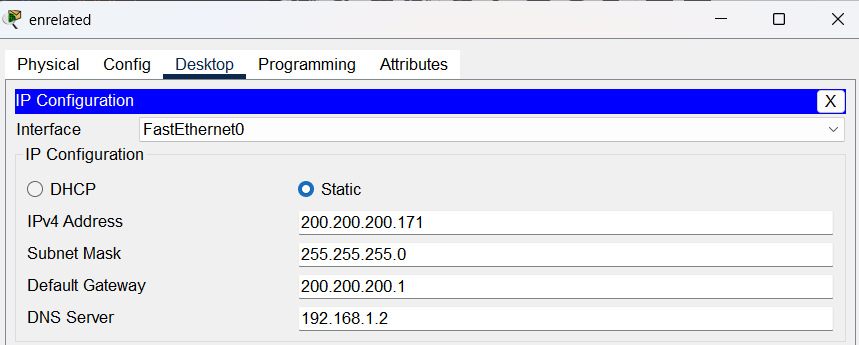
****

**R&D:**

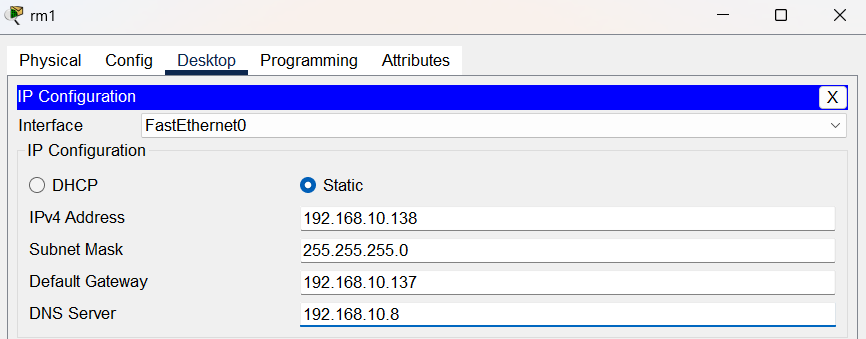
****

**CS:**

****

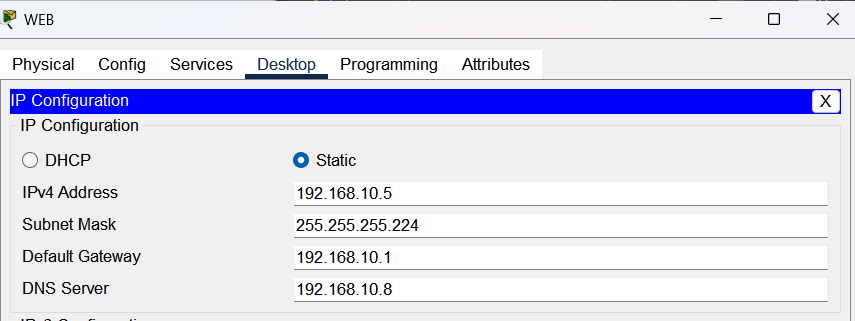
**Unrelated:  
**

**Rm1(remote\_user):**

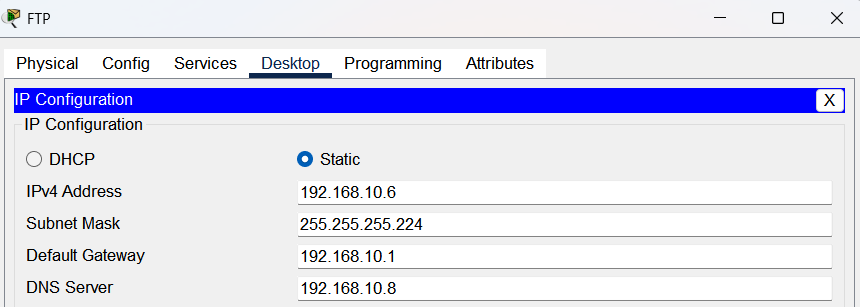
****

**Manually Configured Servers**

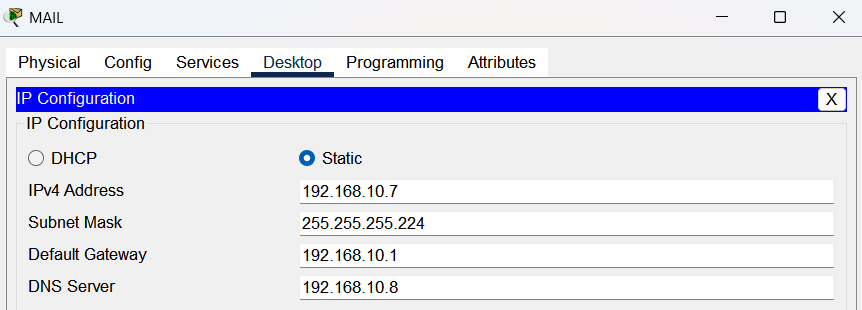
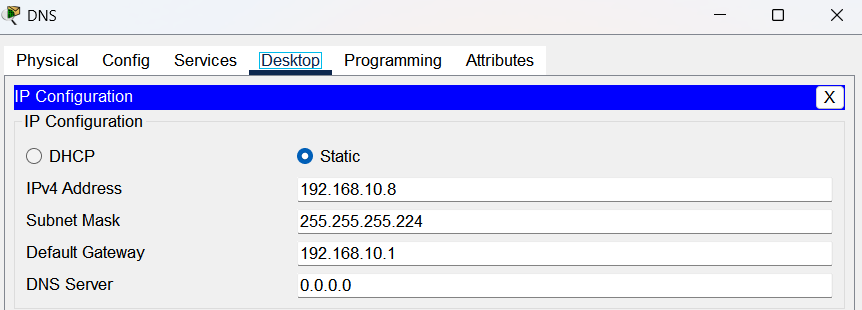
**WEB\_SERVER:**



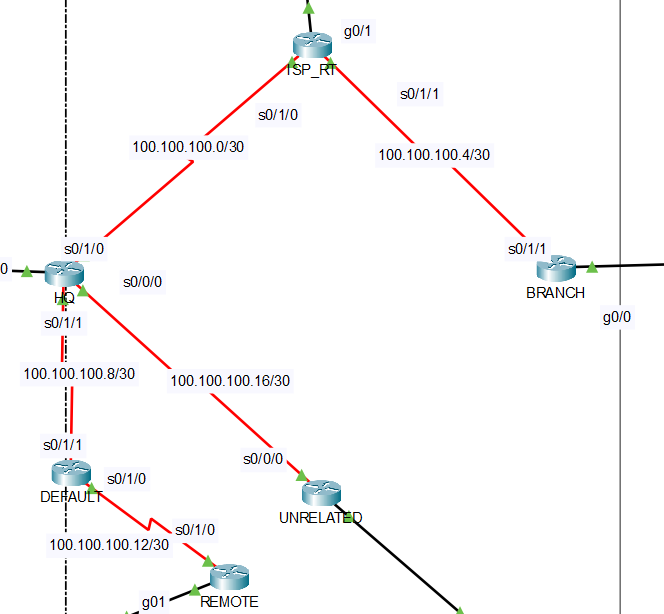
**FTP\_SERVER:**



**MAIL\_SERVER:**

  
  
**DNS\_SERVER:**  


**MANUALLY CONFIGURED ROUTER INTERFACES**



**-Explanation of command above:**

HQ>en #to connect device

HQ#conf t #make configuration

HQ(config)#int s0/1/0 #go to interface

HQ(config-if)#ip address 100.100.100.1 255.255.255.252 #give IP address that interface

HQ(config-if)#no sh #to open the port

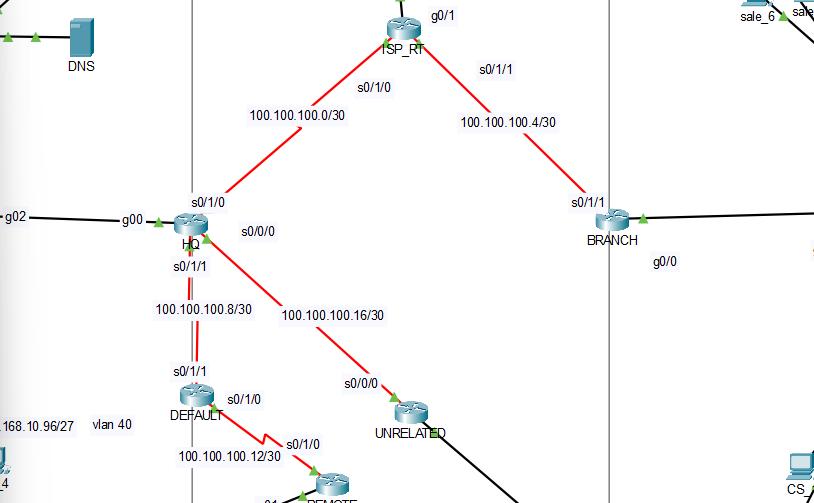
\*It is same for each interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HQ\_s0/1/0 | ISP\_router\_s0/1/0 | ISP\_router\_s0/1/1 | branch\_s0/1/1 | ISP\_router\_g0/1 |
| HQ>en  HQ#conf t  HQ(config)#int s0/1/0  HQ(config-if)#ip address 100.100.100.1 255.255.255.252  HQ(config-if)#no sh | ISP\_RT#conf t  ISP\_RT(config)#int s0/1/0  ISP\_RT(config-if)#ip address 100.100.100.2 255.255.255.252  ISP\_RT(config-if)#no sh | ISP\_RT#conf t  ISP\_RT(config)#int s0/1/1  ISP\_RT(config-if)#ip address 100.100.100.5 255.255.255.252  ISP\_RT(config-if)#no sh | BRANCH#conf t  BRANCH (config)#int s0/1/1  BRANCH (config-if)#ip address 100.100.100.6 255.255.255.252  BRANCH (config-if)#no sh | ISP\_RT(config)#int g0/1  ISP\_RT(config-if)#ip address 192.168.1.1 255.255.255.252  ISP\_RT(config-if)#no sh |

|  |  |  |  |
| --- | --- | --- | --- |
| HQ\_s0/0/0 | HQ\_s0/1/1 | Remote\_s0/1/0 | Remote\_g0/1 |
| HQ>en  HQ#conf t  HQ(config)#int s0/0/0  HQ(config-if)#ip address 100.100.100.17 255.255.255.252  HQ(config-if)#no sh | HQ>en  HQ#conf t  HQ(config)#int s0/1/1  HQ(config-if)#ip address 100.100.100.9 255.255.255.252  HQ(config-if)#no sh | REMOTE#conf t  REMOTE (config)#int s0/1/0  REMOTE (config-if)#ip address 100.100.100.14 255.255.255.252  REMOTE (config-if)#no sh | REMOTE#conf t  REMOTE (config)#int g0/1  REMOTE (config-if)#ip address 192.168.10.137 255.255.255.248  REMOTE (config-if)#no sh |

|  |  |  |  |
| --- | --- | --- | --- |
| Default\_s0/1/0 | Default\_s0/1/1 | Unrelated\_s0/0/0 | Unrelated\_g0/1 |
| DEFAULT>en  DEFAULT #conf t  DEFAULT (config)#int s0/1/0  DEFAULT (config-if)#ip address 100.100.100.13 255.255.255.252  DEFAULT (config-if)#no sh | DEFAULT>en  DEFAULT #conf t  DEFAULT (config)#int s0/1/1  DEFAULT (config-if)#ip address 100.100.100.10  255.255.255.252  DEFAULT (config-if)#no sh | UNRELATED#conf t  UNRELATED (config)#int s0/0/0  UNRELATED (config-if)#ip address 100.100.100.18 255.255.255.252  UNRELATED (config-if)#no sh | UNRELATED#conf t  UNRELATED (config)#int g0/1  UNRELATED (config-if)#ip address 200.200.200.1 255.255.255.252  UNRELATED (config-if)#no sh |

**EIGRP**

****

ROUTING(EIGRP)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| branch | Isp\_rt | hq | remote | default | unrelated |
| BRANCH>en  BRANCH#conf t  BRANCH (config)#router eigrp 10  BRANCH (config-router)#eigrp router-id 1.1.1.1  BRANCH (config-router)#network 192.168.5.0 0.0.0.31  BRANCH (config-router)#network 192.168.5.32 0.0.0.31  BRANCH (config-router)#network 192.168.5.64 0.0.0.31  BRANCH (config-router)#network 100.100.100.4 0.0.0.3  BRANCH (config-router)#no auto-summary | ISP\_RT>en  ISP\_RT#conf t  ISP\_RT(config)#route eigrp 10  ISP\_RT(config-router)#eigrp router-id 2.2.2.2  ISP\_RT(config-router)#network 0.0.0.0 0.0.0.0  (internet) | HQ>en  HQ#conf t  HQ(config)#route eigrp 10  HQ(config-router)#eigrp router-id 3.3.3.3  HQ(config-router)#network 192.168.10.0 0.0.0.31  HQ(config-router)#network 192.168.10.32 0.0.0.31  HQ(config-router)#network 192.168.10.64 0.0.0.31  HQ(config-router)#network 192.168.10.96 0.0.0.31  HQ(config-router)#network 100.100.100.0 0.0.0.3  HQ(config-router)#network 192.168.10.128 0.0.0.127(I added that to make other device can make ssh connection)  HQ(config-router)#no auto-summary | REMOTE#conf t  REMOTE(config)#router eigrp 10  REMOTE(config-router)#eigrp router-id 4.4.4.4  REMOTE(config-router)#network 0.0.0.0 0.0.0.0  (internet) | DEFAULT#conf t  DEFAULT(config)#router eigrp 10  DEFAULT(config-router)#eigrp router-id 5.5.5.5  DEFAULT(config-router)#network 0.0.0.0 0.0.0.0  (internet) | DEFAULT#conf t  DEFAULT(config)#router eigrp 10  DEFAULT(config-router)#eigrp router-id 5.5.5.5  DEFAULT(config-router)#network 0.0.0.0 0.0.0.0  (internet) |

**What are the routing commands?**

BRANCH>en #to access device

BRANCH#conf t #to set some sonfiguration t is terminal so “configure terminal”

BRANCH (config)#router eigrp 10 #determinin type of routing protocol and Autonomous System (AS) number

BRANCH (config-router)#eigrp router-id 1.1.1.1 #giving a router-id

BRANCH (config-router)#network 192.168.5.0 0.0.0.31 # This tells the router to enable EIGRP on interfaces matching 192.168.5.0/27

BRANCH (config-router)#network 192.168.5.32 0.0.0.31 #Same above it also says there is a network to neighbour router

BRANCH (config-router)#network 192.168.5.64 0.0.0.31 #Making some advertisements to share that info

BRANCH (config-router)#network 100.100.100.4 0.0.0.3 #same as the last of 3

BRANCH (config-router)#no auto-summary #prevent sytem from rounding curtain network

ISP\_RT(config)#route eigrp 10 # specify type of routing algorithm and AS number

ISP\_RT(config-router)#eigrp router-id 2.2.2.2 #as a formality, it just help us what is happing and who is who

ISP\_RT(config-router)#network 0.0.0.0 0.0.0.0 #it says I will sent every message (ads) from everwhere (so assuming like it is internet)

(internet)

**Sum:**

* **To begin with, we specifying router protocol and autonomous system number, If we haven’t used same AS number, then they wouldn’t be able to make a connection. Therefore. I supposed there is just one AS and it is 10. Then I remarked every network in interfaces on each router (I do that for Branch and HQ). And also, I assumed there was internet between Branch and HQ so I remarked all interface network as 0.0.0.0 and wildcard mask 0.0.0.0 so that mains route every network.**

**NAT CONFIGURATIONS**

**HQ\_SIDE**

|  |  |
| --- | --- |
| HQ#conf t  HQ(config)#int g0/0.10  HQ(config-subif)#ip nat inside  HQ(config-subif)#int g0/0.20  HQ(config-subif)#ip nat inside  HQ(config-subif)#int g0/0.30  HQ(config-subif)#ip nat inside  HQ(config-subif)#int g0/0.40  HQ(config-subif)#ip nat inside  HQ(config-subif)#int s0/1/0  HQ(config-if)#ip nat outside  HQ(config-if)#ex  HQ(config)#access-list 110 deny ip 192.168.10.0 0.0.0.127 192.168.5.0 0.0.0.127  HQ(config)#access-list 110 permit ip 192.168.10.0 0.0.0.127 any  HQ(config)#ip nat pool NATPOOL 1.1.1.5 1.1.1.13 netmask 255.255.255.0  HQ(config)#ip nat inside source list 110 pool NATPOOL |  |

To show that it works or not?

|  |  |
| --- | --- |
| First ping from hr\_1 to sale\_1    Second ping from hr\_1 to ISP\_DHCP | After first ping we cannot see any translation because we denied packets that go to 192.168.5.0 from being NAT’ed. We can see via second ping, the IP translation and of course because of NAT we send ping but the ping cannot return due to IP changing |

HQ#conf t #to configure terminal of HQ

HQ(config)#int g0/0.10 #going to sub interface of g0/0.10

HQ(config-subif)#ip nat inside #indicating that device inside will be NAT’ed

HQ(config-subif)#int g0/0.20 # going to sub-interface of g0/0.20

HQ(config-subif)#ip nat inside # saying: launch NAT inside

HQ(config-subif)#int g0/0.30 # going to sub-interface of g0/0.30

HQ(config-subif)#ip nat inside # writing this, make IP of devices in that interface network change

HQ(config-subif)#int g0/0.40 # going to sub-interface of g0/0.40

HQ(config-subif)#ip nat inside #Inside will be NAT’ed

HQ(config-subif)#int s0/1/0 # to go interface s0/1/0

HQ(config-if)#ip nat outside #That means the NAT’ted IP go out via that interface

HQ(config-if)#ex #exit from conf\_if

----Creating an acl for NAT----

HQ(config)#access-list 110 deny ip 192.168.10.0 0.0.0.127 192.168.5.0 0.0.0.127

**#**create an acl named 110, deny a packet from HQ network to branch

**\***(we will be using this for NAT ,so actual meaning of it that deny any packet from HQ to be NAT so like it is not allowed to change its IP)

HQ(config)#access-list 110 permit ip 192.168.10.0 0.0.0.127 any

**#**give direction that permit a packet from HQ network to everyone to same acl number(110)

HQ(config)#ip nat pool NATPOOL 1.1.1.5 1.1.1.13 netmask 255.255.255.0

**#**creating NAT pool named NATPOOL   
**#**get an IP range( a range from 5-13) from ISP  
**#**we are using that when we send packet that packet will go through that interface IP.

HQ(config)#ip nat inside source list 110 pool NATPOOL

**#**We are saying that use acl 110 and when you translate ip give an IP them from pool

(we could do with interface and PAT system like I made it for BRANCH, I ended up with NATPOOL for scalability)

**BRANCH \_SİDE NAT**

|  |  |
| --- | --- |
| BRANCH#conf t  BRANCH(config-if)#int g0/0.10  BRANCH(config-subif)#ip nat inside  BRANCH(config-subif)#int g0/0.20  BRANCH(config-subif)#ip nat inside  BRANCH(config-subif)#int g0/0.30  BRANCH(config-subif)#ip nat inside  BRANCH(config-if)#int s0/1/1  BRANCH(config-if)#ip nat outside  BRANCH(config-if)#ex  BRANCH(config)#access-list 110 deny ip 192.168.5.0 0.0.0.127 192.168.10.0 0.0.0.127  BRANCH(config)#access-list 110 permit ip 192.168.5.0 0.0.0.127 any  BRANCH(config)#ip nat pool NATPOOL 1.1.1.14 1.1.1.23 netmask 255.255.255.0  BRANCH(config)#ip nat inside source list 110 interface s0/1/1 overload |  |

**To show that it works or not?**

|  |  |
| --- | --- |
| First ping🡺 from cs\_1 to it\_3    Second ping🡺 from cs\_1 to unrelated | There was made overload PAT (Port Address Translation) , we used same inside global as our s0/1/1 interface IP Also we couldn’t see first ping NAT ,because we write ACL about avoiding NAT when packet goes to HQ |

----Creating an acl for NAT----

BRANCH(config)#access-list 110 deny ip 192.168.5.0 0.0.0.127 192.168.10.0 0.0.0.127

**#**create an acl named 110, deny a packet from BRANCH network to HQ

**\***(we will be using this for NAT ,so actual meaning of it that deny any packet from BRANCH to be NAT so like it is not allowed to change its IP)

BRANCH(config)#access-list 110 deny ip 192.168.5.0 0.0.0.127 host 192.168.1.2

**#**We prevent BRANCH network translation for ISP\_DHCP to get their IP addresses without any problem.

BRANCH(config)#access-list 110 permit ip 192.168.5.0 0.0.0.127 any

**#**give direction that permit a packet from BRANCH network to everyone to same acl number(110)

BRANCH(config)#ip nat pool NATPOOL 1.1.1.14 1.1.1.23 netmask 255.255.255.0

**#**creating NAT pool named NATPOOL   
**#**get an IP range( a range from 14-23) from ISP  
**#**we are using that when we send packet that packet will go through that interface IP.

BRANCH(config)#ip nat inside source list 110 interface s0/1/1 overload

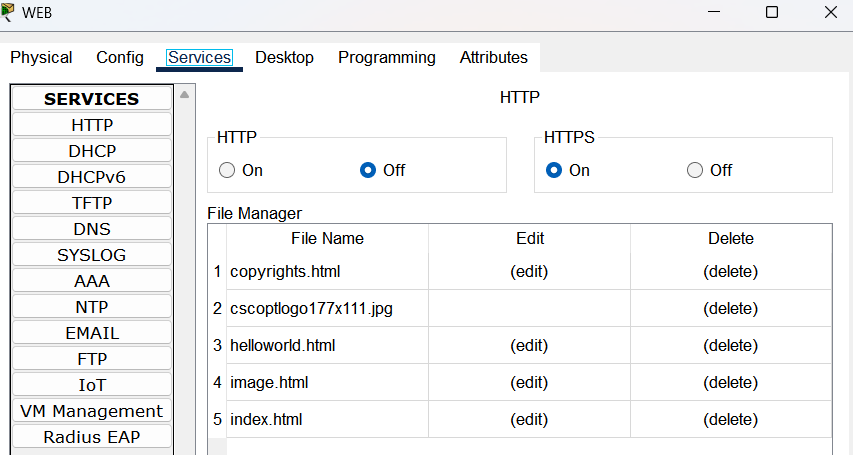
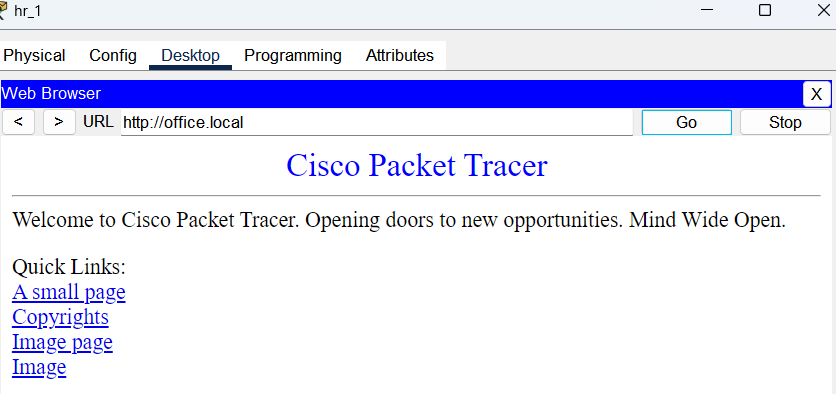
**#**We are saying that use acl 110 and when you translate ip give an IP them interface and use PAT, I mean make sending packet on ports and use s0/0/0 interface (I settle on that one.)

**Sum:**

* **First of all, we indicate each interface which will be inside and which will be outside, then we create an ACL, actually that part crucial because that when we send our message from HQ and NAT’ed that packet IP then when we face another NAT’ted network we couldn’t be communicate. Because router in between them won’t be able to understand which IP is belong to whom. Prevent this we use NAT. And for sure, We can indicate when IP in HQ will be NAT’ed wit help of the ACL. When we write ACL deny that will be meaning of deny that IP address from being Nat’ed to specific network then, the Packet will go that network without any change on its IP address. I permit other network to be denied so when we go internet no one can see that packet actual Ip address. Also, about acl, in first rule I wrote deny translate ip from hq then for second, giving permission to any address. There is no conflict. What we write first it is accepted by ACL. ACL apply the rules row by row below. To put in a nut shell, we made that conf both BRANCH and HQ side to be able to communicate and not to show real IP to internet.**

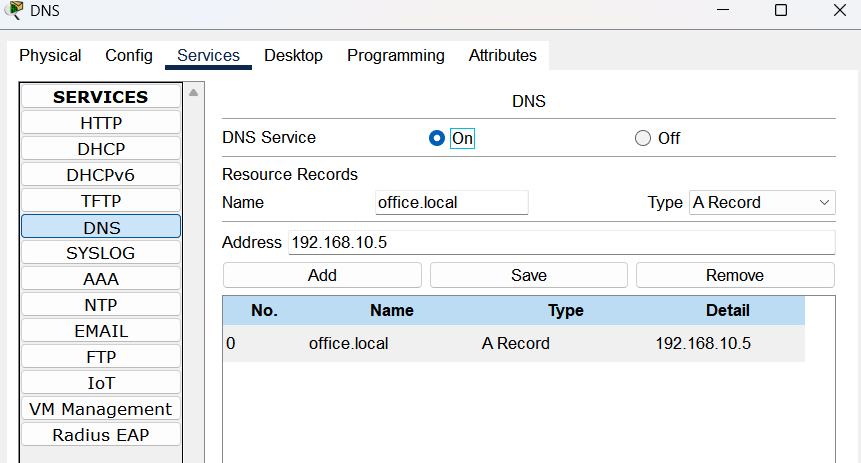
**SERVERS**

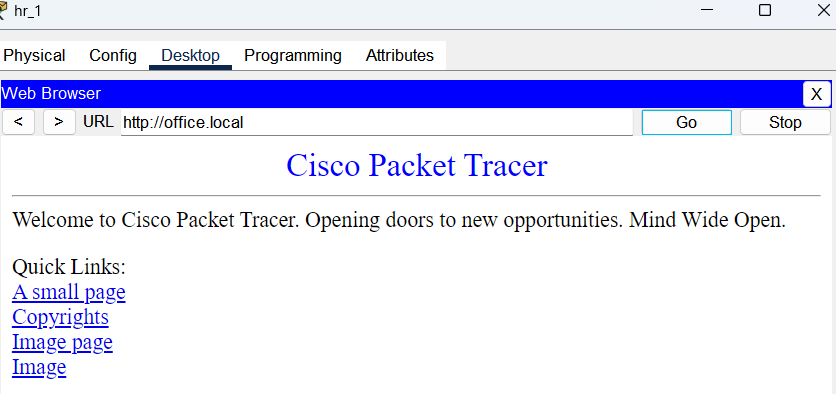
**1.WEB**



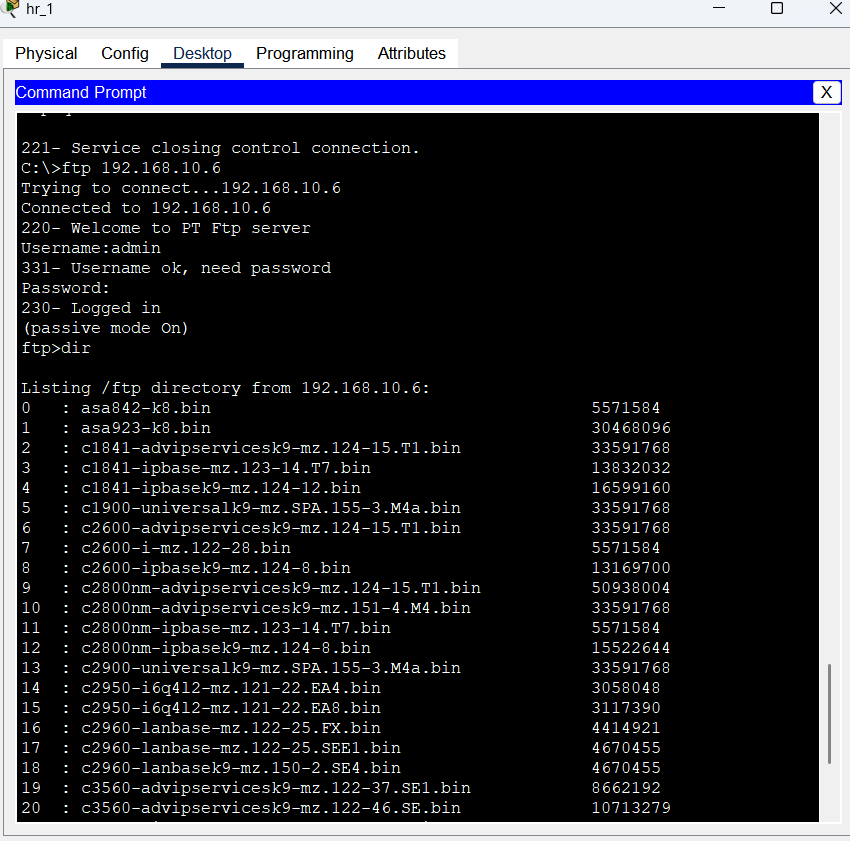
-We turned HTTP off because it includes lots of vulnerability.

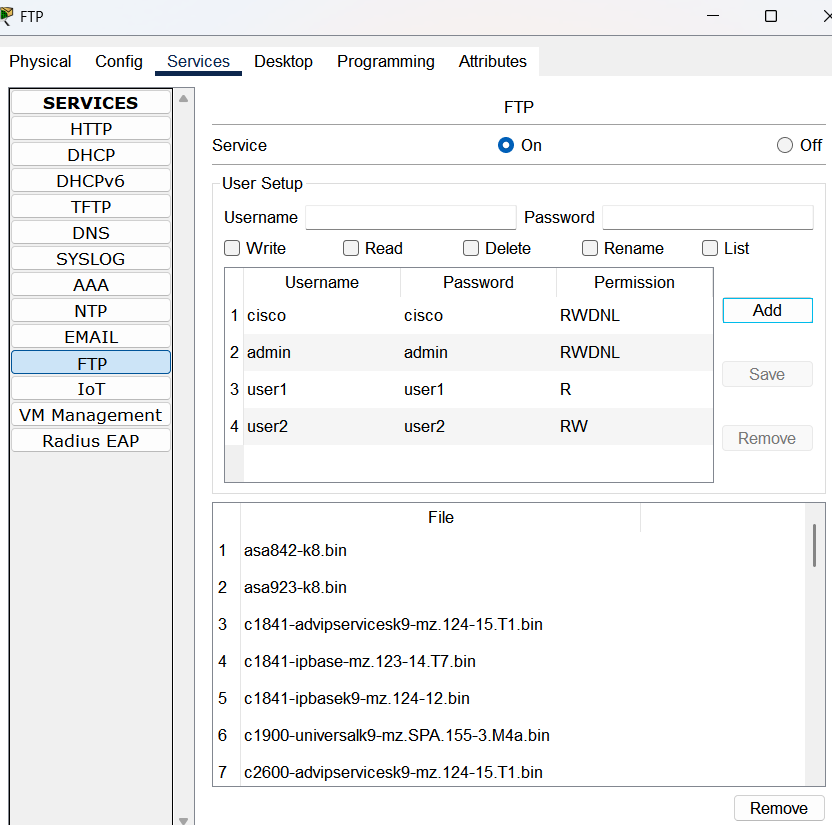
-When someone connect web server they get a page like beside.

2.**DNS**



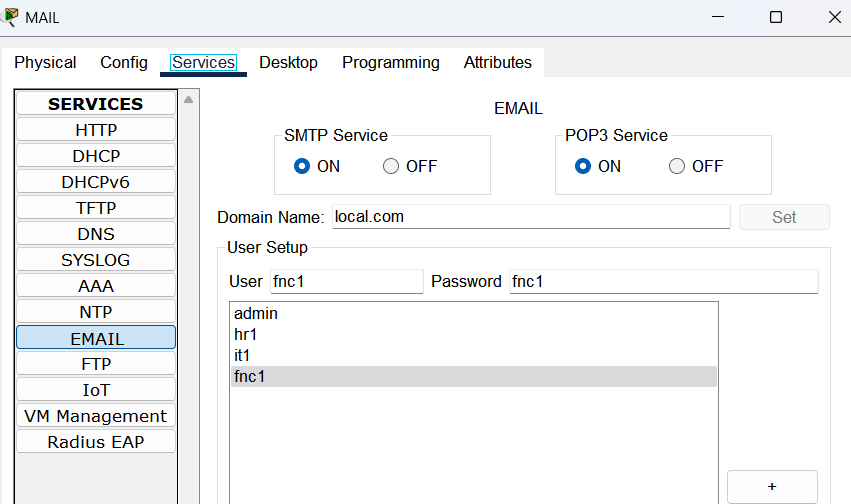
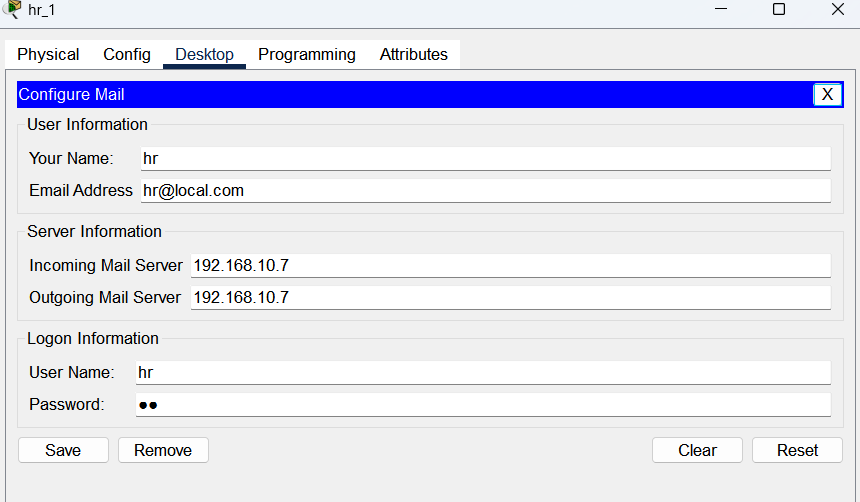
-Image right side indicates that when we write domain name a browser and when it returns an answer that means that domain name was resolved by a DNS.

**3.FTTP**

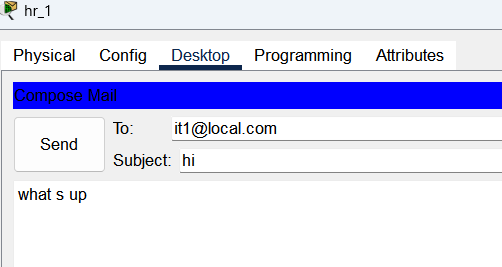
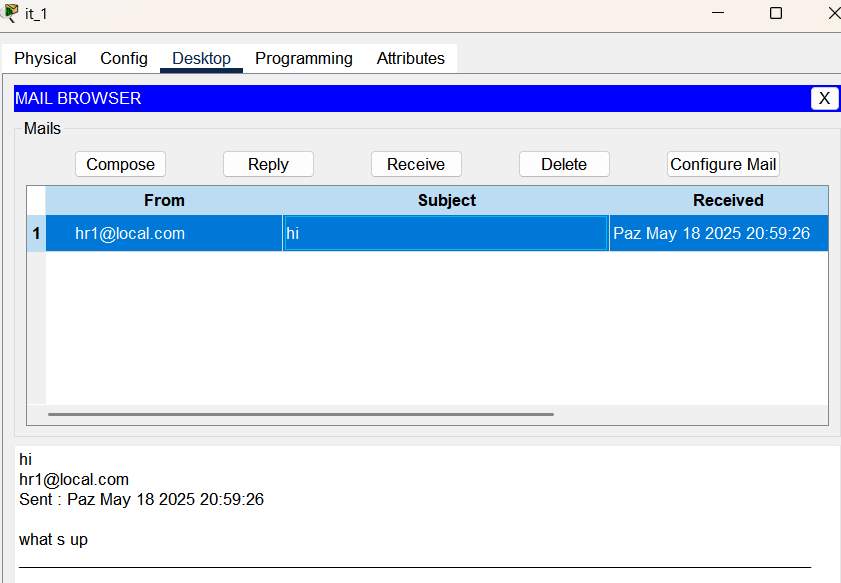


The right side of page there is an image of ftp connection and its result

**4. MAİL**



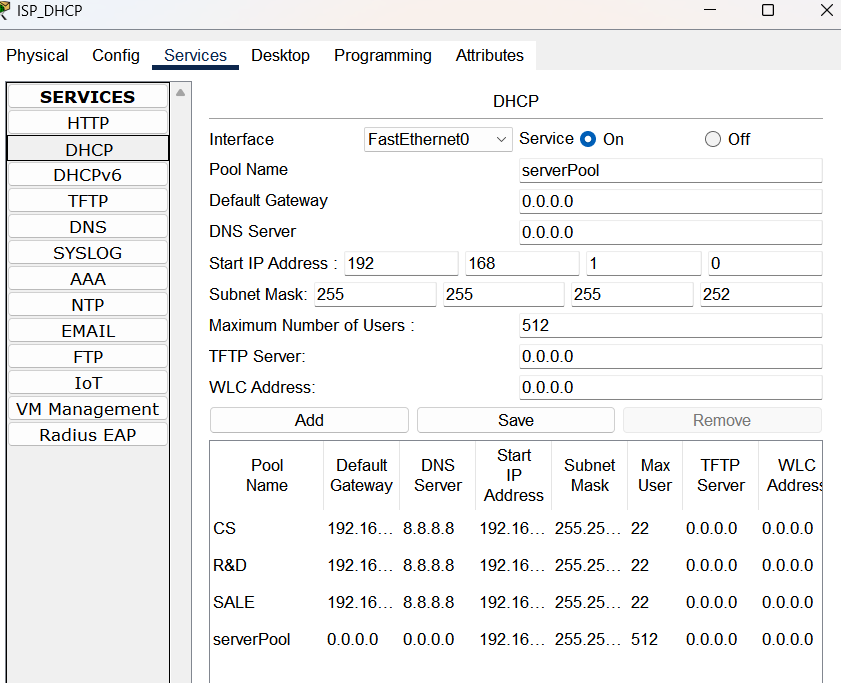
-For mail server, We configured the sytem like givina a domain name and add users.  
-Then, created mail address for most of PC’s



-The last step was sending mail and approve whether it was sent

**5. DHCP**

**Server**

****

**SECURITY OF THE NETWORK**

**ACL (ACCESS CONTROL LIST)**

**HQ**

|  |  |
| --- | --- |
| ***List 110*** | Explanation of commands applied for NAT |
| HQ#conf t  HQ(config)#ip access-list extended 110  HQ(config-ext-nacl)#10 deny ip 192.168.10.0 0.0.0.127 192.168.5.0 0.0.0.127  HQ(config-ext-nacl)#20 permit ip 192.168.10.0 0.0.0.127 any  -- applied to --  HQ(config)#ip nat inside source list 110 pool NATPOOL | ip access-list extended 110 = we are creating a list named 110 and it is extended so we can make restriction on ports as well  rule 10 – deny packet from IP translation of HQ Network 10.0/25 when packet goes to BRANCH Network  rule20 –give permission to change ip address when a HQ packet go anywhere |

|  |  |
| --- | --- |
| ***List 120*** | Explanation of commands \* Sub-interface |
| HQ#conf t  HQ(config)#ip access-list extended 120  HQ(config-ext-nacl)#10 permit tcp host 192.168.10.98 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#20 permit tcp host 192.168.10.34 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#30 permit tcp host 192.168.10.66 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#40 permit tcp host 192.168.10.138 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#50 permit tcp host 192.168.5.34 host 192.168.10.6 eq ftp  HQ(config)#60 permit tcp host 192.168.5.2 host 192.168.10.6 eq ftp  HQ(config)#70 deny tcp any host 192.168.10.6 eq ftp  HQ(config)#80 permit icmp any any  HQ(config)#90 permit ip any any  --applied to --  HQ(config-ext-nacl)#ex  HQ(config)#int g0/0.10  HQ(config-subif)#ip ac  HQ(config-subif)#ip access-group 120 in  HQ(config-subif)#int g0/0.20  HQ(config-subif)#ip access-group 120 in  HQ(config-subif)#int g0/0.30  HQ(config-subif)#ip access-group 120 in  HQ(config-subif)#int g0/0.40  HQ(config-subif)#ip access-group 120 in | IP access-list extended 120 = we are creating a list named 120 and it is extended so we can make restriction on ports as well  Rule 10, just 98 PC can reach FTP server  Rule 20, just 10.34 PC can reach FTP server  Rule 30, just 66 PC can reach FTP server  Rule 40, just 138 PC can reach FTP server  Rule 50, just 5.34 PC can reach FTP server  Rule 60, just 5.2 PC can reach FTP server  Rule 70, no one can reach FTP server  Rule 80, everyone can ping each other  Rule 90, any IP can reach any IP  \*Acl was implemented to sub-interfaces  ip access-group 120 in  **#**that means put list 120 the inbound of interface so checking the packet come from inside not going out |

|  |  |
| --- | --- |
| ***List 130*** | Explanation of the commands*\*ISP\_DHCP serve DNS future* |
| HQ#conf t  HQ(config)#ip access-list extended 130  HQ(config-ext-nacl)#10 permit tcp host 192.168.5.2 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#20 permit tcp host 192.168.5.34 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#30 permit tcp host 192.168.5.66 host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#40 deny tcp any host 192.168.10.6 eq ftp  HQ(config-ext-nacl)#50 permit tcp 192.168.5.0 0.0.0.255 host 192.168.10.7 eq smtp  HQ(config-ext-nacl)#60 permit udp host 192.168.1.2 any  HQ(config-ext-nacl)#70 permit udp any host 192.168.1.2 eq domain  HQ(config-ext-nacl)#80 permit tcp 192.168.5.0 0.0.0.255 host 192.168.10.5 eq 443  HQ(config-ext-nacl)#90 permit eigrp any any  HQ(config-ext-nacl)#100 deny icmp any 192.168.10.0 0.0.0.31  HQ(config-ext-nacl)#110 permit icmp any any  -applied to—  HQ(config)#int s0/1/0  HQ(config-if)# ip access-group 110 in | IP access-list extended 130 = we are creating a list named 130 and it is extended so we can make restriction on ports as well  Rule 10, just 5.2 PC can reach FTP server via 21 port  Rule 20, just 5.34 PC can reach FTP server via 21 port  Rule 30, just 5.66 PC can reach FTP server via 21 port  Rule 40, block every request for ftp to FTP server via 21 port  Rule 50, full BRANCH Network can send mail to MAIL server via 25 port  Rule 60, ISP\_DHCP can send udp packet everywhere  Rule 70, every packet from 53 port is allowed to reach ISP\_DHCP  Rule 80, only BRANCH Network can reach WEB server via 443 port  Rule 90, permit every eigrp packet (hello packets)  Rule 100, block everone from ping to HQ Network  Rule 110, everyone can ping each other  \*we applied that rules to inbound of interface s0/1/0 |

|  |  |
| --- | --- |
| ***List 150*** | Explanation of the commands\*s0/0/0 |
| HQ#conf t  HQ(config)#ip access-list extended 150  HQ(config-ext-nacl)#10 permit tcp any host 192.168.10.5 eq 443  HQ(config-ext-nacl)#20 permit tcp host 192.168.10.5 any eq 443  HQ(config-ext-nacl)#30 deny ip 192.168.10.0 0.0.0.255 any  HQ(config-ext-nacl)#40 deny ip any 192.168.10.0 0.0.0.255  HQ(config-ext-nacl)#50 permit ip any any  -applied to—  HQ(config)#int s0/0/0  HQ(config-if)# ip access-group 150 in | IP access-list extended 150 = we are creating a list named 150 and it is extended so we can make restriction on ports as well  Rule 10, everyone can reach WEB server via 443 port  Rule 20, WEB server can reach everyone via 443 port  Rule 30, no one in HQ Network can reach other PC’s IP  Rule 40, everyone was blocked to reach HQ Network  Rule 50, everyone can ping each other  \*we applied that rules to inbound of interface s0/0/0 |

|  |  |
| --- | --- |
| ***List 170*** | Explanation of the commands *\*used for VPN* |
| HQ#conf t  HQ(config)#ip access-list extended 170  HQ(config-ext-nacl)#10 permit ip 192.168.10.0 0.0.0.255 192.168.10.136 0.0.0.7  --applied to—  HQ(config-crypto-map)#match address 170  \*\* crypto map VPN\_MAP  \*\*s0/1/1 | IP access-list extended 150 = we are creating a list named 150 and it is extended so we can make restriction  Rule 10, make encryption on packet from HQ Network to REMOTE Network  \*That rule is applied to s0/1/1 for determining our MAP |

**BRANCH**

|  |  |
| --- | --- |
| ***List 110*** | Explanation of commands applied for NAT |
| BRANCH#conf t  BRANCH(config)#ip access-list extended 110  BRANCH(config-ext-nacl)#10 deny ip 192.168.5.0 0.0.0.127 192.168.10.0 0.0.0.127  BRANCH(config-ext-nacl)#20 deny ip 192.168.5.0 0.0.0.127 host 192.168.1.2  BRANCH(config-ext-nacl)#30 permit ip 192.168.5.0 0.0.0.127 any  -- applied to --  BRANCH(config)#ip nat inside source list 110 interface s0/1/1 overload | ip access-list extended 110 = we are creating a list named 110 and it is extended so we can make restriction on ports  rule 10 – deny packet from IP translation of BRANCH Network 5.0/25 when packet goes to HQ Network  rule 20 –deny BRANCH packets IP translation when it goes to ISP\_DHCP (is maden because of that PC’s getting IP address from ISP\_DHCP)  rule 30 –give permission to change ip address when a BRANCH packet go anywhere  \*The rule list applied for NAT in s0/1/1 interface |

|  |  |
| --- | --- |
| ***List 120*** | Explanation of commands applied for NAT |
| BRANCH#conf t  BRANCH(config)#ip access-list extended 120  BRANCH(config-ext-nacl)#10 permit eigrp any any  BRANCH(config-ext-nacl)#20 permit ip 192.168.10.0 0.0.0.255 any  BRANCH(config-ext-nacl)#30 permit ip host 192.168.1.2 any  BRANCH(config-ext-nacl)#40 deny ip any any  -applied to—  BRANCH(config)#int s0/1/1  BRANCH (config-if)# ip access-group 120 in | ip access-list extended 110 = we are creating a list named 110 and it is extended so we can make restriction on ports  rule 10 – every eigrp packet is allowed to go everywhere  rule 20 –HQ Network can reach BRANCH Network  rule 30 –ISP\_DHCP is allowed to connect BRANCH Network  rule 40 – every PC is denied to connect any PC  \*access-group(list 120) is applied on inbound of interface s0/1/0 |

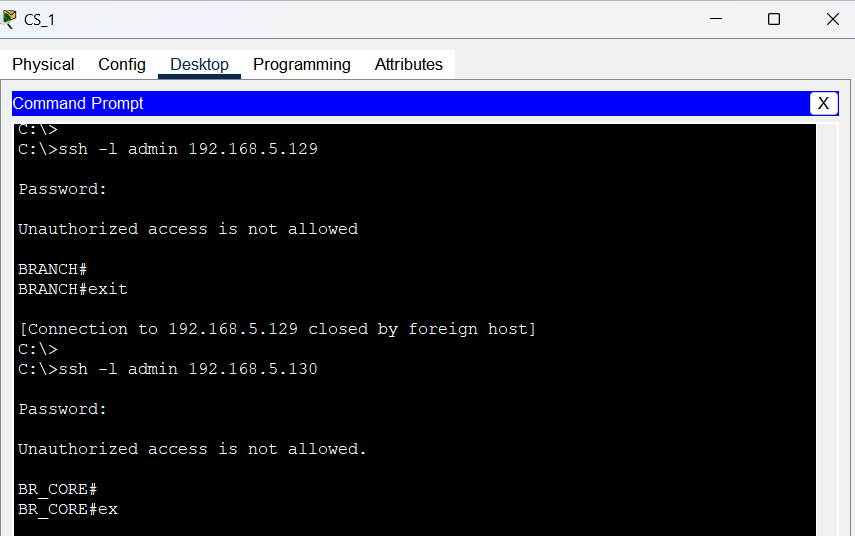
**SUM:**

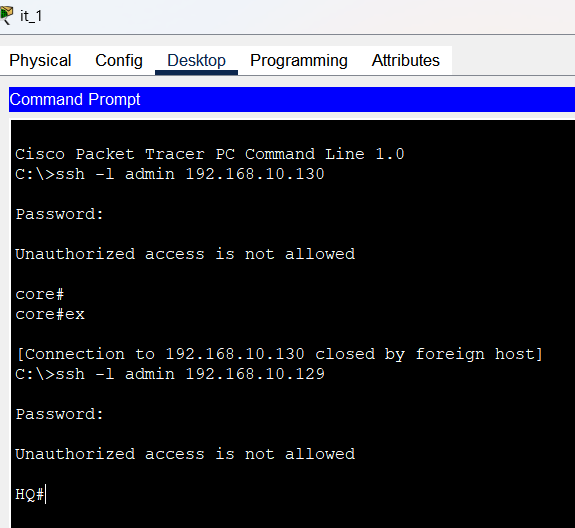
**With those rules we specified that certain IP and port such as 53,443,21,22 only connect certain IP or ports of devices**

**SSH-Enable\_Console\_Password**

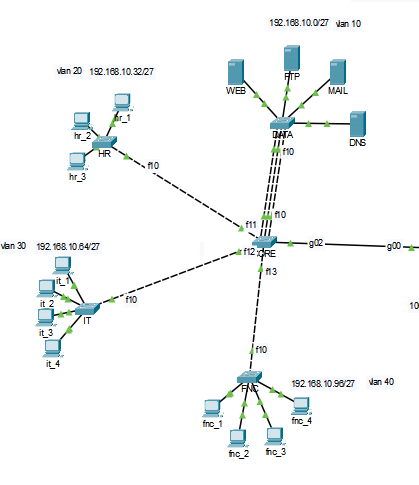
**(Enable\_Console)Password**

|  |  |  |  |
| --- | --- | --- | --- |
| hq | core | branch | br\_core |
| HQ>en  HQ#conf t  HQ(config)#enable secret **hq**  HQ(config)#line console 0  HQ(config-line)#password hq  HQ(config-line)#login  HQ(config-line)#exec-timeout 5 0  HQ(config-line)#logging synchronous  HQ(config-line)#exit  HQ(config)#ip domain-name **office.local**  HQ(config)#crypto key generate rsa  --How many bits in the modulus [512]: 1024  HQ(config)#username admin privilege 15 secret **hq** HQ(config)#line vty 0 4  HQ(config-line)#login local  HQ(config-line)#exec-timeout 5 0  HQ(config-line)#transport input ssh  HQ(config-line)#exit  HQ(config)#security passwords min-length 10  HQ(config)#username user password hq  % Password too short - must be at least 10 characters. Password not configured.  HQ(config)#username user password **hq\_user\_hq**  HQ(config)#login block-for 180 attempts 3 within 60  HQ(config)#banner motd "Unauthorized access is not allowed" | core>en  core#conf t  core(config)#enable secret **core**  core(config)#line console 0  core(config-line)#password **core**  core(config-line)#login  core(config-line)#exec-timeout 5 0  core(config-line)#logging synchronous  core(config)#ip domain-name **office.local**  core(config)#crypto key generate rsa  ---How many bits in the modulus [512]: 1024  core(config)#line vty 0 4  core(config-line)#login local  core(config-line)#exec-timeout 5 0  core(config-line)#transport input ssh  core(config-line)#exit  core(config)#username admin privilege 15 secret **core**  core(config)#banner motd "Unauthorized access is not allowed"  core(config)#login block-for 180 attempts 3 within 60 | BRANCH>en  BRANCH#conf t  BRANCH(config)#security passwords min-length 10  BRANCH(config)#enable secret **branch\_center\_switch**  BRANCH(config)#login block-for 180 attempts 3 within 60  BRANCH(config)#ip domain-name **branch\_affice.local**  BRANCH(config)#crypto key generate rsa  How many bits in the modulus [512]: 1024  BRANCH(config)#username admin privilege 15 secret **branch**  BRANCH(config)#line vty 0 4  BRANCH(config-line)#login local  BRANCH(config-line)#exec-timeout 5 0  BRANCH(config-line)#transport input ssh  BRANCH(config-line)#exit  BRANCH(config)#line console 0  BRANCH(config-line)#password **branch\_center\_switch**  BRANCH(config-line)#login  BRANCH(config-line)#exec-timeout 5 0  BRANCH(config-line)#logging synchronous  BRANCH(config-line)#exit  BRANCH(config)#banner motd "Unauthorized access is not allowed" | BR\_CORE#conf t  BR\_CORE(config)#enable secret **br\_core**  BR\_CORE(config)#username admin privilege 15 secret **br\_core**  BR\_CORE(config)#ip domain-name **branch\_office.local**  BR\_CORE(config)#crypto key generate rsa  ---How many bits in the modulus [512]: 1024  BR\_CORE(config)#line vty 0 4  BR\_CORE(config-line)#login local  BR\_CORE(config-line)#exec-timeout 5 0  BR\_CORE(config-line)#transport input ssh  BR\_CORE(config-line)#exit  BR\_CORE(config)#line console 0  BR\_CORE(config-line)#password br\_core  BR\_CORE(config-line)#exec-timeout 5 0  BR\_CORE(config-line)#logging synchronous  BR\_CORE(config)#banner motd "Unauthorized access is not allowed." |

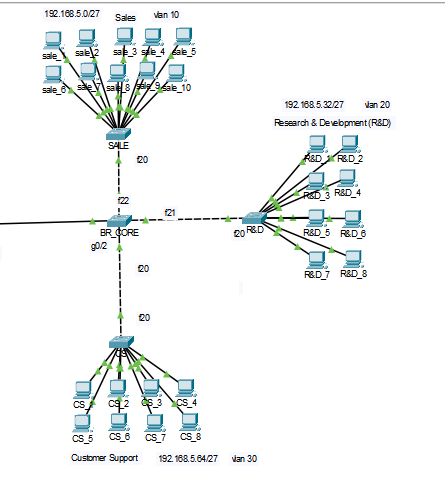
SSH connection to core and hq was succeded via it\_1 SSH connection to br\_core and branch was succeded via cs\_1



\*In the images, it\_1 connected to HQ via ssh connection, at the end it was able to reach. We can see looking at the last line



|  |  |  |  |
| --- | --- | --- | --- |
| data | hr | it | fnc |
| DATA#conf t  DATA(config)#enable secret **data**  DATA(config)#line consol 0  DATA(config-line)#password **data**  DATA(config-line)#login  DATA(config-line)#exec-timeout 5 0  DATA(config-line)#logging synchronous  DATA(config-line)#exit  DATA(config)#service password-encryption | HR#conf t  HR(config)#enable secret **hr**  HR(config)#line console 0  HR(config-line)#password **hr**  HR(config-line)#login  HR(config-line)#exec-timeout 5 0  HR(config-line)#logging synchronous  HR(config-line)#exit  HR(config)#service password-encryption | IT#conf t  IT(config)#enable secret **it**  IT(config)#line console 0  IT(config-line)#password **it**  IT(config-line)#login  IT(config-line)#exec-timeout 5 0  IT(config-line)#logging synchronous  IT(config-line)#exit  IT(config)#service password-encryption | FNC#conf t  FNC(config)#enable secret **fnc**  FNC(config)#line con 0  FNC(config-line)#password **fnc**  FNC(config-line)#login  FNC(config-line)#exec-timeout 5 0  FNC(config-line)#logging synchronous  FNC(config-line)#exit  FNC(config)#service password-encryption |



|  |  |  |
| --- | --- | --- |
| sale | r&d | cs |
| SALE#conf t  SALE(config)#enable secret **sale**  SALE(config)#line con 0  SALE(config-line)#password **sale**  SALE(config-line)#login  SALE(config-line)#exec-timeout 5 0  SALE(config-line)#logging synchronous  SALE(config-line)#exit  SALE(config)#service password-encryption | R&D#conf t  R&D(config)#enable secret **r&d**  R&D(config)#line con 0  R&D(config-line)#password **r&d**  R&D(config-line)#login  R&D(config-line)#exec-timeout 5 0  R&D(config-line)#logging synchronous  R&D(config-line)#exit  R&D(config)#service password-encryption | CS#conf t  CS(config)#enable secret **cs**  CS(config)#line con 0  CS(config-line)#password **cs**  CS(config-line)#login  CS(config-line)#exec-timeout 5 0  CS(config-line)#logging synchronous  CS(config-line)#exit  CS(config)#service password-encryption |

**What are the commands?**

HQ>en #connect Switch

HQ#conf t #makine changes on terminal

HQ(config)#enable secret **hq** #to connect or make SW enable we create password**(hq)** and make it secret so it is not plain text

HQ(config)#line console 0 #to go console 0

HQ(config-line)#password **hq** #to create a password for console 0

HQ(config-line)#login # it means when we see a line is like “password: “, after try login

HQ(config-line)#exec-timeout 5 0 # When there is a inactivity in 5 minutes 0 second then session will be closed

HQ(config-line)#logging synchronous #When I write commands,system messages wont interrupt me

HQ(config-line)#exit #exiting from line console 0

HQ(config)#ip domain-name **office.local** #giving domain name for ssh like creating new envorioument for ssh connection

HQ(config)#crypto key generate rsa #using crypto and make a key in rsa

--How many bits in the modulus [512]: 1024 #that bey will be 1024 bits

HQ(config)#username admin privilege 15 secret **hq**

**#**creating user who has the highest authority( 15) and creating encrypted password(hq)

HQ(config)#line vty 0 4 #at the same that only 5 connection is allowed for ssh connection to HQ under the network (**office.local)**

HQ(config-line)#login local #just that enviourement devices ( **office.local)** is permitted

HQ(config-line)#exec-timeout 5 0 # When there is a inactivity in 5 minutes 0 second then session will be closed

HQ(config-line)#transport input ssh #only permission to ssh no other connection type is allowed such as telnet.

HQ(config-line)#exit # #exiting from line vty

HQ(config)#security passwords min-length 10

**#**it used to specify some rule when password created. It says password should be at least 10 characters

HQ(config)#username user password **hq** # when we create a new user and password. The error raised ad say:

% Password too short - must be at least 10 characters. Password not configured.

HQ(config)#username user password **hq\_user\_hq** #we created that user with longer password

HQ(config)#login block-for 180 attempts 3 within 60

**#**in 180 second inactivity or more then 3 times wrong password attempts it will close session and after 3 incorrect attempts, there will be a 60 second waiting time.

HQ(config)#banner motd "Unauthorized access is not allowed"

**#**Putting a massage of the day , it will be seen after login to warn unauthorized user.

DATA#conf t #configure terminal

DATA(config)#enable secret **data** #when want to connect device there will be encrypted password **(data)** with(password 7)

DATA(config)#line consol 0 #go to console

DATA(config-line)#password **data #set password for console**

DATA(config-line)#login #that password request will be shown on login line

DATA(config-line)#exec-timeout 5 0 #after 5 min inactivity, session will be closed

DATA(config-line)#logging synchronous #not to be interrupted by system logs

DATA(config-line)#exit #login out from line

DATA(config)#service password-encryption **#**make every password in system encrypted

**SUM:**

* **We created user and its password in secure condition such as using encryption and also using security system. We create password for console, enable, ssh connection. For ssh we create a local domain network. Besides, we blocked any telnet which not secure connection.  
  \*Putting duration in system   
  \* Giving specific attempts**

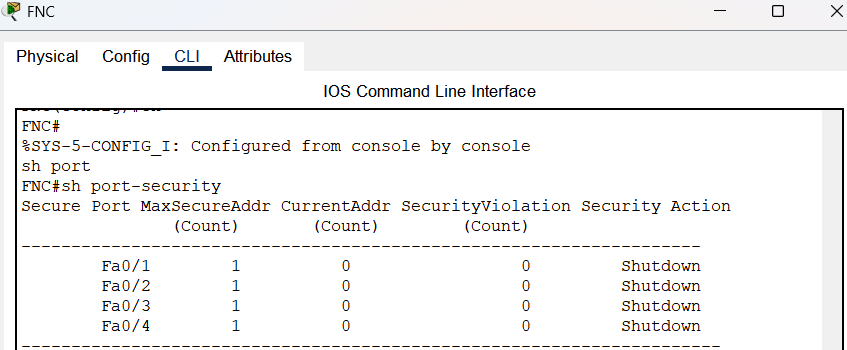
**\*Determining length of password**

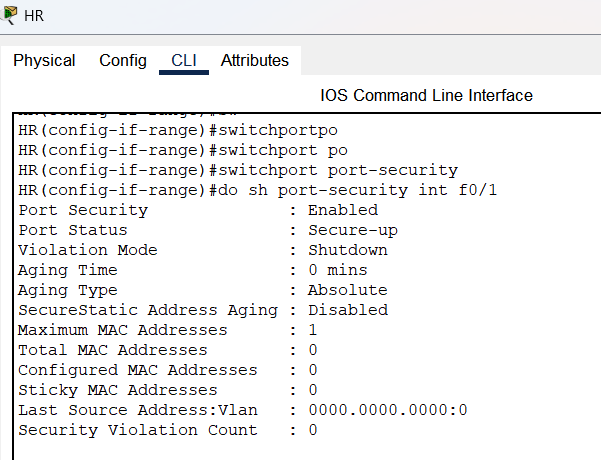
**as not to be exposed easily.**

**MAC-Violation**

|  |  |  |  |
| --- | --- | --- | --- |
| data | hr | it | fnc |
| DATA#conf t  DATA(config)#int range f0/1-4  DATA(config-if-range)#sw port-security  DATA(config-if-range)#switchport port-security mac-address sticky  DATA(config-if-range)#switchport port-security maximum 1  range)#switchport port-security violation shutdown | HR>en  HR#conf t  HR(config)#int range f0/1-3  HR(config-if-range)#sw port-security mac-address sticky  HR(config-if-range)#switchport port-security maximum 1  HR(config-if-range)#switchport port-security violation shutdown | IT#en  IT#conf t  IT(config)#int range f0/1-4  IT(config-if-range)#sw port-security  IT(config-if-range)#switchport port-security mac-address sticky  IT(config-if-range)#switchport port-security maximum 1  IT(config-if-range)#switchport port-security violation shutdown | FNC>en  FNC#conf t  FNC(config)#int range f0/1-4  FNC(config-if-range)#sw port-security  FNC(config-if-range)#sw port-security mac-address sticky  FNC(config-if-range)#sw port-security maximum 1  FNC(config-if-range)#sw port-security violation shutdown |

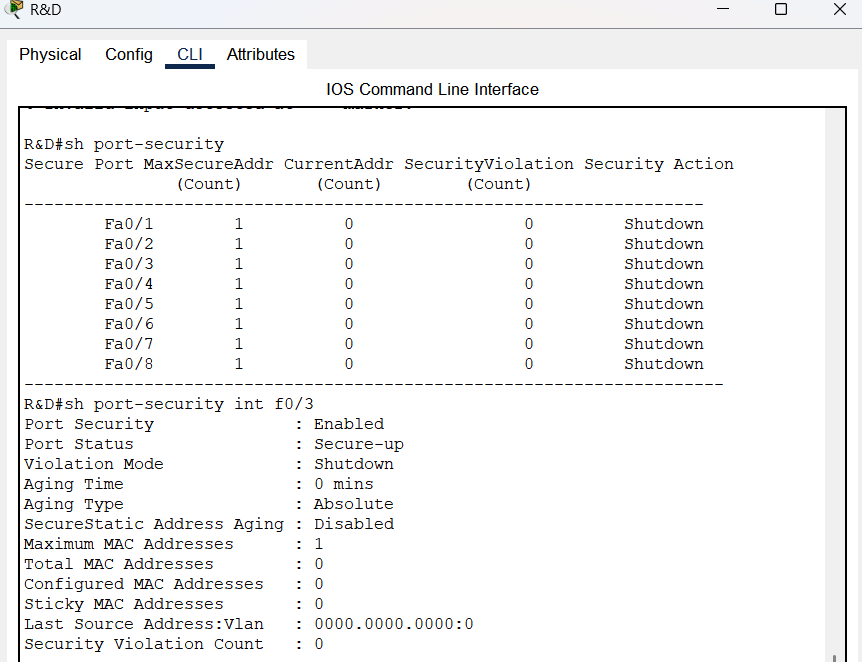
After conf ports turn enabled and they look like:





|  |  |  |
| --- | --- | --- |
| sale | b&d | cs |
| SALE>en  SALE#conf t  SALE(config)#int range f0/1-10  SALE(config-if-range)#sw port-security  SALE(config-if-range)#sw port-security mac-address sticky  SALE(config-if-range)#sw port-security maximum 1  SALE(config-if-range)#sw port-security violation shutdown | R&D#conf t  R&D(config)#int range f0/1-8  R&D(config-if-range)#sw port-security  R&D(config-if-range)#sw port-security mac-address sticky  R&D(config-if-range)#sw port-security maximum 1  R&D(config-if-range)#sw port-security violation shutdown | CS>en  CS#conf t  CS(config)#int range f0/1-8  CS(config-if-range)#sw port-security  CS(config-if-range)#sw port-security mac-address sticky  CS(config-if-range)#sw port-security maximum 1  CS(config-if-range)#sw port-security violation shutdown |

After process:



**What are the commands?**

IT#en #connect IT

IT#conf t #make changes on IT configuration

IT(config)#int range f0/1-4 #going to certain interfaces simultaniously

IT(config-if-range)#switchport port-security mac-address sticky # it enables sticky MAC address learning on a switch port.

IT(config-if-range)#switchport port-security maximum 1 #just one device can connect that interface

IT(config-if-range)#switchport port-security violation shutdown #if there is a violation then shut interface down

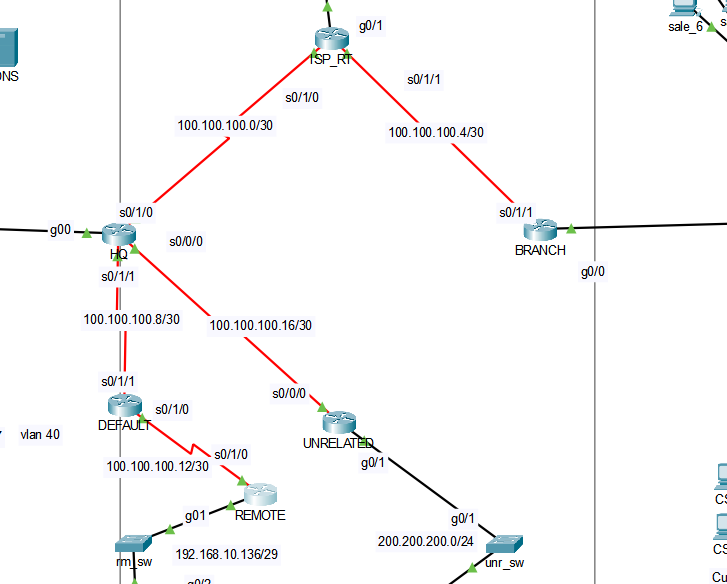
**SUM:**  
**\*WE are using mac violation because we matching each interface just for one PC, if there is an pyshical attack, it make us triggered and we will prevent it because after MAC changing we close the connection to new MAC address**.

**WAN(PPP)  
Links and IP addresses**

**Conf of WAN**

|  |  |
| --- | --- |
| **DEFAULT(s0/1/1)** | **DEFAULT (s0/1/0)** |
| DEFAULT>en  DEFAULT #conf t  DEFAULT (config)#int s0/1/1  DEFAULT (config-if)#ip address **100.100.100.10** 255.255.255.252  DEFAULT (config-if)#no sh  DEFAULT (config-if)#encapsulation ppp | DEFAULT (config)#ex  DEFAULT (config)#int s0/1/0  DEFAULT (config-if)#ip address **100.100.100.13** 255.255.255.252  DEFAULT (config-if)#no sh  DEFAULT(config-if)#encapsulation ppp |

|  |  |
| --- | --- |
| **REMOTE(s0/1/0)** | **REMOTE(g0/1)** |
| REMOTE>en  REMOTE #conf t  REMOTE (config)#int s0/1/0  REMOTE (config-if)#ip address **100.100.100.14** 255.255.255.252  REMOTE (config-if)#no sh  REMOTE (config-if)#encapsulation ppp | REMOTE>en  REMOTE #conf t  REMOTE (config)#int g0/1  REMOTE (config-if)#ip address **192.168.10.137** 255.255.255.248  REMOTE (config-if)#no sh |

****

|  |  |
| --- | --- |
| **BRANCH(s0/1/1)** | **UNRELATED** |
| BRANCH>en  BRANCH(config)#int s0/1/1  BRANCH(config-if)#ip address **100.100.100.6** 255.255.255.252  BRANCH(config-if)#no sh  BRANCH(config-if)#encapsulation ppp | BRANCH>en  BRANCH(config)#int s0/0/0  BRANCH(config-if)#ip address **100.100.100.18** 255.255.255.252  BRANCH(config-if)#no sh  BRANCH(config-if)#encapsulation ppp |

|  |  |  |
| --- | --- | --- |
| **HQ (s0/1/0)** | **HQ(s0/1/1)** | **HQ(s0/0/0)** |
| HQ>Password:  HQ#conf t  HQ(config)#int s0/1/0  HQ(config-if)#ip address **100.100.100.1** 255.255.255.252  HQ(config-if)#no sh  HQ(config-if)#encapsulation ppp | HQ(config-if)#ex  HQ(config)#int s0/1/1  HQ(config-if)#ip address **100.100.100.9** 255.255.255.252  HQ(config-if)#no sh  HQ(config-if)#encapsulation ppp | HQ(config-if)#ex  HQ(config)#int s0/0/0  HQ(config-if)#ip address **100.100.100.17**  255.255.255.252  HQ(config-if)#no sh  HQ(config-if)#encapsulation ppp |

**What the commands do?**

REMOTE>en #enable device

REMOTE #conf t #we can configure terminal

REMOTE (config)#int s0/1/0 # going to interface s0/1/0

REMOTE (config-if)#ip address **100.100.100.14** 255.255.255.252 **#**giving ip address

REMOTE (config-if)#no sh #not allowed to shut interface down

REMOTE (config-if)#encapsulation ppp

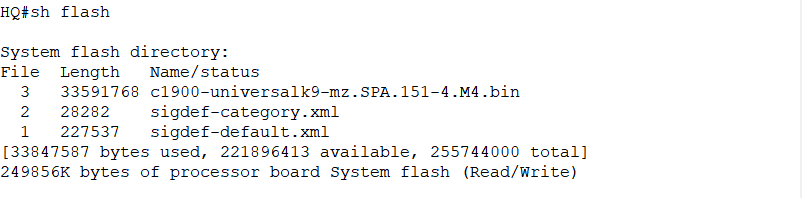
**#**it is point-to-point protocol and it provides authentication on serial interfaces

|  |  |  |
| --- | --- | --- |
| **ISP\_RT (s0/1/0)** | **ISP(s0/1/1)** | **ISP(g0/1)** |
| ISP\_RT>en  ISP\_RT#conf t  ISP\_RT(config)#int s0/1/0  ISP\_RT(config-if)#ip address **100.100.100.2** 255.255.255.252  ISP\_RT(config-if)#no sh  ISP\_RT(config-if)#encapsulation ppp | ISP\_RT(config)#int s0/1/1  ISP\_RT(config-if)#ip address **100.100.100.5** 255.255.255.252  ISP\_RT(config)#no sh  ISP\_RT(config-if)#encapsulation ppp | ISP\_RT(config)#int g0/1  ISP\_RT(config-if)#ip address **192.169.1.1**  255.255.255.252  ISP\_RT(config)#no sh |

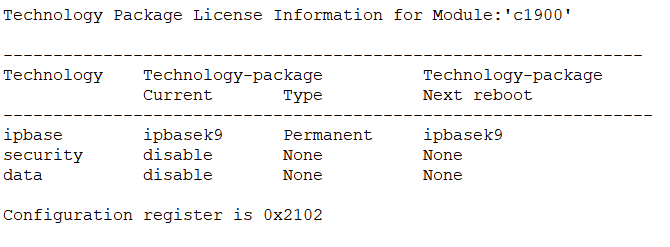
**SUM:**

* **we give IP address manually on serial interfaces and encapsulates packet into a PPP frame to providing security in Layer 2**

**VPN CONFIGURATION  
  
Checking for configuring security files like (k9):**

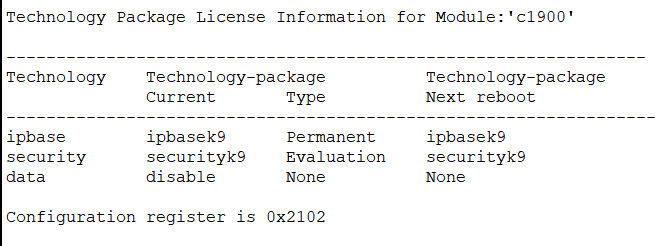
****

**Even though we have that file, due to not having security license the vpn couldn’t implemented, so :**

****

**we ended up getting security packet:**

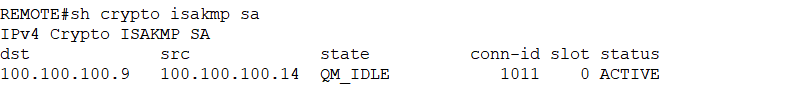
|  |
| --- |
| HQ(config)#license boot module c1900 technology-package securityk9  REMOTE(config)#license boot module c1900 technology-package securityk9 |



|  |  |
| --- | --- |
| REMOTE | HQ |
| REMOTE(config)#crypto isakmp policy 10  REMOTE(config-isakmp)#encr aes  REMOTE(config-isakmp)#hash sha  REMOTE(config-isakmp)#authentication pre-share  REMOTE(config-isakmp)#group 2  REMOTE(config-isakmp)#lifetime 86400  REMOTE(config)#crypto isakmp key VPN\_KEY address 100.100.100.9  REMOTE(config)#crypto ipsec transform-set VPN\_SET esp-aes esp-sha-hmac  REMOTE(config)#access-list 170 permit ip 192.168.10.136 0.0.0.7 192.168.10.0 0.0.0.255  REMOTE(config)#crypto map VPN\_MAP 10 ipsec-isakmp  REMOTE(config-crypto-map)#set peer 100.100.100.9  REMOTE(config-crypto-map)#set transform-set VPN\_SET  REMOTE(config-crypto-map)#match address 170  REMOTE(config-crypto-map)#ex  REMOTE(config)#int s0/1/0  REMOTE(config-if)#crypto map VPN\_MAP  \*Jan 3 07:16:26.785: %CRYPTO-6-ISAKMP\_ON\_OFF: ISAKMP is ON | HQ#conf t  HQ(config)#crypto isakmp policy 10  HQ(config-isakmp)#encr aes  HQ(config-isakmp)#hash sha  HQ(config-isakmp)#authentication pre-share  HQ(config-isakmp)#group 2  HQ(config-isakmp)#lifetime 86400  HQ(config)#crypto isakmp key VPN\_KEY address 100.100.100.14  HQ(config)#crypto ipsec transform-set VPN\_SET esp-aes esp-sha-hmac  HQ(config)#crypto map VPN\_MAP 10 ipsec-isakmp  HQ(config-crypto-map)#set peer 100.100.100.14  HQ(config-crypto-map)#set transform-set VPN\_SET  HQ(config-crypto-map)#match address 170  HQ(config-crypto-map)#ex  HQ(config)#access-list 170 permit ip 192.168.10.0 0.0.0.255 192.168.10.136 0.0.0.7  HQ(config)#int s0/1/1  HQ(config-if)#crypto map VPN\_MAP  \*Jan 3 07:16:26.785: %CRYPTO-6-ISAKMP\_ON\_OFF: ISAKMP is ON |

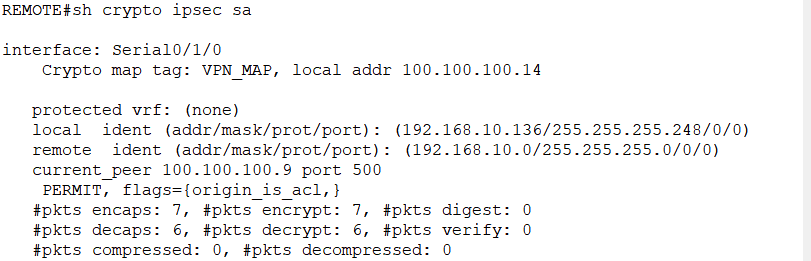
**CONFIGURATION:**

**Then the statue is active**

****

**AND**

**We can see encrypted packets:**



**How vpn commands works? And What are those?**

REMOTE(config)#crypto isakmp policy 10 # defines an ISAKMP policy with priority **10**

REMOTE(config-isakmp)#encr aes # it says encryption will be AES encryption for securing the VPN

REMOTE(config-isakmp)#hash sha # it says that hashing will be SHA hashing for integrity verification

REMOTE(config-isakmp)#authentication pre-share #pre-shared keys will be used for authentication

REMOTE(config-isakmp)#group 2 # VPN will use Diffie-Hellman Group 2 for key exchange

REMOTE(config-isakmp)#lifetime 86400 # we set the lifetime(duration) for 24 hours

---IPsec configuration---

REMOTE(config)#crypto isakmp key VPN\_KEY address 100.100.100.9 #pre-shared key for that HQ

REMOTE(config)#crypto ipsec transform-set VPN\_SET esp-aes esp-sha-hmac # we will use esp-aes for encryption and esp-sha-hmac for hashing

REMOTE(config)#access-list 170 permit ip 192.168.10.136 0.0.0.7 192.168.10.0 0.0.0.255 #it says HQ router and Remote Network traffic will be encrypted

REMOTE(config)#crypto map VPN\_MAP 10 ipsec-isakmp #Creates a crypto map for IPsec

REMOTE(config-crypto-map)#set peer 100.100.100.9 #We indicate the HQ router as a peer

REMOTE(config-crypto-map)#set transform-set VPN\_SET #That defines encryption methods we set before

REMOTE(config-crypto-map)#match address 170 # The traffic encryption will be according to access-list 170

REMOTE(config-crypto-map)#ex #exiting from crypto-map

REMOTE(config)#int s0/1/0 #going to interface s0/1/0

REMOTE(config-if)#crypto map VPN\_MAP #appling the crypto map that interface

\*Jan 3 07:16:26.785: %CRYPTO-6-ISAKMP\_ON\_OFF: ISAKMP is ON \*Our VPN is active

HQ(config)#license boot module c1900 technology-package securityk9

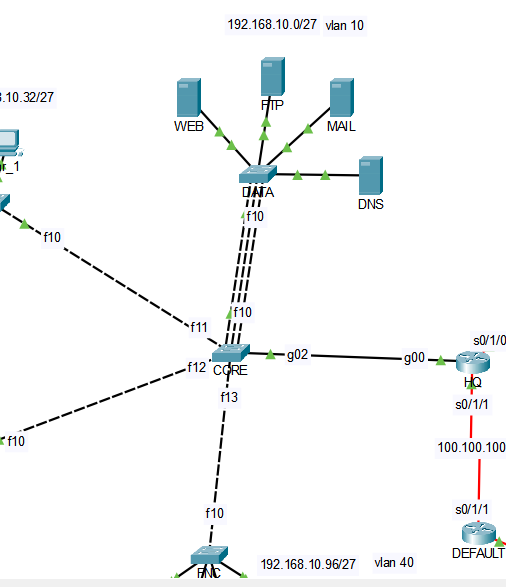
**#**booted that security packet because without it we couldn’t use VPN

**SUM:**

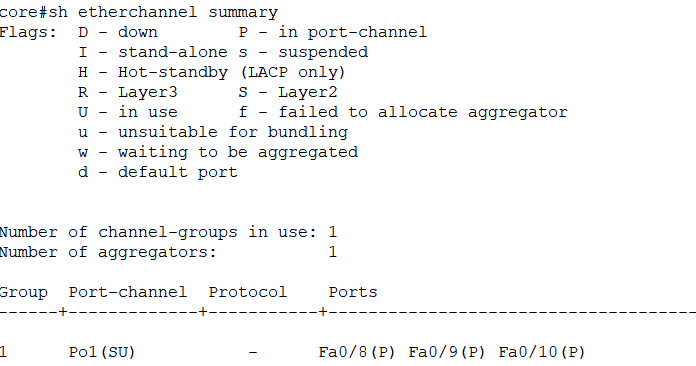
* **At the beginning I wasn’t able to use VPN but instead of VPN I could just make TUNNEL. But that time it wouldn’t provide me encryption. So I ended up loading and operating security packet like (k9).Then I created VPN\_MAP which determines which tarffic will be encypted. Moreover, In encryption there was 2 phase:  
  -first one is ISAKMP policy , I indicated the hashing and encryption function such as aes and sha for tunnel security and pre-shared authentication.**

**-second one is Transform Set, in this I named set such as VPN\_SET and defined hash and encryption function like esp-aes esp-sha-hmac for protecting actual data.**

**ETHERCHANNEL**



|  |  |
| --- | --- |
| **CORE** | **DATA** |
| core#conf t  core(config)#int range f0/8-10  range)#switchport mode trunk  core(config-if-range)#sw tr nat vlan 99  core(config-if-range)#channel-group 1 mode on | DATA#conf t  DATA(config)#int range f0/8-10  DATA(config-if-range)#sw tr nat vlan 99  DATA(config-if-range)#channel-group 1 mode on |



P = Port is bundled and working  
SU = Layer 2, up and active

**What are etherchannel commands?**

core#conf t #configure the terminal

core(config)#int range f0/8-10 # configure multi-interface at the same time

core(config-if-range)#switchport mode trunk #if I making them trunk then that means they will be carrying more than one access interface

core(config-if-range)#sw tr nat vlan 99

**#**it is “switchport trunk native vlan 99” meaning native vlan for those interface is vlan 99. Native vlan carries untagged traffic.

core(config-if-range)#channel-group 1 mode on

**#** we created ether channel to make multiple links act as one and help bandwith.

**#**Also it says we are naming etherchannel as 10

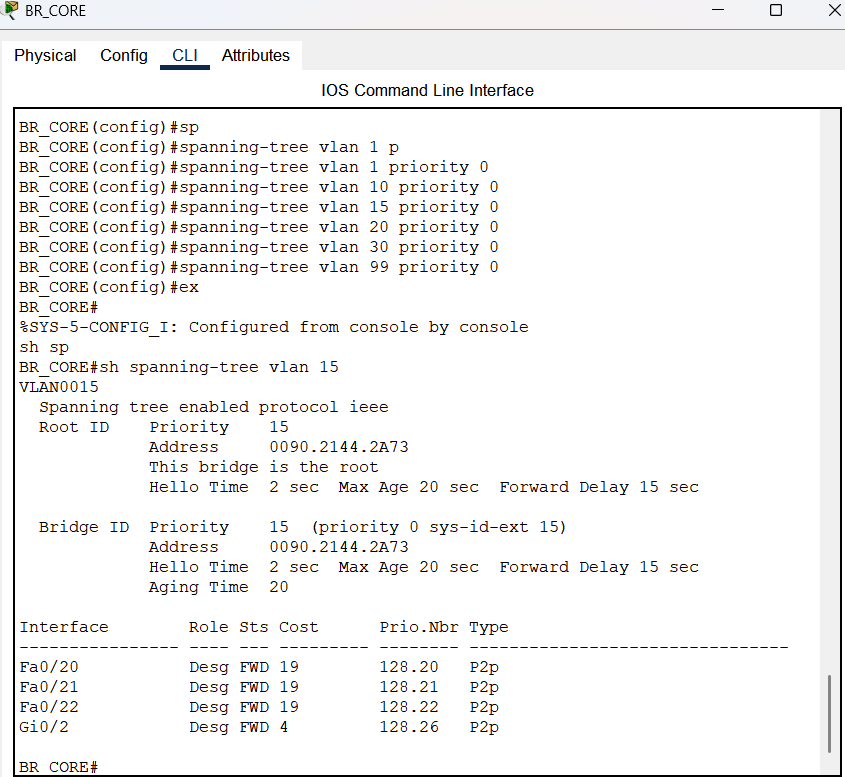
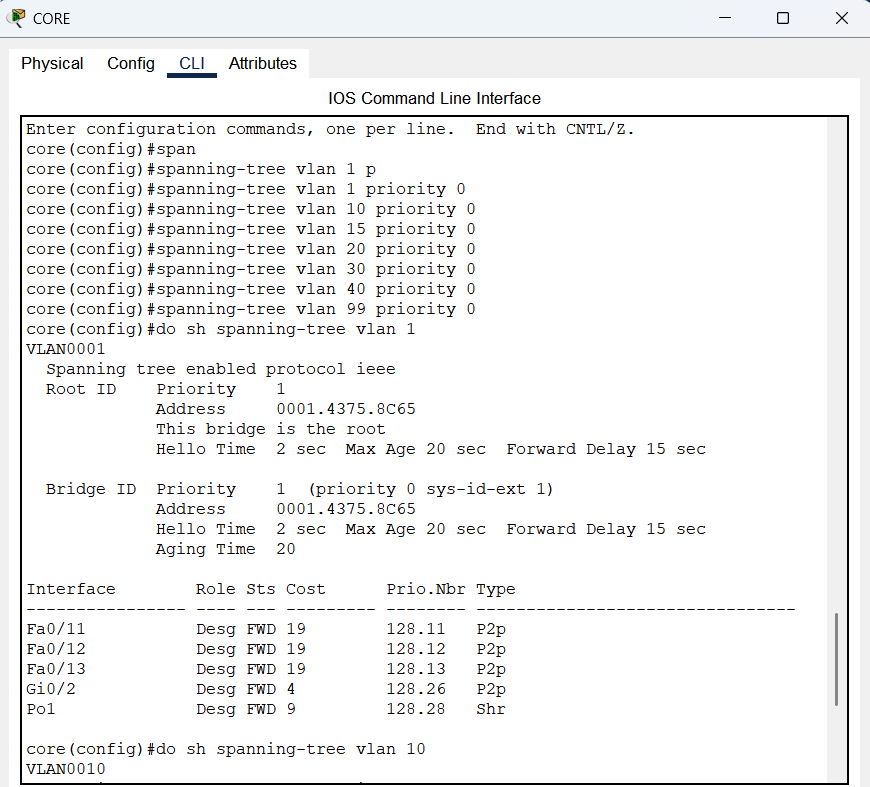
**“Mode on “** is I am configuring etherchannel without negotiation such as (LACP, PAgP)

**SUM:**

* **There is data center and generally traffic goes that way more, so I added ether channel there to reduce conjunction and increase bandwith. So I choose 3 interface and merge them as a one, after that made channel-group on them. Also, I made same configuration both sides. So now, the system will provide us efficiency**

**STP(Spanning-tree-protocol)**

**CORE BR\_CORE**

****

We created that to prevent a broadcast storm and control the all traffic and of course for preventing looping

**What commands are used?**

core (config)# spanning-tree mode pvst( #making its mode pvst to run STP separately for each VLAN

core (config)# spanning-tree vlan 10 #priority 0 like image we decided give the highest priority to that vlan

data(config)#interface range f0/1-4 #going to a range(f0/1-4)  
data(config-if-range )#spanning-tree portfast #make their connection fast

\*I do this for all access interface and that provides me when I open the pkt.file at the begging of everything we see some orange (disconnection) but when the access interfaces configured as portfast, then they won’t wait for negotiation and will be able to send packets. We couldn’t do it for trunks because it is not possible.

**THE CHELLANGE I FACED**

|  |  |
| --- | --- |
|  | After configuration like beside,  T defines Trunk ports   A defines Access ports |

**Problem\_1:** I should have created vlans on each sw before determining as a client because I didn’t and get the error of (‘VTP VLAN configuration not allowed when device is in CLIENT mode.’)

**Problem\_1.2**: If I made configuration right that kind of error wouldn’t pop up, because we are already using VTP not to make switch configure a vlan on its own.  
**Problem\_2:** When I created vtp ı put all switch same vtp domain network but to give different password for each switch made vtp run out. Hence, I should have given same password to run VTP correctly.  
  
  
  
**Problem\_3:**   
When I write a rule for dns  
there was a problem the destination port was always changing so I write something like this

Extended IP access list 130

10 permit tcp host 192.168.5.2 host 192.168.10.6 eq ftp

20 permit tcp host 192.168.5.34 host 192.168.10.6 eq ftp

30 permit tcp host 192.168.5.66 host 192.168.10.6 eq ftp

40 deny tcp any host 192.168.10.6 eq ftp

50 permit tcp 192.168.5.0 0.0.0.255 host 192.168.10.7 eq smtp

60 permit udp host 192.168.1.2 any eq domain

65 permit udp any host 192.168.1.2 eq domain

70 permit ip any host 192.168.1.2

80 permit tcp 192.168.5.0 0.0.0.255 host 192.168.10.5 eq 443 (10 match(es))

90 permit eigrp any any (313 match(es))

100 deny icmp any 192.168.10.0 0.0.0.31 (4 match(es))

110 permit icmp any any (85 match(es))

HQ(config-ext-nacl)#no 60

HQ(config-ext-nacl)#no 65

HQ(config-ext-nacl)#60 permit udp host 192.168.1.2 any

HQ(config-ext-nacl)#no 70

HQ(config-ext-nacl)#70 permit udp any host 192.168.1.2 eq 53

**--Solution--**

Then I solve that problem the packet come from ISP\_DHCP woudn’t be configured like 60 permit udp host 192.168.1.2 any eq domain  
because we don’t know port number the destination is 53 but source is random one so when we write acl which deny any ip to reach on s0/1/1, it deny every packet even though we write acl to prevent this I couldn’t notice that when ISP\_DHCP answer the packet but it will put the random port as a destination. Therefore, that ACL wouldn’t work because the rule is not related to random port.

So I applied that:

60 permit udp host 192.168.1.2 any I wrote that permission without “eq domain” so can make ISP\_DHCP sent its answer.

**Problem\_3:** I tried both side nat conf but when ı do that hq and branch start not to ping  
Then I had got this scnerio in simulation and I saw everything was okay but when the packed come back they was changing source address, So the first destination address and expected address wasn’t watching. Because of this, I couldn’t ping hq\_vlans to branch\_vlans  
Then I thought there would be way and decided to use extended acl   
Solution:  
I learned when we want two network which has NAT connected , we cannot connect without writing specific ACL,SO I wrote ACL 110 for that reason. And it says when a packet want to go BRANCH : don’t let IP converted when the packet goes to HQ.

I used that:

Extended IP access list 110

10 deny ip 192.168.10.0 0.0.0.127 192.168.5.0 0.0.0.127

20 permit ip 192.168.10.0 0.0.0.127 any

**Problem 4:** VPN creating   
**Solution:**  
I handle it with loading a packet. So, it works on remote side.  
I got the license.

HQ(config)#license boot module c1900 technology-package securityk9

**PROBLEM 5:** IPV6 and NAT/64

No solution due to packet tracer

I couldn’t make nat-pt/nat64 conf because packet tracer doesn’t support it, even so I have given hq s0/1/1 interface ipv6 IP address but it wont be used.