An Extended IP ACL can filter a packet based on its source and destination IP address, protocol information, port number, message type for ICMP and TCP/IP protocol such as FTP, HTTP, SSH, Telnet etc. In this tutorial I will explain extended access list configuration commands and its parameters in detail with example.

This tutorial is the fourth part of our article “Cisco IP ACL Configuration Guide”. You can read other parts of this article here:-

http://computernetworkingnotes.com/ccna-study-guide/access-control-list-explained-with-examples.html

This tutorial is the first part of this article. In this part I provided a brief introduction to Cisco IP ACLs such as what is ACL and how it works including ACLs direction and locations.


This tutorial is the second part of this article. In this part I explained Standard Access Control List configuration commands and its parameters in detail with examples.


This tutorial is the third part of this article. In this part I provided a step by step configuration guide for Standard Access Control List.


This tutorial is the last part of this article. In this part I will provide a step by step configuration guide for Extended Access Control List.

Just like Standard ACL we can create Extended ACL in two ways:-

1. Classical approach (numbered)
2. Modern approach (numbered or named)

**Classical approach (numbered ACL)**

To create an Extended numbered ACL following global configuration mode command is used:-

```
Router(config)# access-list ACL_Identifier_number permit|deny IP_protocol source_address source_wildcard_mask [protocol_information] destination_address destination_wildcard_mask [protocol_information] [log]
```

This command prompt indicates that we are in global configuration mode.
access-list

Through this parameter we tell router that we are creating or accessing an access list.

ACL_Identifier_number

With this parameter we specify the type of access list. We have two types of access list; standard and extended. Both lists have their own unique identifier numbers. Extended ACL uses numbers range 100 to 199 and 2000 to 2699. We can pick any number from this range to tell the router that we are working with Extended ACL. This number is used in grouping the conditions under a single ACL. This number is also a unique identifier for this ACL in router.

permit/deny

As we know an ACL condition has two actions; permit and deny. If we use permit keyword, ACL will allow all packets that match with parameters specified next in command. If we use deny keyword, ACL will drop all packets which match with following specified parameters.

IP_protocol

This parameter tells router that what kind of filtering we want. We have two choices here, host level filtering and application level filtering. Host level filtering is used for generic filtering while application level filtering is used for more specific filtering. In easy language Host level filtering checks “Whether host A is allowed to access host B or not” while application level filtering checks “How much host A is allowed to access host B”.

Host Level filtering

IP

For host level filtering we need to use IP keyword here. Please make sure if you choose IP here, you will not be able to specify a specific application layer protocol in this statement later. Generic command for host level filtering is following

Router(config)#  access-list 100-199|2000-2699 permit|deny ip  source_address source_wildcard_mask destination_address destination_wildcard_mask  [log]

After IP keyword we need to provide source and destination address with wildcard mask. I have already explained wildcard mask in detail with example in second part of this article.

In standard ACL, to match a specific host we are allowed to type IP address alone. (Router will automatically add host keyword with it). But in extended ACL we have to type host keyword with IP address to match a specific host.
Application level filtering

For application level filtering we need to use appropriate layer 4 (Transport) protocol here such as TCP, UDP and ICMP. Depending on protocol we are allowed to use more specific filtering parameters in statement later.

TCP/UDP

To filter a packet based on TCP/UDP header we need to use following command

```
Router(config)# access-list 100-199|2000-2699 permit|deny tcp|udp source_address source_wildcard_mask [operator source_port_#] destination_address destination_wildcard_mask [operator destination_port_#] [established] [log]
```

Port

Port numbers are used to distinguish between different applications data. For example a server performs a number of functions like email, FTP, DNS, Web service, file service, data service etc. TCP/UDP assigns a unique number to each application, so its data doesn’t get mix-up with others applications in transmission. These unique numbers are called Port number. Extended ACL can filter data packet based on port numbers or application names. Following table lists some most common port numbers and their associate applications.

<table>
<thead>
<tr>
<th>TCP Port Number</th>
<th>Application</th>
<th>ACL Keyword</th>
<th>UDP Port Number</th>
<th>Application</th>
<th>ACL Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>FTP (Data)</td>
<td>ftp-data</td>
<td>53</td>
<td>DNS</td>
<td>domain</td>
</tr>
<tr>
<td>21</td>
<td>FTP (Control)</td>
<td>ftp</td>
<td>67,68</td>
<td>DHCP</td>
<td>nameserver</td>
</tr>
<tr>
<td>22</td>
<td>SSH</td>
<td></td>
<td>69</td>
<td>TFTP</td>
<td>tftp</td>
</tr>
<tr>
<td>23</td>
<td>Telnet</td>
<td>telnet</td>
<td>123</td>
<td>NTP</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
<td>smtp</td>
<td>161</td>
<td>SNMP</td>
<td>snmp</td>
</tr>
<tr>
<td>53</td>
<td>DNS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>HTTP</td>
<td>www</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>POP3</td>
<td>pop3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>443</td>
<td>SSL (HTTPs)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Operators

Operators are used to match port numbers or application names. There are five operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lt</td>
<td>Less than</td>
</tr>
<tr>
<td>gt</td>
<td>Greater than</td>
</tr>
<tr>
<td>neq</td>
<td>Not equal to</td>
</tr>
<tr>
<td>eq</td>
<td>equal to</td>
</tr>
<tr>
<td>Range</td>
<td>Range of port numbers</td>
</tr>
</tbody>
</table>
Only TCP/UDP uses operators. Other IP protocols do not require them.

Operators are optional. If we omit them, ACL will match all TCP/UDP packets.

Established

Established keyword is used only with TCP packets. With this keyword we can control the direction of data flow. As we know, user data packets are always transported in TCP packets. If we use this keyword, ACL will allow only the TCP packets which have establish flag bit set in their header. Logic behind this keyword is that allow traffic only if it is originated from inside.

Log

Log keyword is used to log every matched packet. It asks router to log a message every time when an ACL is hit. This feature is extremely useful in monitoring inappropriate access attempts.

ICMP

Sending a packet is not a guarantee of delivering the packet. Sometime packets get lost in their way to destination. In such a situation nearest device sends error message back to sender. So it can get an idea about undelivered packets and their possible reasons. Networking devices use ICMP protocol to send error messages.

Following command is used to filter the ICMP traffic

```
Router(config)# access-list 100-199|2000-2699 permit|deny icmp
source_address source_wildcard_mask destination_address destination_wildcard_mask
[icmp_message] [log]
```

If we do not specify a particular message type, ACL will match all message types.

Following table lists some of common ICMP messages.

<table>
<thead>
<tr>
<th>Message (Keyword)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo</td>
<td>Used to check the status of destination (up/down)</td>
</tr>
<tr>
<td>echo-reply</td>
<td>Reply from destination on echo request</td>
</tr>
<tr>
<td>host-unreachable</td>
<td>Network is reachable, but particular host is not responding</td>
</tr>
<tr>
<td>net-unreachable</td>
<td>Network is not reachable</td>
</tr>
<tr>
<td>traceroute</td>
<td>Filter traceroute information</td>
</tr>
<tr>
<td>administrativelyprohibited</td>
<td>Packet filtered by ACL</td>
</tr>
</tbody>
</table>

Beside IP, TCP, UDP and ICMP we can also filter a packet based on ahp (Authentication Header Protocol), eigrp (Cisco’s EIGRP routing protocol), esp (Encapsulation Security Payload), gre (Cisco’s GRE tunneling), igmp (Internet Gateway Message Protocol), ipinip (IP in IP tunneling),
nos KA9Q (NOS compatible IP over IP tunneling), ospf (OSPF routing protocol), pcp (Payload Compression Protocol) and pim (Protocol Independent Multicast). These options are not included in any associate (CCNA) level exam syllabus. For CCNA level exams we should focus only on four protocols IP, TCP, UDP and ICMP.

Modern approach (numbered or named)

Starting from Cisco IOS version 11.2, routers support modern configuration approach. While in classical style we are not allowed to edit/update/delete a single line from ACL, in modern style we can edit/update/delete a single line from ACL.

To create an extended ACL in modern style use following commands

```
Router(config)# ip access-list extended ACL_name_number

ACL_name_number

Use any unique name or number for ACL.

Once you enter above command, we are moved into the ACL sub-configuration mode

Router(config-ext-acl)#
```

From sub-configuration mode we can create ACL conditions

```
Router(config)# ip access-list extended ACL_name_number
Router(config-ext-acl)# permit|deny IP_protocol source_IP_address wildcard_mask
[protocol_information] destination_IP_address wildcard_mask [protocol_information] [log]
```

Once we are finished use `exit` command to return in global configuration mode.

**How to enable Extended ACL**

No matter which method we used to create an extended ACL, enabling it in interface is the same process.

```
Router(config)#interface interface_number
Router(config-if)# ip access-group ACL_Name_name in|out
```

**How to delete Extended ACL**

Following commands are used to delete an extended ACL.

To delete an ACL created from classical approach

```
Router(config)#no access-list ACL_Identifier_number
```
To delete an ACL created from both classical and modern approach

`Router(config)# no ip access-list extended ACL_name_number`

That’s all for this part. In next part we will practically implement what we have learnt from this part.

For more articles please visit our site

[ComputerNetworkingNotes.com](http://ComputerNetworkingNotes.com)