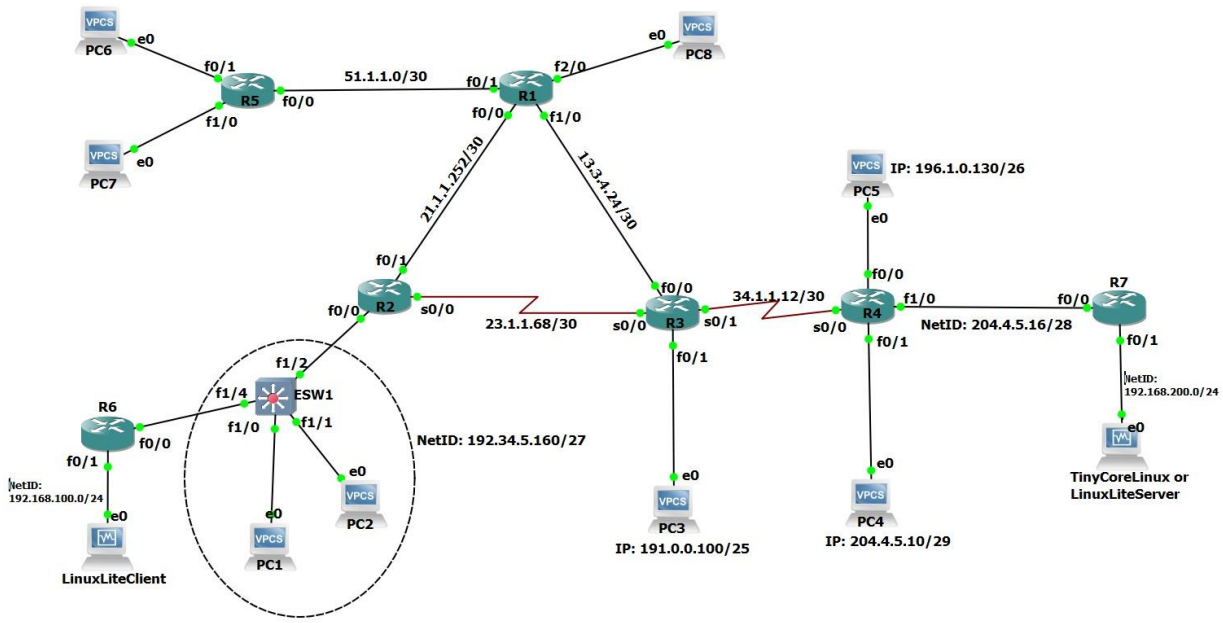
Netlab 7: VPN in GNS3

Purpose: In this Lab exercise, you will experiment with Network Address Translation (NAT) and Firewalls

Procedure

We are using the topology from the previous lab. We need to do a few modifications to it. Use the depicted topology below to implement the changes. In this case, routers R6 and R7 are going to operate as VPN Firewalls devices. It is a good idea to create a snapshot of the previous lab in case you need to go back to that configuration.



Notes:

1.- Login credentials for the VMs are: username: tc and pass: netlab2020

2.- If you are using TinyCoreLinux VM, you need to update the password for the user ***tc***. In a CLI window apply the command: sudo passwd tc (enter the password **netlab2020**) . If you are using the LinuxLiteServer VM, you don't need to do do this step.

VPN Setup

As discussed during lecture, IPSec VPNs "tunnel mode" are formed in two phases with the ultimate goal to create 3 security associations (SA). Phase 1 is a SA for management. The other two, which happen in phase 2, provide the transform set parameters to form the SA for TX and RX traffic. Let's apply the commands listed below for the two pahses:

Phase 1 – R6

1. Create an ISAKMP (internet security association key management protocol)

R6(config)#crypto isakmp policy 10
R6(config-isakmp)#encryption des
R6(config-isakmp)#hash md5
R6(config-isakmp)#authentication pre-share R6(config-isakmp)#exit

1. Specify pre-shared key and the **remote** peer address. Note that the pre-shared key has to match with the the one in R7 (*NETLAB2020*)

R6(config)#crypto isakmp key 0 NETLAB2020 address 204.4.5.18

Phase 2 – R6

1. Create the phase 2 for actual data encryption.

R6(config)#crypto ipsec transform-set NETLABSET esp-des esp-md5-hmac R6(config-crypto-trans)#exit

2. Create the ACL for the traffic to be encrypted

R6(config)#access-list 110 permit ip 192.168.100.0 0.0.0.255 192.168.200.0 0.0.0.255

3. Create the crypto map. This step will put all phase 2 parameter to work together. Specify the peer IP address, transform set, and ACL for the split tunnel

R6(config)#crypto map NETLABMAP 10 ipsec-isakmp
R6(config-crypto-map)#set peer 204.4.5.18
R6(config-crypto-map)# set transform-set NETLABSET
R6(config-crypto-map)#match address 110
R6(config-crypto-map)#exit
R6(config)# access-list 110 permit ip 192.168.100.0 0.0.0.255 192.168.200.0 0.0.0.255

4. Apply the crypto map in outside interface (note: this is based on the topology depicted in this paper, your interface name might have different name)

   R6(config)#interface f0/0
   R6(config-if)# crypto map NETLABMAP
   R6(config-if)#end

You should see a message in the CLI that says the ISAKMP is ON. Save you configuration with "wr mem". Repeat the same steps in R7.

Phase 1 – R7

1. Create an ISAKMP (internet security association key management protocol)
   R7(config)#crypto isakmp policy 10
   R7(config-isakmp)#encryption des
   R7(config-isakmp)#hash md5
   R7(config-isakmp)#authentication pre-share
   R7(config-isakmp)#exit

2. Specify pre-shared key and the **remote** peer address. Note that the pre-shared key has to match with the the one in R6 (*NETLAB2020*)

   R7(config)#crypto isakmp key 0 NETLAB2020 address 192.34.5.162

Phase 2 – R7

1. Create the phase 2 for actual data encryption

   R7(config)#crypto ipsec transform-set NETLABSET esp-des esp-md5-hmac R7(config-crypto-trans)#exit

2. Create the ACL for the traffic to be encrypted

   R7(config)#access-list 110 permit ip 192.168.200.0 0.0.0.255 192.168.100.0 0.0.0.255

3. Create the crypto map. This step will put all phase 2 parameter to work together. Specify the peer IP address, transform set, and ACL for the split tunnel

   R7(config)#crypto map NETLABMAP 10 ipsec-isakmp
   R7(config-crypto-map)#set peer 192.34.5.162
   R7(config-crypto-map)# set transform-set NETLABSET
   R7(config-crypto-map)#match address 110
   R7(config-crypto-map)#exit
   R7(config)#access-list 110 permit ip 192.168.200.0 0.0.0.255 192.168.100.0 0.0.0.255

4. Apply the crypto map in outside interface (note: this is based on the topology depicted in this paper, your interface name might have different name)

   R7(config)#interface f0/0
   R7(config-if)# crypto map NETLABMAP R7(config-if)#end

5. One thing we need to modify in R6 configuration from the previous Lab (NAT Lab) is the ACLs. The current ACL used by the NAT system has to change to deny traffic going to the remote network from by NAT'ed. Delete all the ACLs and the "*ip nat inside source list….*" statement. Apply the following commands (only in R6). Note that the recently created ACL 110 should remain in the configuration

   R6(config)#access-list 100 deny ip 192.168.100.0 0.0.0.255 192.168.200.0 0.0.0.255
   R6(config)#access-list 100 permit ip 192.168.100.0 0.0.0.255 any
   R6(config)#route-map NETLAB permit 10
   R6(config-route-map)#match ip address 100
   R6(config)#ip nat inside source route-map NETLAB interface FastEthernet0/0 overload

6. Similarly, as we got rid of the "*ip nat inside list….*" statement and the previous ACL in R6, we need to do the same in R7 and replace it with the following commands: (Once again the recently created ACL 110 should remain in the configuration)

   R7(config)#access-list 100 deny ip 192.168.200.0 0.0.0.255 192.168.100.0 0.0.0.255
   R7(config)#access-list 100 permit ip 192.168.200.0 0.0.0.255 any
   R7(config)#route-map NETLAB permit 10
   R7(config-route-map)#match ip address 100
   R7(config)#ip nat inside source route-map NETLAB interface FastEthernet0/0 overload

At this point the VPN should be ready to be used. Before moving traffic between the Client and the Server VMs. Execute the following commands in R6 or R7 and answer the questions:

To list the security associations:
R6#sh crypto isakmp sa

To display the IPSec tunnel status:
R6#sh crypto ipsec sa

Take a screen shot of each command execution. You will need it to compare the execution of the same commands after the VPN is ON.

Capture the traffic with Wireshark on the link between R7 and R4. Answer the following questions:

1. Try to ping the server VM at the internal IP (a.i. 192.168.200.2). Explain what Wireshark captured:

More hello packets than before when it passing through the VPN.

2. From the Client VM, open the web browser and enter the URL: http://204.4.5.18:8080

Take note of the capture in Wireshark. Now, open another tab in the Client's web browser and use the internal IP address of the server in the URL (a.i. 192.168.200.2). What difference(s) do you see between the two captures? Explain

The traffic passing through the VPN is different.

3.- From the Client VM, ftp the server VM using the external NAT IP address 204.4.5.18 (enter any password, no need to get authenticated). Take note of the capture in Wireshark. Now delte the port translation set in the previous lab pertaining to FTP. Once the nat translation is removed, use the client FTP application to try to login to the server VM using the internal IP (a.i. 192.168.200.2). What difference(s) do you see between the two captures? Explain

Encrypted traffic

4. Try to ping the severContinue both packet captures but modify the filter from *icmp* to *tcp*. Try to ftp or ssh into the LinuxServer from the LinuxClient using the private IP address. ***Explain*** what your captures are showing:

Seems like there is some kind of interference in the traffic.

5.- Once again execute the following commands in R6 or R7 and answer the questions:

To list the security associations:
R6#sh crypto isakmp sa

To display the IPSec tunnel status:
R6#sh crypto ipsec sa

Take a screen shot of each command execution. Contrast/describe the differences from the previous screen shots you took before using the VPN by sending traffic between the Client to the Server VMs:

Save you configuration and upload your portable project, Wireshark captures, and your Lab paper with answers to OneDrive.