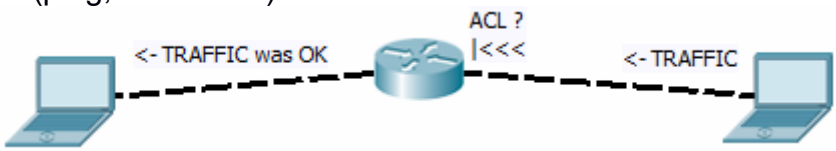
COMPUTER NETWORKS - LABORATORY 072

Subject:
Access Control Lists in Cisco routers

Task A: Datagram Traffic Filtering - standard checklists

1. You must be prepared to work a Cisco router, linking it to the wiring TP (twisted pair) in a router-PC-PC. You must configure and enable the appropriate router interfaces - defining the various directly connected IP network in accordance with generally known principles, eg .:
   Router (config) # int fa 0/0
   Router (config-if) #ip address 200.200.200.1 255.255.255.0
   Router (config-if) shut #no
   Router (config-if) #int fa 0/1
   Router (config-if) #ip address 200.200.201.1 255.255.255.0
   Router (config-if) shut #no
   IP interfaces the PC station must be configured
   Note: In each of the PC station must define the IP address of the default gateway, the IP address of the router interface on the side thereof where the mentioned station.
   Make sure that the router is running in IP routing and possibly run them: Router (config) routing #ip

   After configuring the installation, check the efficiency of communication between stations PC (ping, traceroute).



2. The current series of experiments will involve experimental blocking the network traffic between two stations using PC Access Control List (ACL - Access Controll Lists) defined in the router. These lists are numbered values identifiers. The so-called standard identifiers have a list of 1-99 (ie the value of the identifier when defining the entries in the list determines its type). These lists are simplified and allows blocking traffic for specific IP addresses or groups separately without the possibility of distinguishing features of the sender from the recipient.
   Address range (or a single address), you can specify one of three ways, for example: - any - all the addresses, host- any 10.10.10.0 0.0.0.255 - Network mask

   - host 10.10.10.1 - concrete, one host
3. Create access list, eg .: Router (config) # access-list 90 deny any gdzie 90 is the new ID list, deny that they are still infectious possible values permit (permit) and

remark (comment list).

4. Saved command list can be checked:
   Router # show access-lists
   When refilling the list prohibiting operations on the specific IP remember to add record permitting other traffic, such as .:
   Router (config) # access-list 70 deny host 10.10.10.1
   Router (config) # access-list 70 permit any
   Router (config) # access-list 70 remark is a list removes 10.10.10.1
5. The assignment of letters to the interface fa 0/0 np.:Router(config)#int
   Router (config-if) #ip access-group 90 in
   where n is the filter for packets coming through the interface, similar out - for wychodzących.Sprawdzenie assign ACLs to interfaces:
   Router # show run
6. Deleting the entire list:
   Router (config) #no ip access-list standard 90
   Removing assignment to the interface:
   Router (config-if) #no ip access-group 90 in
   Keep in mind that the structure of the access-lists are not used only for regulate traffic through interfaces, but in many other mechanisms.
   Note: The grouping of hosts in the access-lists are used masks stored in the inversion bit, so for example, a rule:
   Router (config) # access-list 70 deny 10.10.10.0 0.0.0.255
   refers to a pool of addresses 10.10.10.0 - 10.10.10.255 (0 in the mask indicates a command to check the corresponding bit of the address, 1 - ignored)
7. Rules written in letters of access have sequence numbers, guaranteeing order checking (checking is performed in accordance with the rising values of sequence numbers). The numbers are visible when printing lists. To rebuild the numbering must use command, np.:Router(config)#ip access-list resequence 60 20 60 10gdzie this list number 20 is a new beginning for the numbering sequence numbers, 10 is the next increment.

8. In general config mode, you can create lists and add them to the rules without affecting the sequence numbers (assigned automatically as increasing values: 10,20,30, ... etc.). Therefore, to make entries must be in compliance with the order "from the particular to the general." Using a different order (eg. By adding a rule for a network, and later for the host) will cause an error. As a result, the list will not include rules that checked first cover other more specific. To be able to use the selected sequence numbers in defining entries need to go into edit mode list (ACL), eg .:
   Router (config) #ip access-list standard 70
   Router (config-std-nacl) # 26 deny host 10.10.10.1
   where 26 is the sequence number at which the rule is placed in the list number 70. Here, however, also can not lead to a situation where rule with a lower sequence number (previously validated) to bear its generality another one with a higher sequence number.
   When you want to delete the entry for the selected sequence number, type:
   Router(config-std-nacl)#no 26

ACL entries configuration mode of typing the sequence number can also be used for other types of lists discussed in the following sections.
Test correct functioning ACL filters for the traffic between stations PC (after defining the list and assigned to the interface) results in checking the sending of such a movement.
The report generated by the type of command:
Router # show access-list interface fa 0/0 in
Router # show access-list 70
note the number of matches (matches) datagrams for individual entries ACLs.
Counter reset matches:
Router # clear access-list counters

Task B: Traffic filtering datagrams – extended list

1. Using configuration tasks you need to prepare a list extended. They are recorded in a range of identifiers 100 -199. They allow you to specify separate range of IP addresses of the recipient and sender filtering out types of protocols used over IP, datagrams which are the subject of transmission, or other characteristics of these datagrams. In the case of TCP and UDP it is even possible to identify numbers / ranges of ports for these protocols. CLI allows the router here too the use of the names of the protocols associated with these default port numbers (eg. Telnet, SNMP, ISAKMP, FTP, WWW, etc.).
   Principle of defining IP address ranges are subject to filtering is the same as in the lists of simple words are used deny / permit / remark. When the pool of protocols simply choose IP - give up any additional typing protocols over IP (define filters then only for IP source or destination).
2. Creating an expanded sample access list
   Router (config) # access-list 190 deny tcp 10.10.10.0 0.0.0.255 any eq 23

   where are the tcp protocol identifier over IP, any means any source IP address 10.10.10.0 0.0.0.255 any means the destination host on the network 10.10.10.0/24, while eq 23 is further limitations associated with TCP (in this case the destination port TCP = 23 or telnet)
3. Other examples of the possibilities of formulating filters:
   Router (config) # access-list 190 deny tcp any any eq 23, eq 40
   Router (config) # access-list 190 deny tcp 10.1.1.1 200.200.200.1 0.0.0.1 eq 23 0.0.0.1 eq 100
   Router (config) # access-list 190 deny tcp any host 10.10.10.1
   Router (config) # access-list 190 deny ip any 10.10.10.0 0.0.0.255
   Router (config) # access-list 190 deny icmp any host 10.10.10.1
   Router (config) # access-list 190 deny tcp any host 10.10.10.1
   Router (config) # access-list 190 deny udp any 10.10.10.0 0.0.0.255 eq tftp
   You can check the list of registered command:
   Router # show access-lists
4. As before - to replenish the list of selectively blocking specific traffic should be aware of its completion permission to send other traffic (which is not going to block):
   Router (config) # access-list 190 permit ip any any

5. The assignment of letters to the interface - the same way as before:
   Router(config)#int fa 0/0
   Router (config-if) #ip access-group 190 in
   where n is the filter packets entering the interface out - for outgoing calls.
6. After checking the correct functioning of the selected set of traffic blockade must be removed assignment:
   Router (config-if) #no ip access-group 190 in

Task C: packet filtering - the list of named (named lists)

1. Lists named identify string instead of a number.
2. Create a standard named access list, np.:
   Router(config)#ip access-list standard moja_lista
   Router (config-std-nacl) #deny 10.10.10.1 0.0.0.0
   Router (config-std-nacl) #permit 10.10.10.2 0.0.0.0
   Router (config-std-nacl) #exit
3. Create an extended named access list, np.:
   Router(config)#IP access-list extended moja_lista2
   Router (config-ext-nacl) #deny TCP 10.10.10.1 0.0.0.0 20.10.10.1 0.0.0.0 eq www
   Router (config-ext-nacl) #exit
4. Assign letters to the interface, eg .:
   Router (config-if) # access-group in moja_lista2
5. Check the contents defined list
   Router # sh access-lists
6. After checking the correct functioning of the letter must be removed.