COMPUTER NETWORKS - LABORATORY 072

Subject:
Access Control Lists in Cisco routers

Task A: Datagram Traffic Filtering - standard checklists

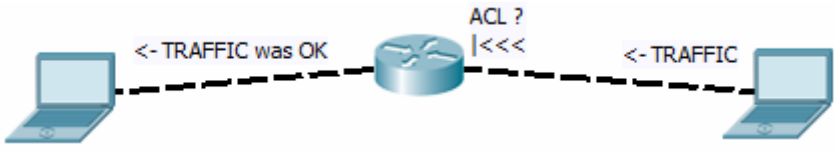
1. Prepare a Cisco router, connecting it with an Ethenet cable in PC - router - PC order. Configure and enable the appropriate router interfaces - defining the two different directly connected IP networks (in accordance with generally known principles), e.g.:

   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 tboth computers must also be properly configured.

   Note: In each computer a default gateway address must be configured, pointing the nearest router interface.
   Make sure the IP routing is running on routers, and enable if needed:
   Router (config) #ip routing

   After configuring the infrastructure check communication between computers (ping, traceroute).



2. The current experiments will involve network traffic blocking between two computers using ACLs (Access Controll Lists) defined in a router. These lists are numbered with unique identifiers. The so-called standard ACL identifiers are between 1 and 99 (the value determines ACL type). Standard lists are simplified allowing only IP address-specific blocking, even without the possibility of distinguishing the sender from the recipient.
   It's possible to specify address ranges, a single address or any host:
   - any - all the addresses,
   - host 10.10.10.0 0.0.0.255 - a network range,
   - host 10.10.10.1 - one host

3. Create first access list, e.g.:
   Router (config) # access-list 90 deny any

   where *90* is the new ID o a list. Nest to *deny* a permit and remark values are possible.

4. Check your list:
   Router # show access-lists

   When restricting traffic on specified IP remember to add a rule permitting other traffic afterwards (the list should be complete):
   Router (config) # access-list 70 deny host 10.10.10.1
   Router (config) # access-list 70 permit any

5. Until now the list is considered just s a resource (it's not active in any sense). The assignment of such list to some flow (such as the interface) activates it:

   Router(config)#int fa 0/0
   Router(config-if)#ip access-group 70 in

   Check your list in the config:
   Router # show run

   Restrict one of your computers with ACL and check connectivity between routers again.

6. Deleting the ACL list:
   Router (config) #no ip access-list standard 70
   Removing the ACL assignment (to the interface):
   Router(config)#int fa 0/0
   Router (config-if) #no ip access-group 70 in

   Note:
   - access control lists are not used only for traffic restrictions (through interfaces), but in many other mechanisms.
   - host scopes 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:
   Router(config)#ip access-list resequence 70 20 10
   where 70 list the ACL number, 20 is a new beginning of sequence numbering and 10 is the 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 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 moja_lista2 in
5. Check the contents defined list
   Router # sh access-lists
6. After checking the correct functioning of the letter must be removed.