How the network was created

Creating the subnet :  
For this part of the assignment, we were tasked with creating a network for HSBC. The main branch Leeds (70 hosts) the other branches being Birmingham (250 hosts), Edinburgh (220hosts), Manchester (130hosts} Had to stay between the 192.x.x.x/ 22. We had to create an ipv4 addressing based on the design requirements. This included the address for future devices. We had to create a table that provides all the details including network address, subnet mask, first usable address, last useable address and broadcast address.

Creating the table:

I had to start off the base with a class C network that was within the 192.x.x.x/22, I wanted to make it easier for me calculate the address. I started with the normal 192.168 .0.0.

You always must start with the most host, so this means that I had to start with Birmingham. When creating the table, you always start off with the first address so in this case it was 192.168.0.0 you then move on to the next part of the subnet table. You go up in powers of 2 up until, 2^8=256(max hosts) you want to find out how many hosts can fit under one of the powers so in this case it was bigger than 2^7= 128 but has less hosts than 2^8=256. This would mean there are 8 zeros in the subnet mask. You then take away the number of zeros by 32 as there are 32bits in a subnet mask Birmingham's was /24. Next on the table, was first useable host, you just add 1 to the original the first useable host would be 192.168.0.1 you then move onto the next which is broadcast, you add in the amount in the subnet mask and minus it by one, 192.168.0.256 then you take this away by 1 because in ipv4 the broadcast takes one so it becomes 192.168.0.255 finally for Last you take away one for the host of this one so it becomes 192.168.0.254 This is your first network then you move onto the next.

Edinburgh you do this one next as this is the second biggest, as the last host has reached its maximum of 256, you have to now add 1 to the zero before the last point 192.168.1.0 becoming your new start address, you then do the same again for the subnet mask Edinburgh has 220hosts it is bigger than 2^7=128 but less then 2^8= 256 so the subnet mask is /24 . You then move onto the next this being first useable add one to the end so 192.168.1.1 broadcast is the same as before, the broadcast 192.168.1.255, then for the last useable host you minus 1 from the broadcast for the network which becomes 192.168.1.254.

Manchester is the next biggest hosts. The last host has reached a maximum of 256 so you must add one to the point behind to free up some space. 192.168.2.0. You start with this address. Do the same for the subnet mask Manchester has 130hosts more 2^7=128 but less then 2^8=256 so it will be /24, after the subnet mask you add one for the first useable host 192.168.2.1 /24 then you move onto the broadcast take away 1 from the max amount of host 192.168.2.255 and then for the last it will be taking one off the broadcast so this ends up being 192.168.2.254.

Leeds is last. You must add one to the last point to free up space, becoming 192.168.3.0 as the start, to get the subnet mask you see how many hosts Leeds has which is(70) less then 2^7=128 but more then 2^6=64 so the subnet mask will have 7 zeros, so you take this away leading to /25 you then do first add one to starting so 192.168.3.1 you then move on to the broadcast this time 2^7= 128 so you take one away from that 192.168.3.127. The last is taking one away from the broadcast 192.168.3.126

the whole table that I will be basing part 2 of section A

Refer to V.1 in the appendix

When making this have, I have considered future expansion, so I have added in extra spaces in the network if more hosts are to join. By using this table I can now move on to create my point-to-point network topology on packet tracer which is part B.

Part B

By using cisco Packet tracer, I have created a point-to-point topology as this logical topology was the one that worked with this network. I designed this using:

* 5-routers
* 4-switches
* 8-pc

Usings the routers to communicate with the other networks and the switches to communicate within the network.

Creating the routers.

We needed a router for each branch as well as another router for the ISP which is the internet service provider allowing the company to use the internet. Setting up the routers meant that everything had to be connected via serial ports. The base routers start off with fast Ethernets but only two of them so I couldn’t connect them to other routers, so I added in extra port space. To do this you have to press on the router, press physical then you press on wic-2T, turn off the router and drag and variable into the router turn back on the router. Please refer to V.2 on the appendix.

After doing this to all the routers and connecting them via serial ports it's time to connect the routers together. As the routers are connected via serial port outside of the network, the Ip addresses do not have to be the ones that you have created within the network, meaning you can put any Ip address onto this network:

* Birmingham=
  + Serial 0/0=192.168.10.1 255.255.255.252
* Leeds=
  + Serial 0/0/0=192.168.10.10 255.255.255.252
* Manchester=
  + Serial 0/0=192.168.10.6 255.255.255.252
  + Serial 0/1=192.168.10.5 255.255.255.252
* Edinburgh=
  + Serial 0/0= 192.168.10.2 255.255.255.252
  + Serial 0/1= 192.168.10.5 255.255.255.252

I made them all within the same Lan network so that they are all able to connect and communicate with each other, otherwise pinging would be difficult to achieve.

Please refer to v.3 in the appendix to see the command being done in packet tracer

After setting up the serial ports with their specific Ip addresses, you have to route them so that they can communicate with each other. To do this you start off with the Ip address of the network that you are trying to communicate with and the subnet mask of that network, however you always put a .0 at the end to show that you are jumping to the whole network.

Brimingham- Edinburgh= 192.168.1.0 255.255.255.0 192.168.10.2

Please refer to v.4 in the appendix for the whole image of the route

The first address is the router's whole network, the second is the subnet mask, the third is the serial port that it's connected to; after doing this on every router the routers should be able to communicate with each other. You can test this later when you have added in the pcs and the switches. However, you can see that they have been connected when the arrow next to routers has been highlighted as green. Please refer to v.5 in the appendix to see an example

After connecting the switches this is where the table will be very helpful as you have to assign the ip address of the hosts based on the table that you have created, to assign this press on the cable that is connecting your switch to the router enter commands shown in

B.1 of the appendix

Vlan 1 then you will see that the line between switch and router has lit up green. Please refer to the example in V.6

Now everything is connected, you need to secure the network of the routers by adding in passwords and cryptography.

I made it as all the passwords that enable router cli access is 123 .

Please refer to v.7 in the appendix to see the commands in the cli

I continued to do this on all of the routers to ensure that the router is safely secure as well as this I also added in service password-encryption. This makes it so that the password doesn’t get stored in text so no one can see the inputs.

Please refer to v.7 in the appendix to see what it looks like when you enter the cli of the router

After you have secured it with password it's time to secure it using SSH, secure shell allowing the hosts to be secure even in unsecure websites.

To do this you have to first define the hostname that you want to use ssh on and the Ip domain for mine I used abc.net. What you do now is generate the crypto key, I would recommend 1048-2064bits are the most secure as 1048 is the discrete logarithm.

Then to restrict remote access:

* Line vty 0 4
* Transport input ssh
* Login local

Please refer to v.8 in the appendix to examples of this in the CLI

After this all the routers were ready to go

Creating the switches

The switch is only used so that the devices within the network can communicate. So, what I done was that I connected switch to the routers using fast ethernet 0/1 then on the router you configure the Ip as said in “creating the router” you then connect the switch to the pcs using fastEthernet 0/2 and 0/3 so now the switches are connected to the Pc,

you will need to configure the Vlan 1 where you just input the ip address of another available host ip address of your table, you then input the commands for changing ip but interface Vlan1. Then you do that to all the switches so now the switches are all fully working.

Creating the Pc

To create go into the pc and you press ip configuration, once you are there you will add in a ipv4 address as well as a subnet mask and a default gate way you will need to use the next available network for each different pc. Eg 192.168.0.4 on pc1 and on pc2 192.168.0.5 for Birmingham

If the pc is inside the same network, you can use the same subnet mask as the one that you created in your table EG. 255.255.255.0 Birmingham one

The default gate should be one of the serial ports of the router this is so that you can send messages outside the servers on the pc eg 192.168.10.1

Refer to v.9 in the appendix

Testing

The final part was testing that they all ping to each other, to show that they were all connected to do this I had to use 5 show commands these beings

* Show interface brief- this shows all the interfaces are in the up state and that all the ip are correctly assigned. V.10 appendix for the result of the interface brief
* Show ip route- this shows all the networks are listed and shows correct next hops v.11 to see the results
* Show cdp neighbors- this shows all the directly connected cisco devices, their interfaces to ensure that all devices are listed v.12 appendix results
* Show running-config- shows active configurations, this includes routing settings and security to ensure everything is working v.13 appendix results
* Show vlan brief- this shows that the switch configurations are working working v.14 appendix

As well as all the show commands I also used commands like ping to show that everything was working, and all the packets were getting send

Please refer to v.14 for the pings working and the successful packets sent!

C) Firewall IDS/IPS

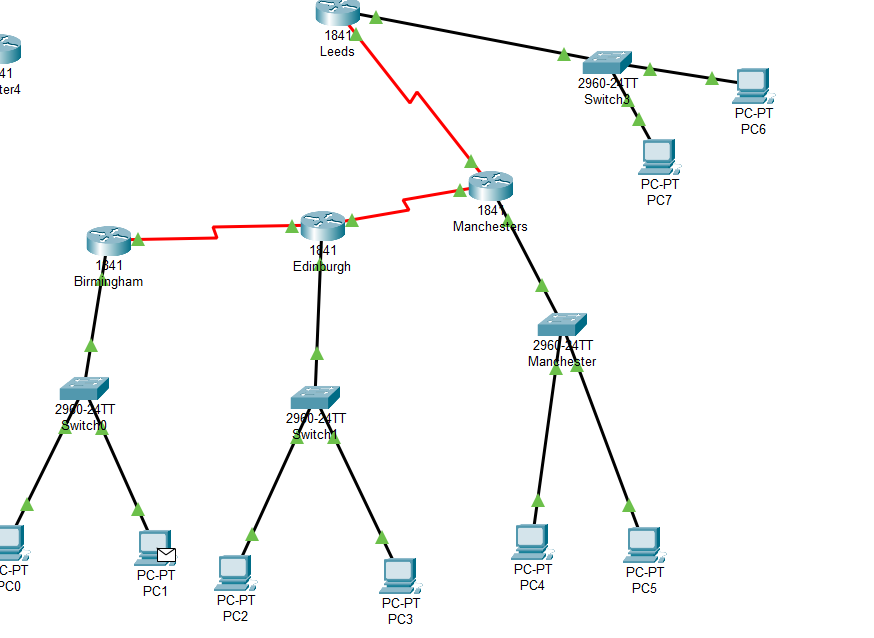
I also tried to secure it using a zone firewall and ips/ids- these are more ways to secure the network and to make it safer from hackers, HSBC is such a big company making sure the network is secure with more than passwords is important.

Firewall options- I added in a Zone firewall on all the routers so the network security can be segmented into different zones with different level of security as firewalls create a barrier for internal trusted networks and external untrusted networks, firewalls make sure security policies are met by inspecting traffic the fire wall will examine packets based on header information and it will either allow or deny access to the server. As well as implementing a hardware and software firewall will make sure the network is mostly secure.

IDS/IPS- Intrusion detection system scans and detects potential threats, it will log the information report the activity for further investigation, you cannot implement it on cisco packet tracer. Software-like Snort and Suricata HSBC can be deployed.

Intrusion prevention system- Detects, Blocks and reduces threats in real time, by directly inspecting and controlling the flow of data packets. It works in the traffic path so all the packets flow through it so it can defend as soon as it's detected unlike IDS. I tried to implement IPS and you cannot but like IDS real life Snort, Suricata are software HSBC can deploy.

Please refer to v.15 and v.16 in the appendix to see the commands for both the ZBFW commands

Photo of the final network

Appendix

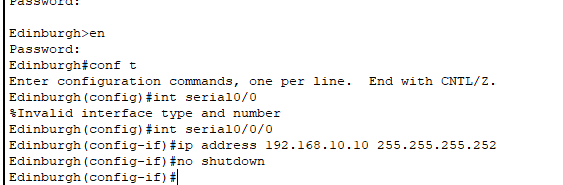
V.1 Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name of place | ipv4 | Subnet | First | Last | Broadcast |
| Birmingham | 192.168.0.0 | /24 | 192.168.0.1 | 192.168.0.254 | 192.168.0.255 |
| Edinburgh | 192.168.1.0 | /24 | 192.168.1.1 | 192.168.1.254 | 192.168.1.255 |
| Manchester | 192.168.2.0 | /24 | 192.168.2.1 | 192.168.2.254 | 192.168.2.255 |
| Leeds | 192.168.3.0 | /25 | 192.168.3.1 | 192.168.3.126 | 192.168.3.127 |

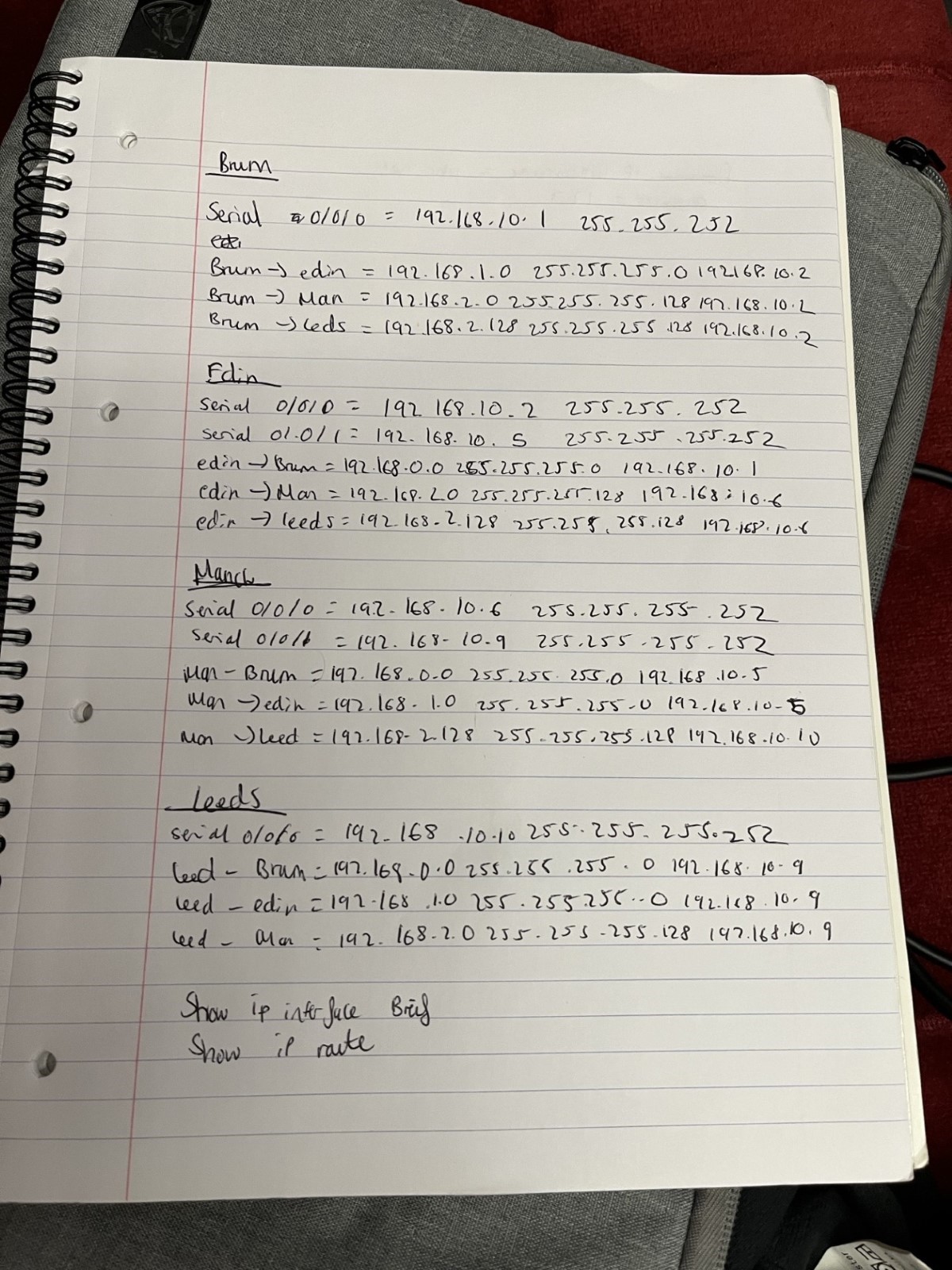
V.2 Input physical onto the router



v.3 Commands for inputting ipv4 address



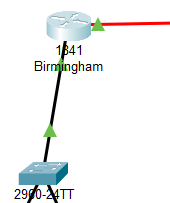
v.4 Paper routing that I worked out



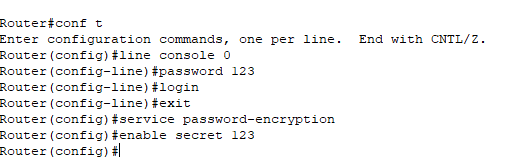
v.5 Green highlight to show the routers connection



v.6 Green highlight to show the switch are connected



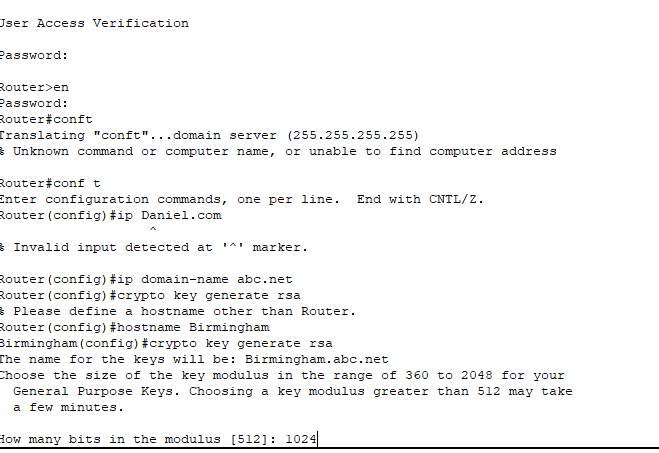
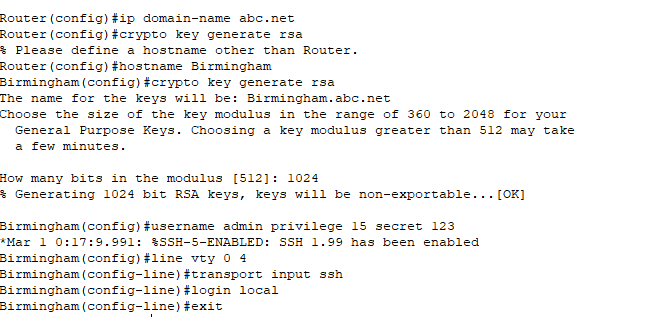
v.7commands in the cli for password



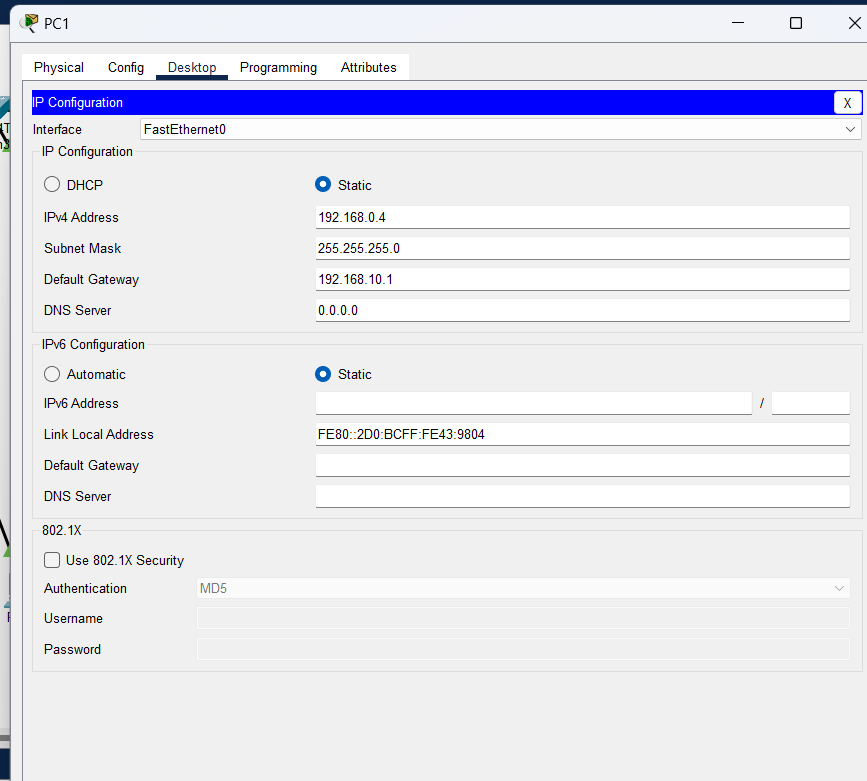
v.7 password



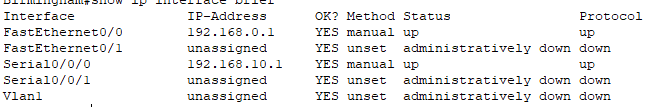
v.8 examples of securing the router



v.9 Computer input

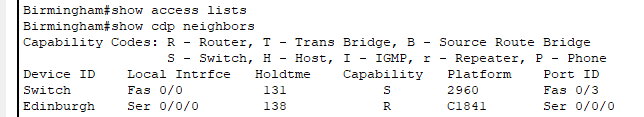


V.10 interface brief

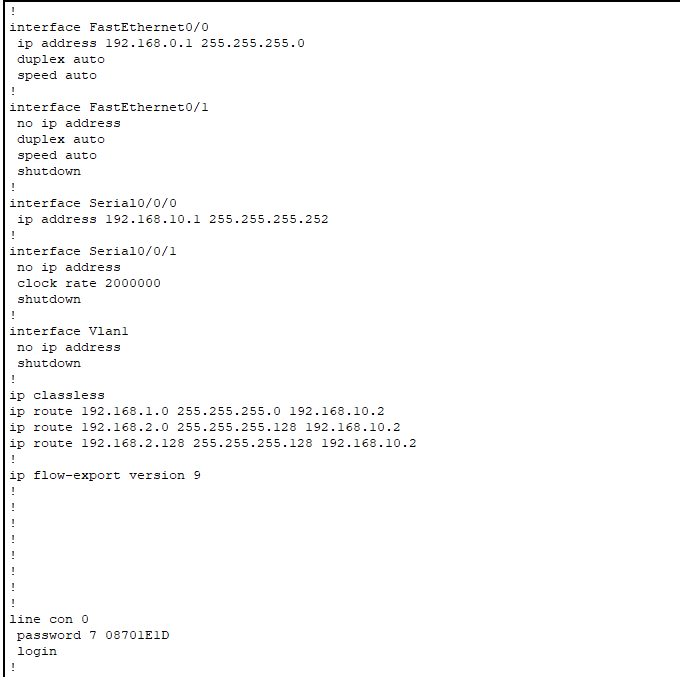


V.11

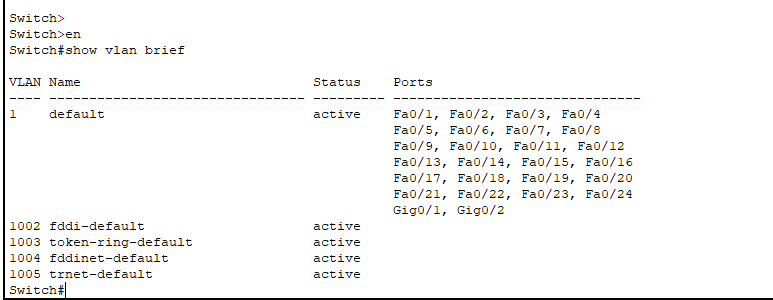


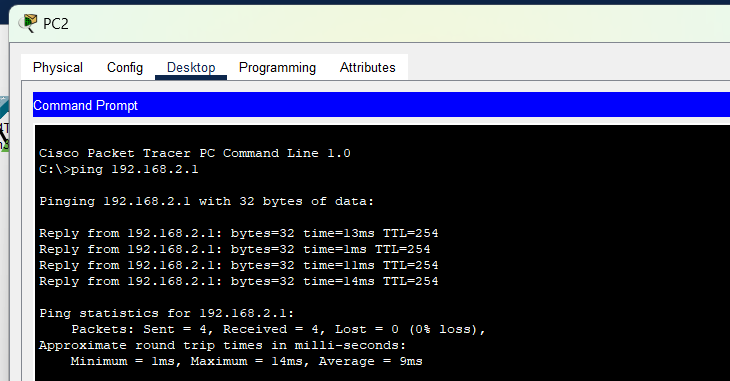
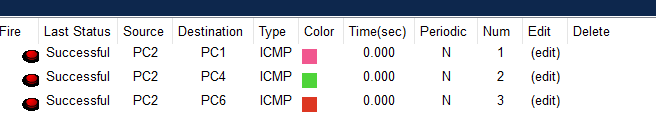
v.12

v.13

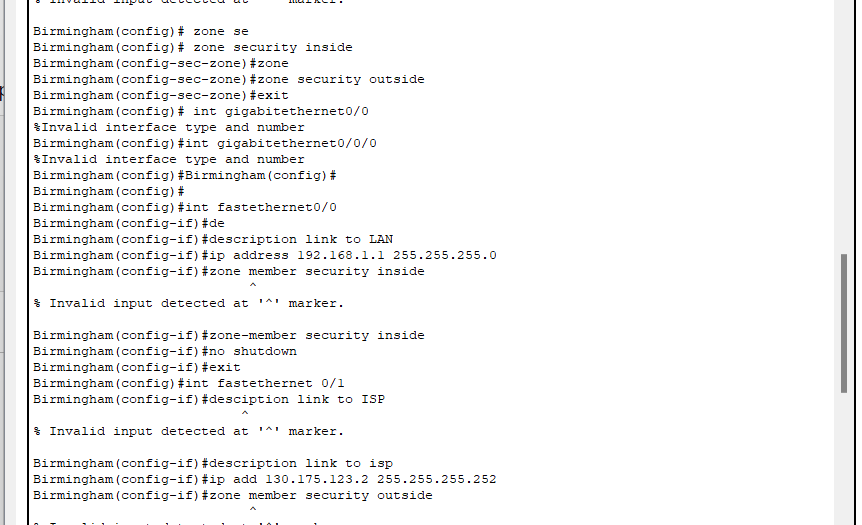


v.14

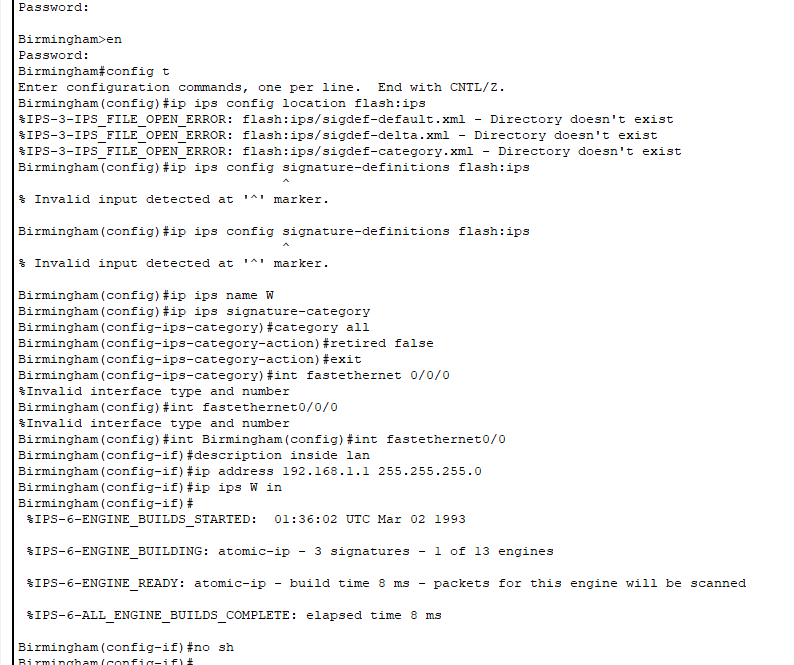


V.15

v.15- adding in zone security on a router



v.16 using isp

B1. Configure connection between router and switch

