Access Control List (Basic)

Safwan Muntasir (Sufi)
Networking Enthusiast

Context

Introduction

In the early days of computer networking, when the internet was in its infancy, security was not a significant concern. Networks were small and tightly controlled. However, as networks grew in size and complexity, the need for security measures became apparent. ACLs (Access Control List) were ***first introduced*** in the ***Multics operating system*** in ***1965***. Multics was a groundbreaking OS that introduced many concepts that are still used today, including ACLs. ACLs were originally used to control to files and directories, but they have since been extended to other types of resources and now a common feature of most OS and network devices. They are used to protect resources from unauthorized access and to enforce security policies.

Why do we need ACLs?

- ***Prevent unauthorized*** users from accessing our ***files and directories***.
- Prevent unauthorized users from accessing our ***networks and applications***.
- Prevent unauthorized users from ***making changes to our systems***.
- Ensure that only authorized users can perform certain actions, such as ***deleting files or creating new users***.
- ACLs can also be used to define traffic to Network Address Translate (NAT), encrypt or filter non-IP protocols such as AppleTalk or IPX.
- ACLs are used in troubleshooting network issues and implementing QoS (Quality of Services) policies.

Key Features

ACL is a set of rules defined for ***controlling network traffic*** and reducing network attacks. ACLs are used to ***filter traffic*** based on the set of rules defined for the incoming or outgoing of the network. ACL set of rules matches ***source IP, destination IP address, IP protocol, ports***.

- ACLs use ***first-match logic***. That means, the set of rules defined are matched serial wise (***sequential order***). Matching stars with the first line, then 2nd , then 3rd and so on.
- The packets are ***matched only until it matches one rule***. Once a rule is matched then no further comparison takes place and that rule will be performed.
- There is an ***implicit denial*** at the end of every ACL, if no condition or rule matches then the packet will be discarded.

Each rules or lines are called ***ACE*** (Access Control Entries) or ACL statements. A group of ACEs or rules referred as ACL.

Types

There are two basic types of ACLs-

1. **Filesystem ACLs:** These work as filters, managing access to directories or files. A filesystem ACL gives the operating system instructions as to the users that are allowed to access the system, as well as the privileges they are entitled to once they are inside.
2. **Networking ACLs:** Manage network access by providing instructions to network switches and routers that specify the types of traffic that are allowed to interface with the network. These ACLs also specify user permissions once inside the network. The network administrator predefines the networking ACL rules. In this way, they function similar to a firewall.

ACLs can also be categorized by the way they identify traffic. These two are widely used ACL types-

1. **Standard ACLs:** Standard ACLs are the ***simpler*** type of ACL. It can only filter traffic ***based on the source IP Address*** (Network, Host or Subnet).
2. **Extended ACLs:** Extended ACLs can filter traffic ***based on*** a wider range of criteria, including ***source IP addresses, destination IP addresses, port numbers, protocol type and ICMP message type***.

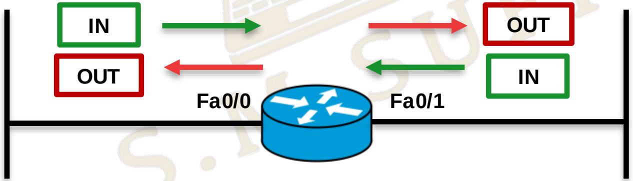
ACLs can also be classified by where they are applied-

i) Interface ACLs **ii) Router ACLs** **iii) Firewall ACLs**

Types

In addition to these two main types of ACLs, there are also a number of other types of ACLs-

- **Reflexive ACLs:** A type of extended ACL that can be used to filter traffic generated by the router itself. In other words, these are used to permit inbound traffic in response to outbound traffic.
- **Dynamic ACLs:** A type of ACL that can be updated dynamically based on information from other sources, such as routing protocols or security devices. They are often used in conjunction with authentication mechanisms such as RADIUS or TACACS+ to control user access.
- **Numbered ACLs**: A type of ACL that is identified by a ***numeric value***.
- **Named ACLs:** A type of ACL that can be given a ***descriptive name***. This can make it easier to manage and maintain ACLs, specially in large network.



According to direction, ACL can be divided into two types-

1. **Inbound:** Any packet ***coming to the router***, before the router makes its forwarding (routing) decisions.
2. **Outbound:** Any packet ***going out of the router***, after the router makes its forwarding decisions and has determined the exit interface.

*****Only one inbound and one outbound ACL can be defined in an interface.**

Wild Card Mask

A wildcard mask is a ***bitmask*** used to specify range of IP addresses. It tells the IOS which portion of the ***bits to match or ignore***. A wildcard mask is a ***32-bit bitmask***, similar to a subnet mask, but it works in the opposite way.

- **Decimal 0 –** The router ***must compare*** this octet as normal.
- **Decimal 255 –** The router ***ignores*** this octet, considering it to already match.
- To match a specific but in an IP address, the corresponding bit in the wildcard mask must be zero.
- To ignore a specific bit in an IP address, the corresponding bit in the wildcard mask must be one.

Notice the ACL equivalents-

- The source/wildcard of ***0.0.0.0/255.255.255.255*** means ***any (host).***
- The source/wildcard of ***10.1.1.2/0.0.0.0*** is the ***same*** as ***host 10.1.1.2***.

Wildcard mask is also used for ***network summarization***. Network summarization or route aggregation is the act of taking two or more IP networks and using a single IP network to represent them all.

Wild Card Mask

Subnet Mask	CIDR	Wildcard Mask
0.0.0.0	**/0**	**255.255.255.255**
255.0.0.0	**/8**	**0.255.255.255**
255.255.0.0	**/16**	**0.0.255.255**
255.255.255.0	**/24**	**0.0.0.255**
255.255.255.128	/25	0.0.0.127
255.255.255.192	/26	0.0.0.63
255.255.255.224	/27	0.0.0.31
255.255.255.240	/28	0.0.0.15
255.255.255.248	/29	0.0.0.7
255.255.255.252	/30	0.0.0.3
255.255.255.254	/31	0.0.0.1
255.255.255.255	**/32**	**0.0.0.0**

Standard ACLs

- Can be named of numbered. Range of Standard ACL number is ***(1-99)*** or ***(1300-1999)***.
- ACL must be applied on the transit router and interface.
- Standard ACLs should be placed ***near to the destination*** of the packets (not always).
- Standard ACLs can only filter traffic based on the ***source IP Address*** (Network, Host or Subnet).
- There is an ***implicit denial*** at the end of every ACL, if no condition or rule matches then the packet will be discarded.
- Commands for ***numbered*** standard ACLs-

  'Router(config)# access-list <acl no> <permit/deny> <source address> <source wildcard mask>'

  'Router(config)# access-list <acl no> <permit/deny> any'
- Commands for ***numbered*** standard ACL on single host-

  'Router(config)# access-list <acl no> <permit/deny> <source host address>'

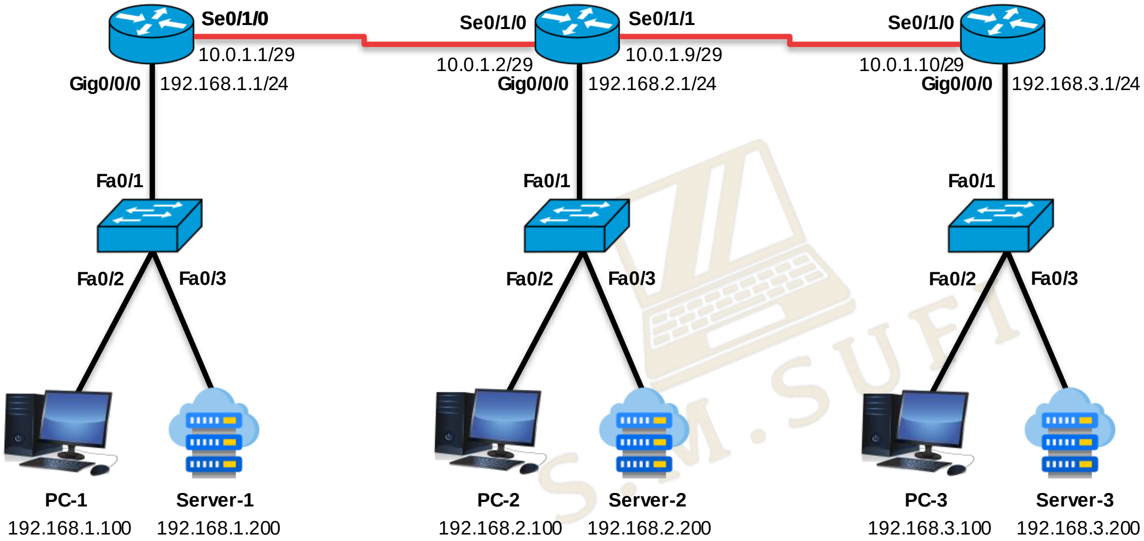
  'Router(config)# access-list <acl no> <permit/deny> host <source host address>'

  'Router(config)# access-list <acl no> <permit/deny> <source host address> 0.0.0.0'
- After creating ACLs, it is time to ***apply*** the ACLs on the ***right interface and direction*** using following commands-

  'Router(config)# interface <interface name>'

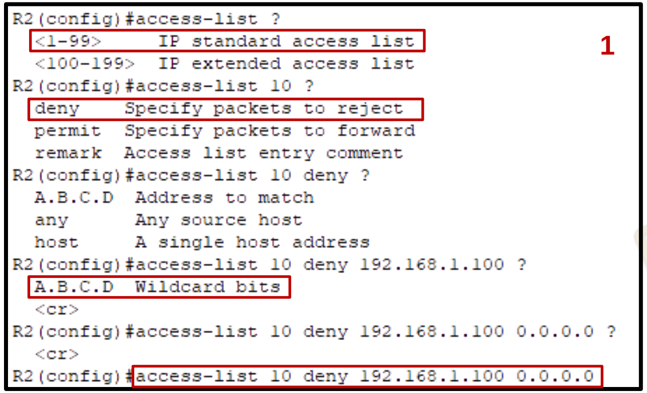
  'Router(config-if)# ip access-group <acl no> <in/out>'

Standard ACLs

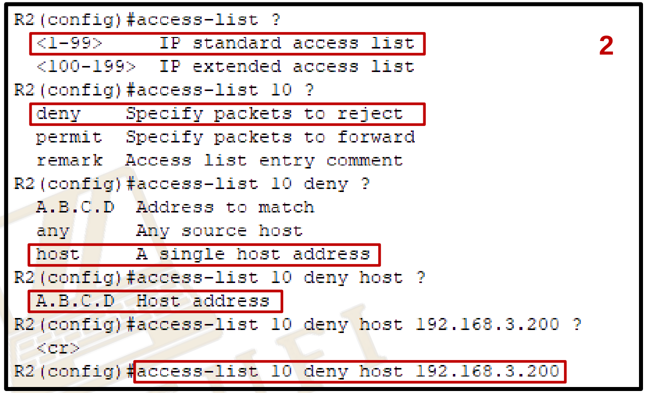


1. ***Host 192.168.1.100*** cannot ping hosts and servers of 192.168.2.0 network
2. ***Server 192.168.3.200*** cannot ping hosts and servers of 192.168.2.0 network
3. Other host/network can ping independently

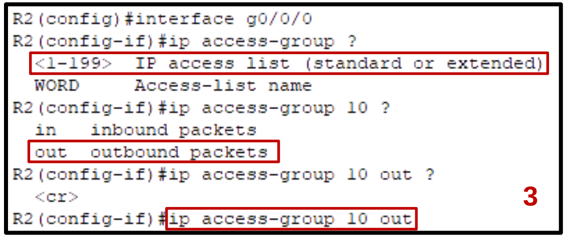
Standard ACLs



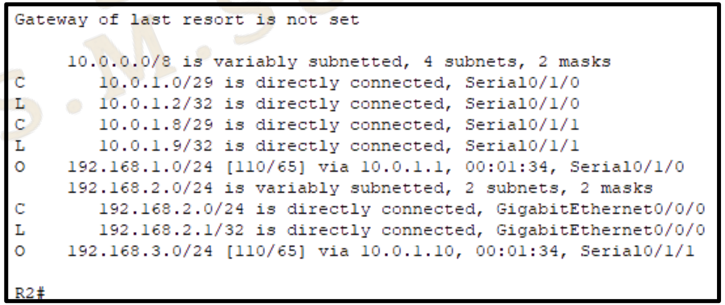
```
R2(config)#access-list ?
  <1-99>       IP standard access list
  <100-199>  IP extended access list
R2(config)#access-list 10 ?
  deny     Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R2(config)#access-list 10 deny ?
  A.B.C.D  Address to match
  any        Any source host
  host       A single host address
R2(config)#access-list 10 deny 192.168.1.100 ?
  A.B.C.D  Wildcard bits
  <cr>
R2(config)#access-list 10 deny 192.168.1.100 0.0.0.0 ?
  <cr>
R2(config)#access-list 10 deny 192.168.1.100 0.0.0.0
```



```
R2(config)#access-list ?
  <1-99>       IP standard access list
  <100-199>  IP extended access list
R2(config)#access-list 10 ?
  deny     Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R2(config)#access-list 10 deny ?
  A.B.C.D  Address to match
  any        Any source host
  host       A single host address
R2(config)#access-list 10 deny host ?
  A.B.C.D  Host address
R2(config)#access-list 10 deny host 192.168.3.200 ?
  <cr>
R2(config)#access-list 10 deny host 192.168.3.200
```



```
R2(config)#interface g0/0/0
R2(config-if)#ip access-group ?
  <1-199>  IP access list (standard or extended)
  WORD       Access-list name
R2(config-if)#ip access-group 10 ?
  in    inbound packets
  out   outbound packets
R2(config-if)#ip access-group 10 out ?
  <cr>
R2(config-if)#ip access-group 10 out
```

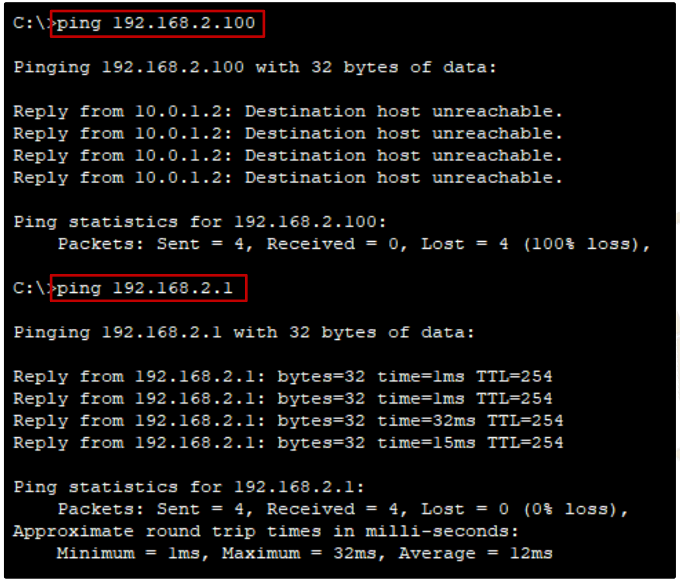


```
Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.0.1.0/29 is directly connected, Serial0/1/0
L       10.0.1.2/32 is directly connected, Serial0/1/0
C       10.0.1.8/29 is directly connected, Serial0/1/1
L       10.0.1.9/32 is directly connected, Serial0/1/1
O    192.168.1.0/24 [110/65] via 10.0.1.1, 00:01:34, Serial0/1/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/0/0
L       192.168.2.1/32 is directly connected, GigabitEthernet0/0/0
O    192.168.3.0/24 [110/65] via 10.0.1.10, 00:01:34, Serial0/1/1

R2#
```

1. ***Host 192.168.1.100*** cannot ping hosts and servers of 192.168.2.0 network
2. ***Server 192.168.3.200*** cannot ping hosts and servers of 192.168.2.0 network
3. Other host/network can ping independently

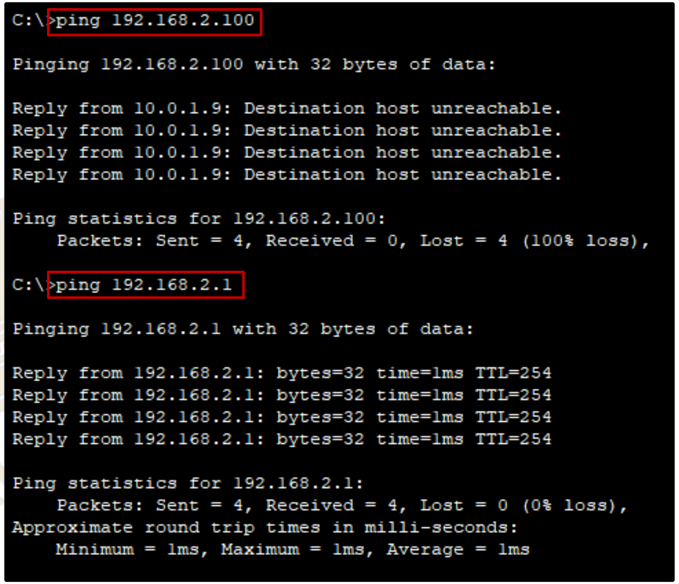
There is OSPF route from each network to every other network.

Standard ACLs



```
C:\>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.

Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=32ms TTL=254
Reply from 192.168.2.1: bytes=32 time=15ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 32ms, Average = 12ms
```

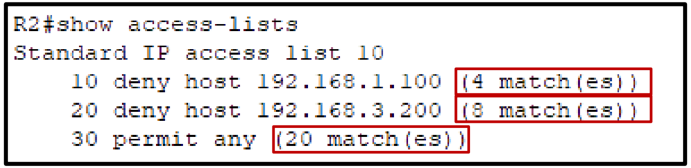
From PC-1 192.168.1.100



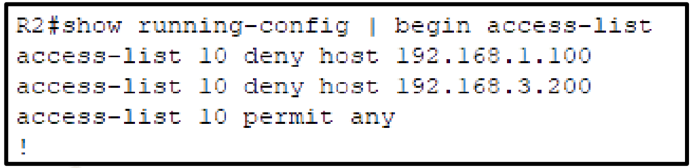
```
C:\>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Reply from 10.0.1.9: Destination host unreachable.
Reply from 10.0.1.9: Destination host unreachable.
Reply from 10.0.1.9: Destination host unreachable.
Reply from 10.0.1.9: Destination host unreachable.

Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

From Server-3 192.168.3.200

Standard ACLs



```
R2#show access-lists
Standard IP access list 10
    10 deny host 192.168.1.100 (4 match(es))
    20 deny host 192.168.3.200 (8 match(es))
    30 permit any (20 match(es))
```



```
R2#show running-config | begin access-list
access-list 10 deny host 192.168.1.100
access-list 10 deny host 192.168.3.200
access-list 10 permit any
!
```

- Named ACLs are ***case-sensitive***. Commands for ***named*** standard ACLs-

  'Router(config)# ip access-list <standard/extended> <acl name>

  'Router(config-std-nacl)# <permit/deny> <source address> <source wildcard mask>'

  'Router(config-std-nacl)# <permit/deny> any'

- Commands for ***named*** standard ACL on single host-

  'Router(config-std-nacl)# <permit/deny> <source host address>'

  'Router(config-std-nacl)# <permit/deny> host <source host address>'

  'Router(config-std-nacl)# <permit/deny> <source host address> 0.0.0.0'

- After creating ACLs, it is time to ***apply*** the ACLs on the ***right interface and direction*** using following commands-

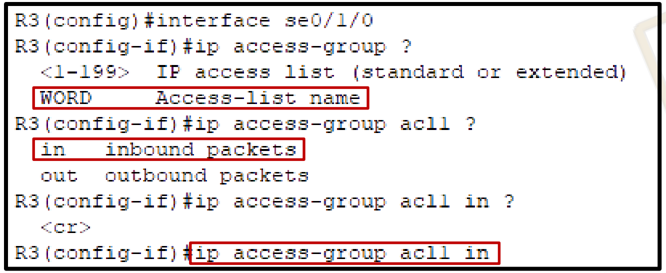
  'Router(config)# interface <interface name>'

  'Router(config-if)# ip access-group <acl name> <in/out>'

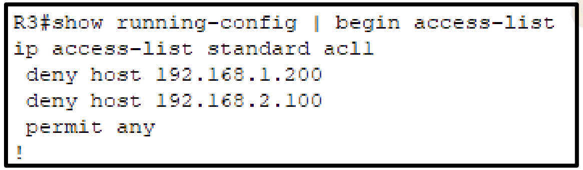
Standard ACLs

1. ***Server 192.168.1.200*** cannot ping 192.168.3.0 network
2. ***Host 192.168.2.100*** cannot ping 192.168.3.0 network
3. Other host/network can ping independently

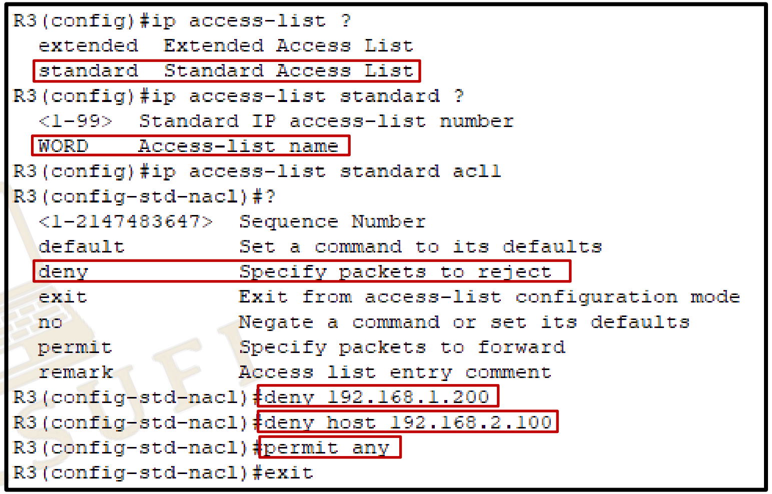
There is OSPF route from each network to every other network.



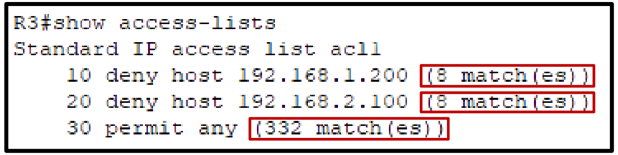
```
R3(config)#interface se0/1/0
R3(config-if)#ip access-group ?
  <1-199>  IP access list (standard or extended)
  WORD     Access-list name
R3(config-if)#ip access-group acl1 ?
  in   inbound packets
  out  outbound packets
R3(config-if)#ip access-group acl1 in ?
  <cr>
R3(config-if)#ip access-group acl1 in
```



```
R3#show running-config | begin access-list
ip access-list standard acl1
 deny host 192.168.1.200
 deny host 192.168.2.100
 permit any
!
```

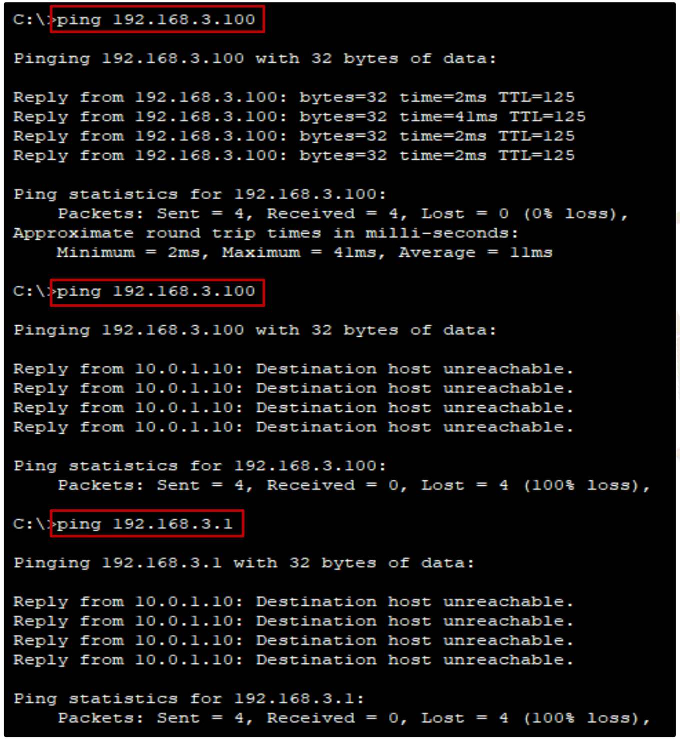


```
R3(config)#ip access-list ?
  extended  Extended Access List
  standard  Standard Access List
R3(config)#ip access-list standard ?
  <1-99>  Standard IP access-list number
  WORD    Access-list name
R3(config)#ip access-list standard acl1
R3(config-std-nacl)#?
  <1-2147483647>  Sequence Number
  default         Set a command to its defaults
  deny            Specify packets to reject
  exit            Exit from access-list configuration mode
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
R3(config-std-nacl)#deny 192.168.1.200
R3(config-std-nacl)#deny host 192.168.2.100
R3(config-std-nacl)#permit any
R3(config-std-nacl)#exit
```



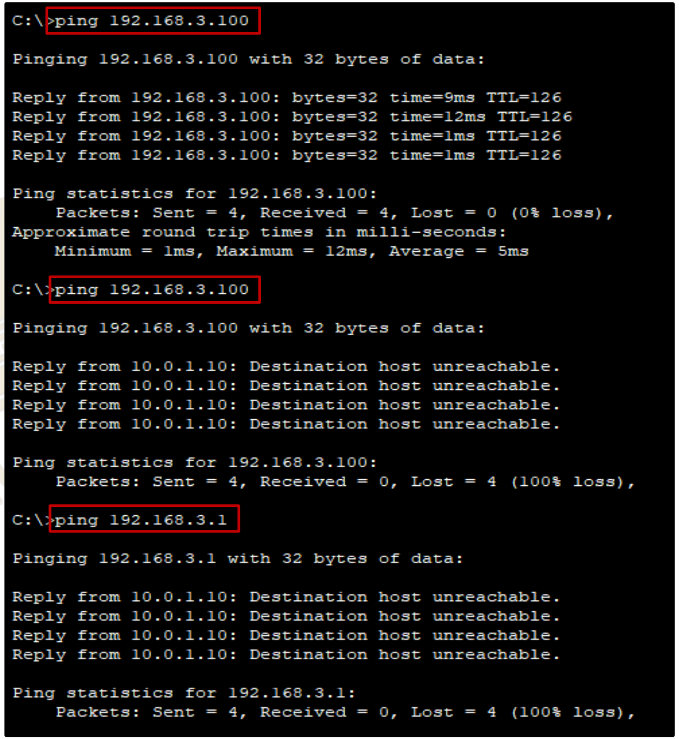
```
R3#show access-lists
Standard IP access list acl1
    10 deny host 192.168.1.200 (8 match(es))
    20 deny host 192.168.2.100 (8 match(es))
    30 permit any (332 match(es))
```

Standard ACLs



```
C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 192.168.3.100: bytes=32 time=2ms TTL=125
Reply from 192.168.3.100: bytes=32 time=41ms TTL=125
Reply from 192.168.3.100: bytes=32 time=2ms TTL=125
Reply from 192.168.3.100: bytes=32 time=2ms TTL=125

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 41ms, Average = 11ms

C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

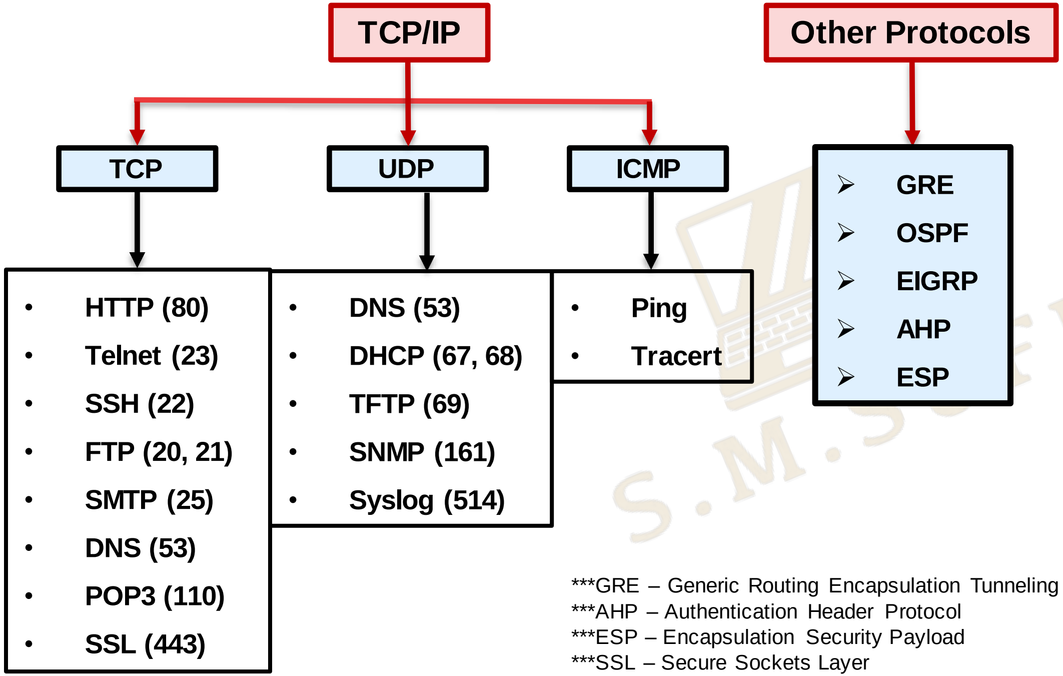
From Server-1 192.168.1.200



```
C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 192.168.3.100: bytes=32 time=9ms TTL=126
Reply from 192.168.3.100: bytes=32 time=12ms TTL=126
Reply from 192.168.3.100: bytes=32 time=1ms TTL=126
Reply from 192.168.3.100: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 12ms, Average = 5ms

C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

From Host-2 192.168.2.100

Protocols



***GRE – Generic Routing Encapsulation Tunneling
***AHP – Authentication Header Protocol
***ESP – Encapsulation Security Payload
***SSL – Secure Sockets Layer

Services	ACL Keyword	Port Numbers
FTP Data	*‘ftp-data’*	20
FTP Control	*‘ftp’*	21
SSH	-	22
Telnet	*‘telnet’*	23
SMTP	*‘smtp’*	25
DNS	*‘domain’*	53
DHCP Server	*‘bootps’*	67
DHCP Client	*‘bootpc’*	68
TFTP	*‘tftp’*	69
HTTP	*‘www’*	80
POP3	*‘pop3’*	110
SNMP	*‘snmp’*	161
SSL	-	443
Syslog	-	514

Extended ACLs

- Can be named of numbered. Range of Extended ACL number is ***(100-199)*** or ***(2000-2699)***.
- Extended ACLs should be placed ***near to the source*** of the packets to save some ***bandwidth*** (not always).
- Extended ACLs can filter traffic ***based on the source IP*** (Network, Host or Subnet), ***destination IP, port no, protocols*** and ***services***.
- There is an ***implicit denial*** at the end of every ACL, if no condition or rule matches then the packet will be discarded.
- Extended ACL ***operators*** are used to match TCP and UDP Port Numbers. Operators are optional to use, ***not mandatory***.
- Commands for ***numbered*** extended ACLs-

```
'Router(config)# access-list <acl no> <permit/deny> <protocol> <source> <wcm> <destination> <wcm> <operator> <service/port>'
'Router(config)# access-list <acl no> <permit/deny> <ip> any any'
```

- Commands for ***numbered*** extended ACL on single host-

```
'Router(config)# access-list <acl no> <permit/deny> <protocol> <source>…..'
'Router(config)# access-list <acl no> <permit/deny> <protocol> host <source>…..'
'Router(config)# access-list <acl no> <permit/deny> <protocol> <source> 0.0.0.0 …..'
```

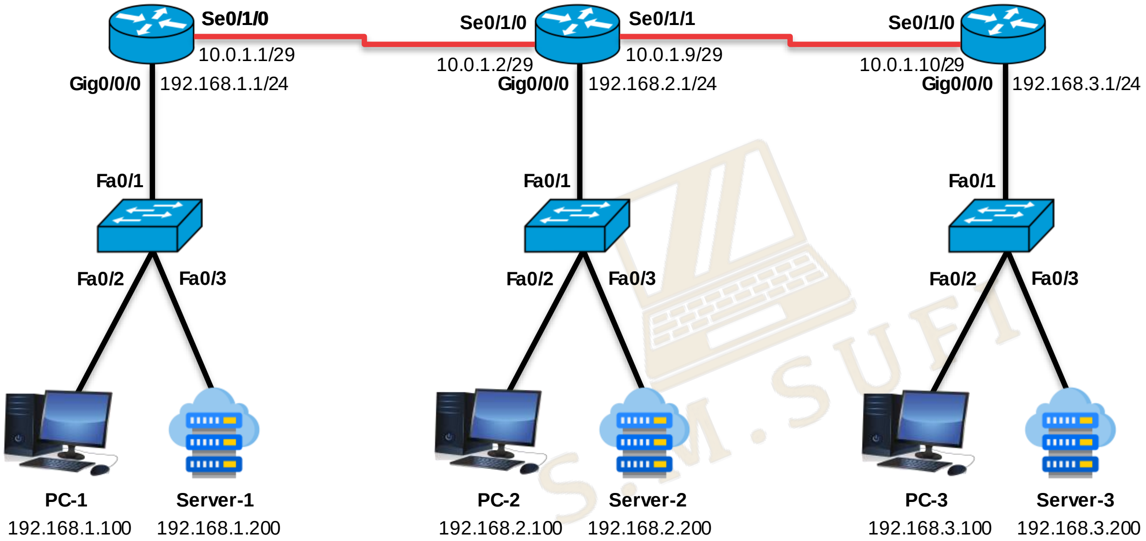
- ***Apply*** the ACLs on the ***right interface and direction*** using following commands-

```
'Router(config)# interface <interface name>'
'Router(config-if)# ip access-group <acl no> <in/out>'
```

Most used ACL Operators:

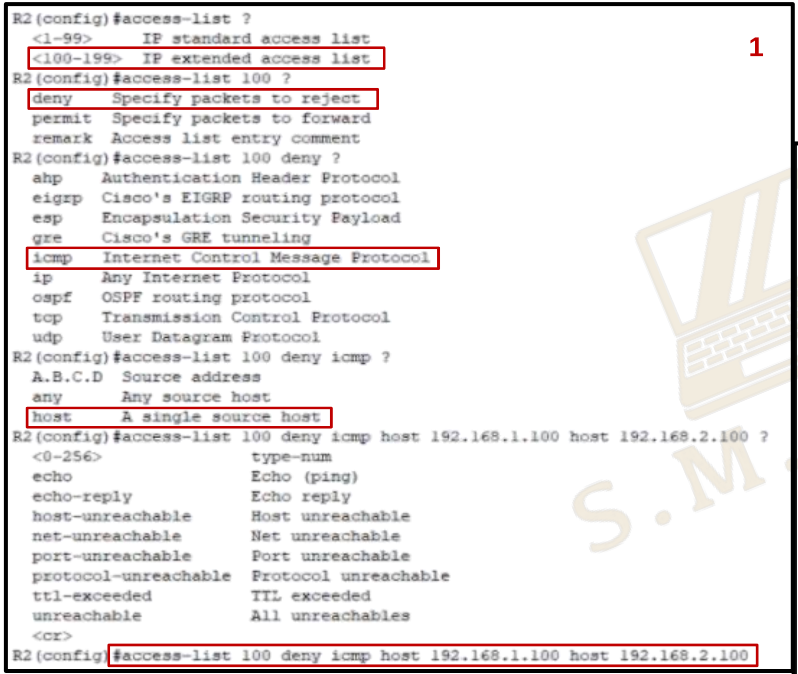
- ***'eq'*** **– Equal To**
- ***'neq'*** **– Not Equal To**
- ***'lt'*** **– Less Than**
- ***'gt'*** **– Greater Than**
- ***'range'*** **– Range of Port Number**

Extended ACLs



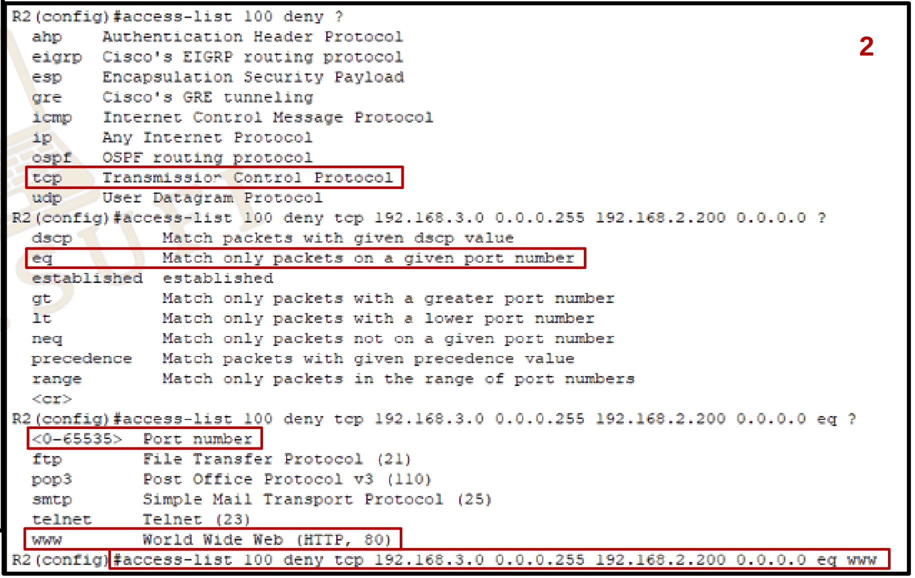
1. ***Host 192.168.1.100*** cannot ping hosts of 192.168.2.0 network
2. ***Network 192.168.3.200*** cannot access web servers of 192.168.2.0 network
3. Other host/network can reach independently

Extended ACLs



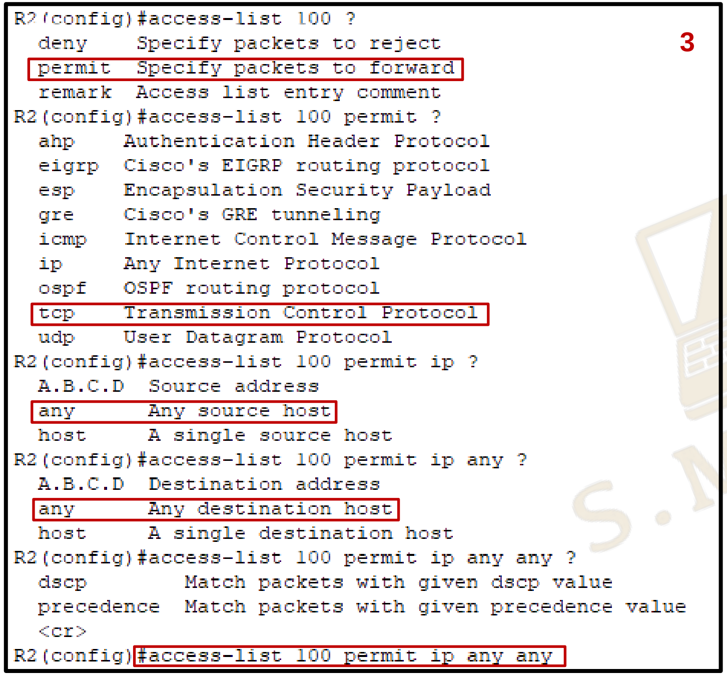
```
R2(config)#access-list ?
  <1-99>      IP standard access list
  <100-199>   IP extended access list
R2(config)#access-list 100 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R2(config)#access-list 100 deny ?
  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
R2(config)#access-list 100 deny icmp ?
  A.B.C.D  Source address
  any      Any source host
  host     A single source host
R2(config)#access-list 100 deny icmp host 192.168.1.100 host 192.168.2.100 ?
  <0-256>               type-num
  echo                  Echo (ping)
  echo-reply            Echo reply
  host-unreachable      Host unreachable
  net-unreachable       Net unreachable
  port-unreachable      Port unreachable
  protocol-unreachable  Protocol unreachable
  ttl-exceeded          TTL exceeded
  unreachable           All unreachables
  <cr>
R2(config)#access-list 100 deny icmp host 192.168.1.100 host 192.168.2.100
```

1. ***Host 192.168.1.100*** cannot ping hosts of 192.168.2.0 network
2. ***Network 192.168.3.200*** cannot access web http of 192.168.2.0 network
3. Other host/network can reach independently



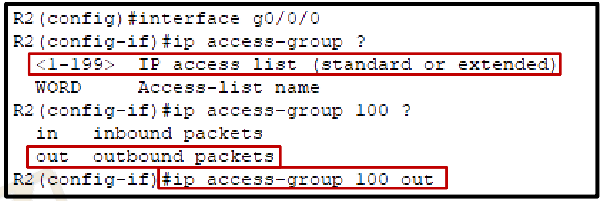
```
R2(config)#access-list 100 deny ?
  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
R2(config)#access-list 100 deny tcp 192.168.3.0 0.0.0.255 192.168.2.200 0.0.0.0 ?
  dscp         Match packets with given dscp value
  eq           Match only packets on a given port number
  established  established
  gt           Match only packets with a greater port number
  lt           Match only packets with a lower port number
  neq          Match only packets not on a given port number
  precedence   Match packets with given precedence value
  range        Match only packets in the range of port numbers
  <cr>
R2(config)#access-list 100 deny tcp 192.168.3.0 0.0.0.255 192.168.2.200 0.0.0.0 eq ?
  <0-65535>  Port number
  ftp        File Transfer Protocol (21)
  pop3       Post Office Protocol v3 (110)
  smtp       Simple Mail Transport Protocol (25)
  telnet     Telnet (23)
  www        World Wide Web (HTTP, 80)
R2(config)#access-list 100 deny tcp 192.168.3.0 0.0.0.255 192.168.2.200 0.0.0.0 eq www
```

Extended ACLs

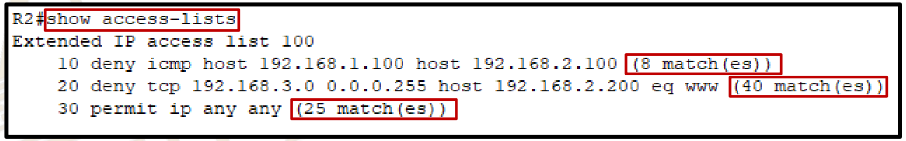


3

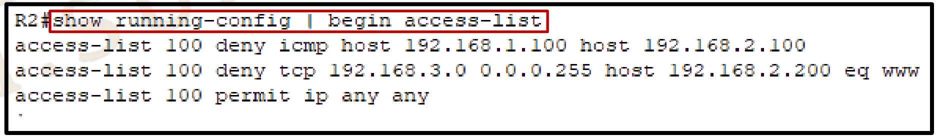
```
R2(config)#access-list 100 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R2(config)#access-list 100 permit ?
  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
R2(config)#access-list 100 permit ip ?
  A.B.C.D  Source address
  any      Any source host
  host     A single source host
R2(config)#access-list 100 permit ip any ?
  A.B.C.D  Destination address
  any      Any destination host
  host     A single destination host
R2(config)#access-list 100 permit ip any any ?
  dscp        Match packets with given dscp value
  precedence  Match packets with given precedence value
  <cr>
R2(config)#access-list 100 permit ip any any
```



```
R2(config)#interface g0/0/0
R2(config-if)#ip access-group ?
  <1-199>  IP access list (standard or extended)
  WORD     Access-list name
R2(config-if)#ip access-group 100 ?
  in   inbound packets
  out  outbound packets
R2(config-if)#ip access-group 100 out
```



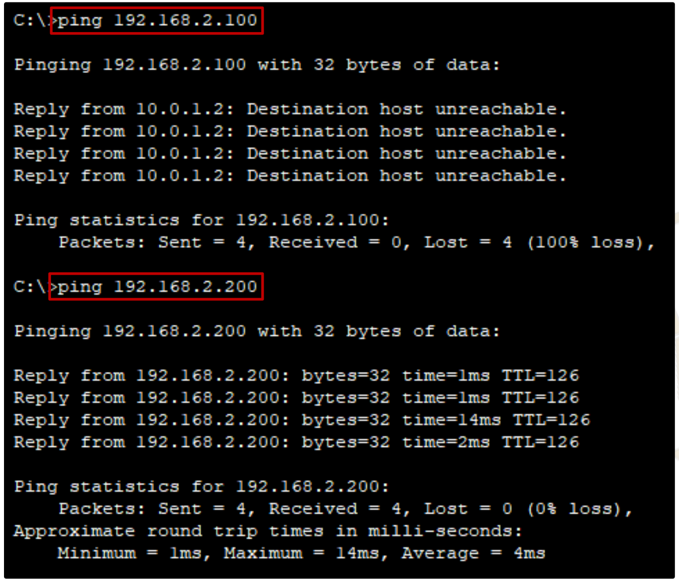
```
R2#show access-lists
Extended IP access list 100
    10 deny icmp host 192.168.1.100 host 192.168.2.100 (8 match(es))
    20 deny tcp 192.168.3.0 0.0.0.255 host 192.168.2.200 eq www (40 match(es))
    30 permit ip any any (25 match(es))
```



```
R2#show running-config | begin access-list
access-list 100 deny icmp host 192.168.1.100 host 192.168.2.100
access-list 100 deny tcp 192.168.3.0 0.0.0.255 host 192.168.2.200 eq www
access-list 100 permit ip any any
```

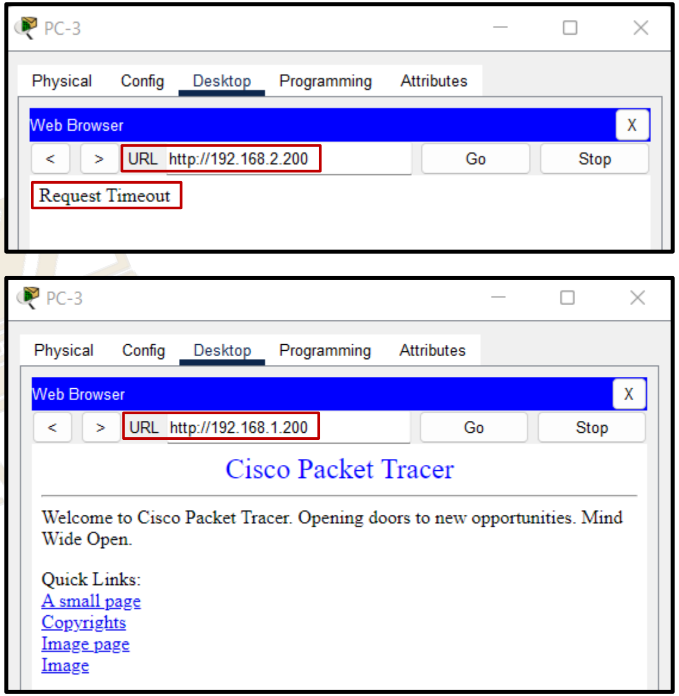
*****Extended ACLs could be applied in Router R1 and R3 also, not on Router R2 to save Bandwidth.**

Extended ACLs



```
C:\>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.
Reply from 10.0.1.2: Destination host unreachable.

Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.200

Pinging 192.168.2.200 with 32 bytes of data:

Reply from 192.168.2.200: bytes=32 time=1ms TTL=126
Reply from 192.168.2.200: bytes=32 time=1ms TTL=126
Reply from 192.168.2.200: bytes=32 time=14ms TTL=126
Reply from 192.168.2.200: bytes=32 time=2ms TTL=126

Ping statistics for 192.168.2.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 14ms, Average = 4ms
```

From PC-1 192.168.1.100



URL http://192.168.2.200

Request Timeout

URL http://192.168.1.200

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

From Server-3 192.168.3.200

Extended ACLs

- Named ACLs are ***case-sensitive***. Commands for ***named*** extended ACLs-

  'Router(config)# ip access-list <extended> <acl name>

  'Router(config-ext-nacl)# <permit/deny> <protocol> <source> <wcm> <destination> <wcm> <operator> <service/port>'

  'Router(config-ext-nacl)# <permit/deny> <ip> any any'

- Commands for ***named*** extended ACL on single host-

  'Router(config-ext-acl)# access-list <permit/deny> <protocol> <source> <wcm> <destination> <wcm> <operator> <service/port>'

  'Router(config-ext-acl)# access-list <permit/deny> <protocol> host <source> <wcm> <destination> <wcm> <operator> <service/port>'

  'Router(config-ext-acl)# access-list <permit/deny> <protocol> <source> 0.0.0.0 <destination> <wcm> <operator> <service/port>'

- After creating ACLs, it is time to ***apply*** the ACLs on the ***right interface and direction*** using following commands-

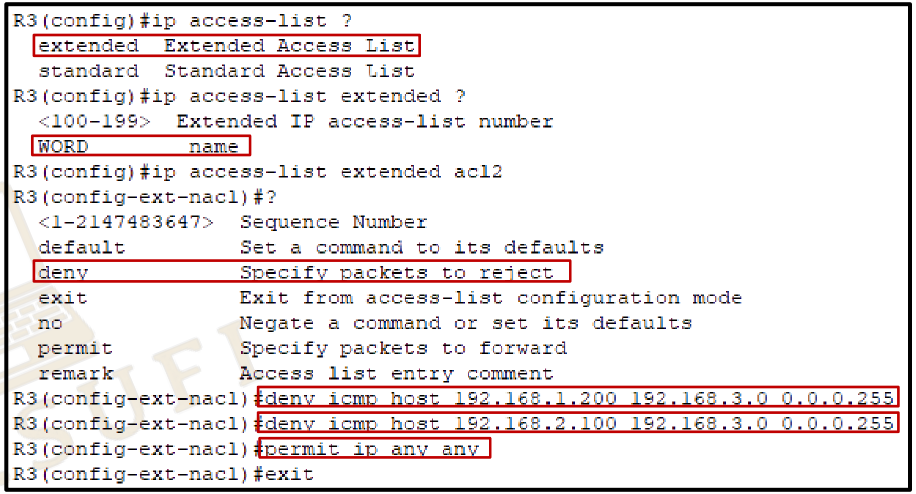
  'Router(config)# interface <interface name>'

  'Router(config-if)# ip access-group <acl name> <in/out>'

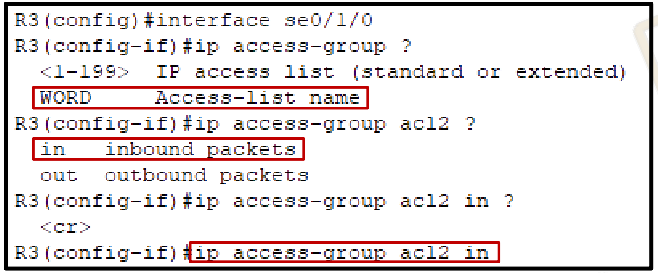
Extended ACLs

1. ***Server 192.168.1.200*** cannot ping 192.168.3.0 network
2. ***Host 192.168.2.100*** cannot ping 192.168.3.0 network
3. Other host/network can ping independently

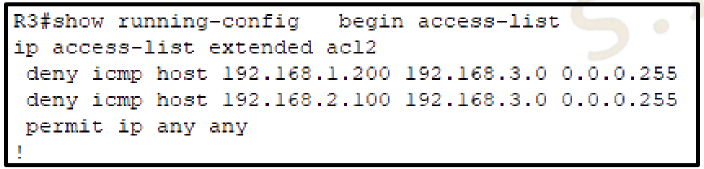
There is OSPF route from each network to every other network.



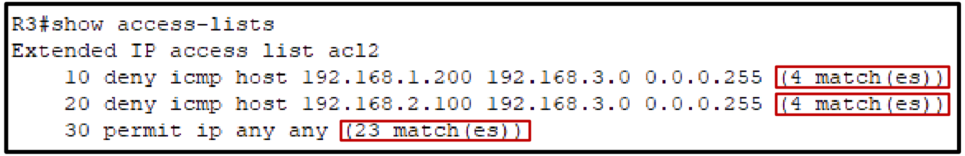
```
R3(config)#ip access-list ?
  extended  Extended Access List
  standard  Standard Access List
R3(config)#ip access-list extended ?
  <100-199>  Extended IP access-list number
  WORD          name
R3(config)#ip access-list extended acl2
R3(config-ext-nacl)#?
  <1-2147483647>  Sequence Number
  default         Set a command to its defaults
  deny            Specify packets to reject
  exit            Exit from access-list configuration mode
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
R3(config-ext-nacl)#deny icmp host 192.168.1.200 192.168.3.0 0.0.0.255
R3(config-ext-nacl)#deny icmp host 192.168.2.100 192.168.3.0 0.0.0.255
R3(config-ext-nacl)#permit ip any any
R3(config-ext-nacl)#exit
```



```
R3(config)#interface se0/1/0
R3(config-if)#ip access-group ?
  <1-199>  IP access list (standard or extended)
  WORD     Access-list name
R3(config-if)#ip access-group acl2 ?
  in    inbound packets
  out   outbound packets
R3(config-if)#ip access-group acl2 in ?
  <cr>
R3(config-if)#ip access-group acl2 in
```

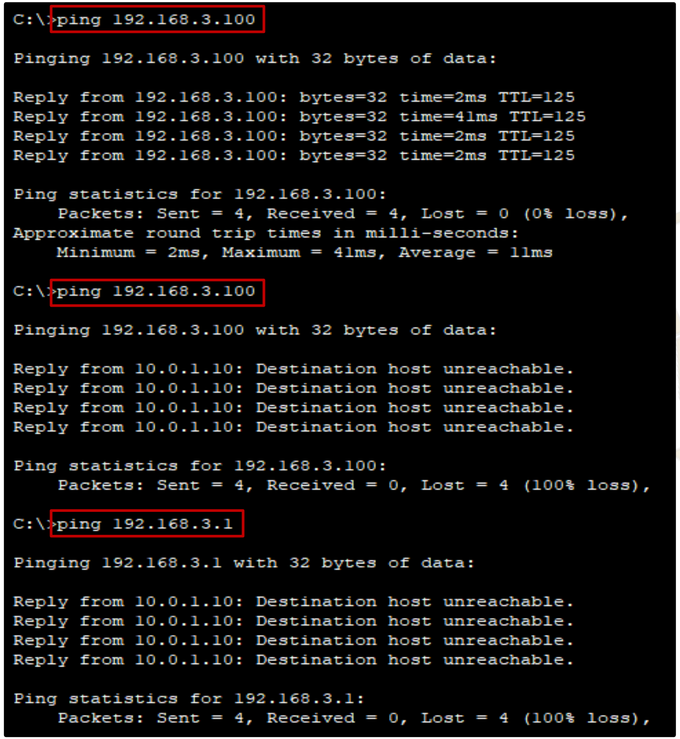


```
R3#show running-config   begin access-list
ip access-list extended acl2
 deny icmp host 192.168.1.200 192.168.3.0 0.0.0.255
 deny icmp host 192.168.2.100 192.168.3.0 0.0.0.255
 permit ip any any
!
```



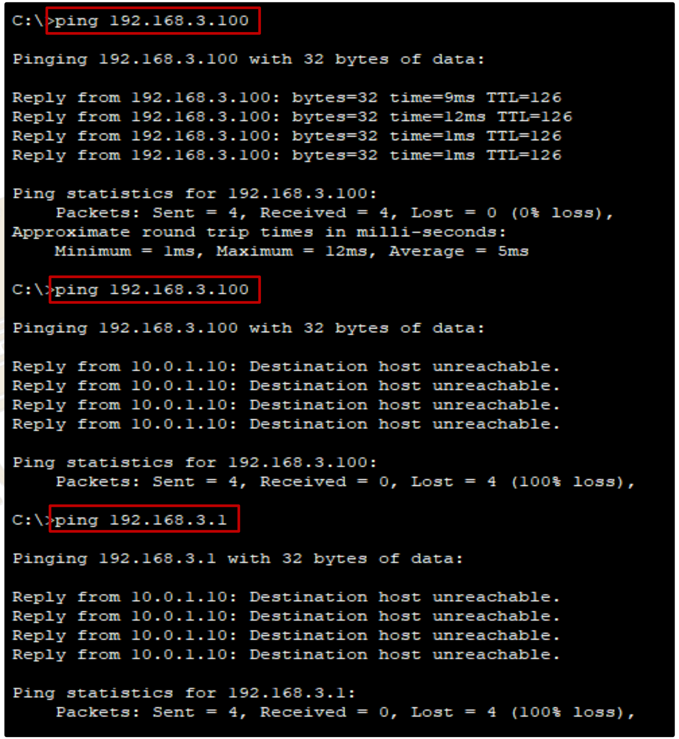
```
R3#show access-lists
Extended IP access list acl2
    10 deny icmp host 192.168.1.200 192.168.3.0 0.0.0.255 (4 match(es))
    20 deny icmp host 192.168.2.100 192.168.3.0 0.0.0.255 (4 match(es))
    30 permit ip any any (23 match(es))
```

Extended ACLs



```
C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 192.168.3.100: bytes=32 time=2ms TTL=125
Reply from 192.168.3.100: bytes=32 time=41ms TTL=125
Reply from 192.168.3.100: bytes=32 time=2ms TTL=125
Reply from 192.168.3.100: bytes=32 time=2ms TTL=125

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 41ms, Average = 11ms

C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

From Server-1 192.168.1.200



```
C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 192.168.3.100: bytes=32 time=9ms TTL=126
Reply from 192.168.3.100: bytes=32 time=12ms TTL=126
Reply from 192.168.3.100: bytes=32 time=1ms TTL=126
Reply from 192.168.3.100: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 12ms, Average = 5ms

C:\>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.
Reply from 10.0.1.10: Destination host unreachable.

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

From Host-2 192.168.2.100

ACLs Editing

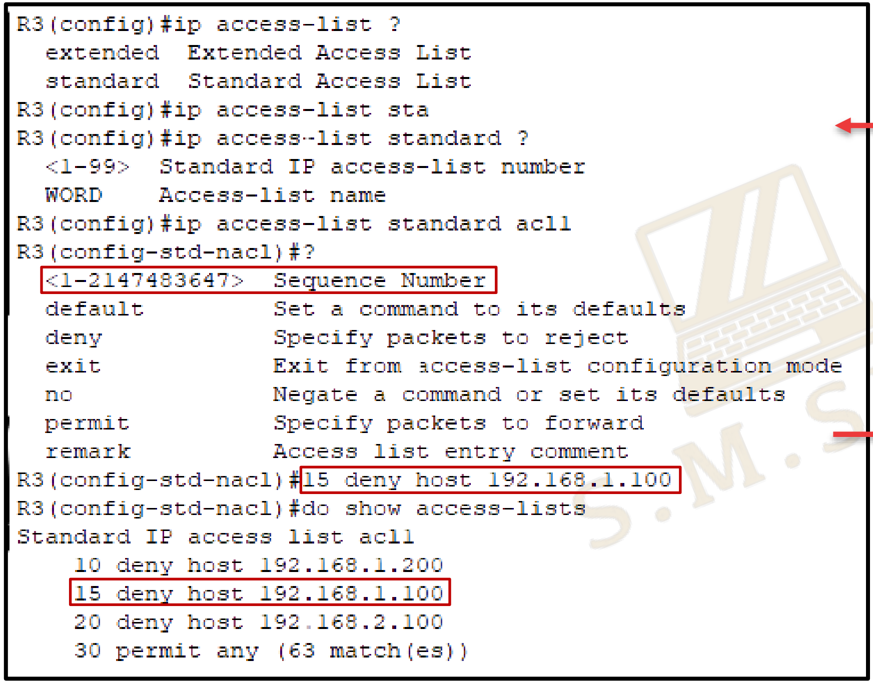
- In ***old version of IOS***, numbered ACLs (Standard or Extended) ***could not be edited***. To simply delete a line from the ACL, the user had to delete the entire ACL and then reconfigure it.
- ***By default***, new ACL statements will be ***added in the last in sequence***. If we wanted to add line in between, we had to copy entire ACL into a notepad, re-arrange the ACEs, and the after deleting old ACL we had to create a new ACL with updated sequenced ACEs.
- However, the ability to edit numbered ACLs was introduced in ***Cisco IOS version 12.3(2)T***, released in 2006. So, all modern versions of Cisco IOS support editing numbered ACLs.
- To verify the version of Cisco IOS running, use the command- ***'Router# show version'***
- It is ***recommended*** to use ***Named ACLs*** instead of numbered ACLs. Because Named ACLs are more flexible and easier to remember and manage than numbered ACLs. They can be applied to ***multiple interfaces*** and they can be ***used in firewall policies*** and ***routing protocols*** where numbered ACLs have limitations using in firewalls. Named ACLs can be edited in all the existing Cisco IOS versions.
- Commands for Inserting an ACE in an existing ACL-

```
'Router(config)# ip access-list <standard/extended> <acl name>'
'Router(config-std-nacl)# <sequence number> <permit/deny> <source>'
```

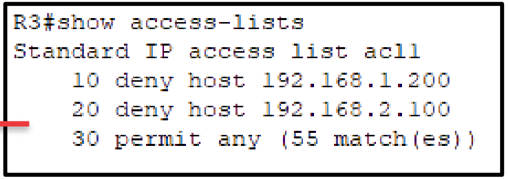
- Commands for Deleting an ACE in an existing ACL-

```
'Router(config)# ip access-list <standard/extended> <acl name>'
'Router(config-std-nacl)# no <sequence number>'
```

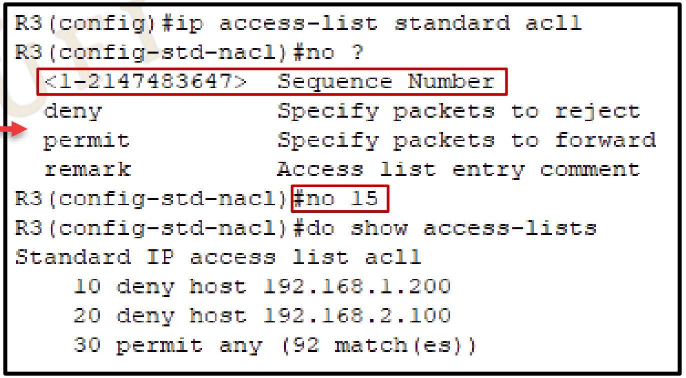
ACLs Editing



```
R3(config)#ip access-list ?
  extended  Extended Access List
  standard  Standard Access List
R3(config)#ip access-list sta
R3(config)#ip access-list standard ?
  <1-99>  Standard IP access-list number
  WORD    Access-list name
R3(config)#ip access-list standard acl1
R3(config-std-nacl)#?
  <1-2147483647>  Sequence Number
  default         Set a command to its defaults
  deny            Specify packets to reject
  exit            Exit from access-list configuration mode
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
R3(config-std-nacl)#15 deny host 192.168.1.100
R3(config-std-nacl)#do show access-lists
Standard IP access list acl1
    10 deny host 192.168.1.200
    15 deny host 192.168.1.100
    20 deny host 192.168.2.100
    30 permit any (63 match(es))
```



```
R3#show access-lists
Standard IP access list acl1
    10 deny host 192.168.1.200
    20 deny host 192.168.2.100
    30 permit any (55 match(es))
```



```
R3(config)#ip access-list standard acl1
R3(config-std-nacl)#no ?
  <1-2147483647>  Sequence Number
  deny            Specify packets to reject
  permit          Specify packets to forward
  remark          Access list entry comment
R3(config-std-nacl)#no 15
R3(config-std-nacl)#do show access-lists
Standard IP access list acl1
    10 deny host 192.168.1.200
    20 deny host 192.168.2.100
    30 permit any (92 match(es))
```

ACLs Editing

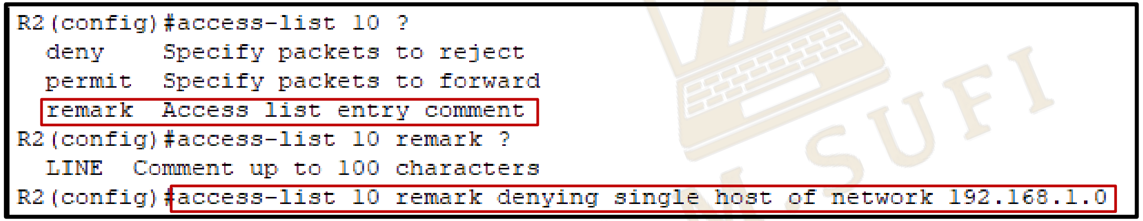
- Commands for Remarking the ACEs in an existing named ACL-

  'Router(config)# ip access-list <standard/extended> <acl name>'

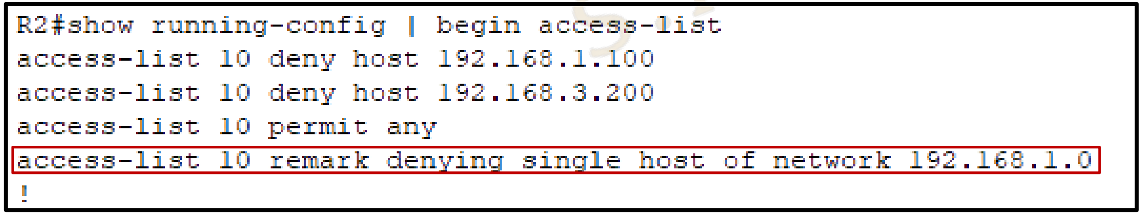
  'Router(config-std-nacl)# <remark> <remarking-string>'

- Commands for Remarking the ACEs in an existing numbered ACL-

  'Router(config)# access-list <acl no> remark <remarking-string>'



```
R2(config)#access-list 10 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R2(config)#access-list 10 remark ?
  LINE  Comment up to 100 characters
R2(config)#access-list 10 remark denying single host of network 192.168.1.0
```



```
R2#show running-config | begin access-list
access-list 10 deny host 192.168.1.100
access-list 10 deny host 192.168.3.200
access-list 10 permit any
access-list 10 remark denying single host of network 192.168.1.0
!
```

Thank You

Feel free to reach out to me for any **suggestions** or **feedback** via **LinkedIn** or **Mail**

www.github.com/smsufi

www.linkedin.com/in/smsufi

safwanm.cse@gmail.com

References

- https://en.wikipedia.org/wiki/Access-control_list
- https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html#anc0
- https://www.geeksforgeeks.org/access-lists-acl/
- https://www.youtube.com/watch?v=ZmcFK_qJRqE&list=PLJqb_j53o7BhgMyY_SlaHDCZtjaQlDYEs
- https://www.cbtnuggets.com/blog/technology/networking/networking-basics-what-are-wildcard-masks-and-how-do-they-work
- https://www.techtarget.com/searchnetworking/definition/access-control-list-ACL#:~:text=Access%20control%20lists%20are%20used%20for%20controlling%20permissions%20to%20a,devices%20that%20users%20access%20directly.
- https://learningnetwork.cisco.com/s/question/0D53i00000KsokACAR/editing-numbered-named-acls
- https://learningnetwork.cisco.com/s/question/0D53i00000Kt6wXCAR/difference-between-numbered-acl-and-named-acl
- https://techhub.hpe.com/eginfolib/networking/docs/switches/RA/15-18/5998-8151_ra_2620_asg/content/ch10s10.html