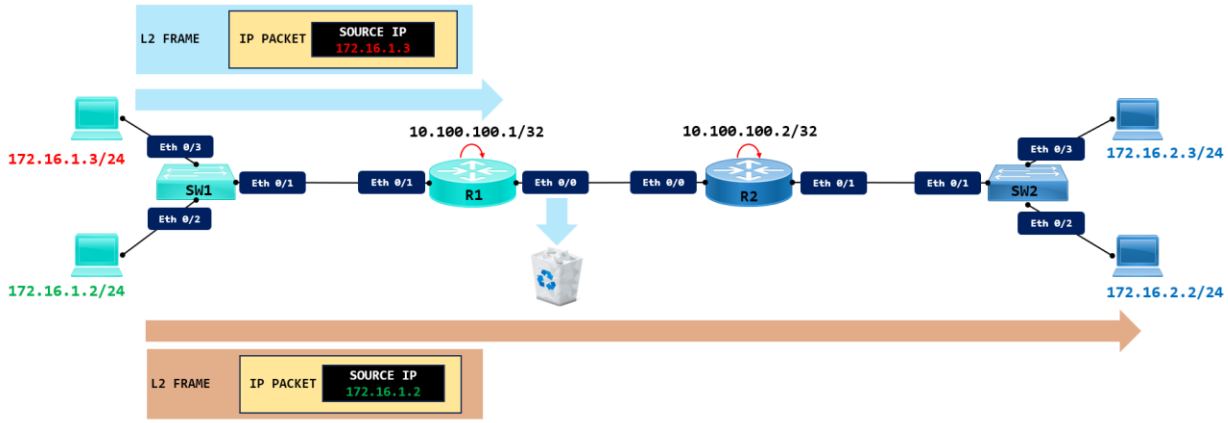
Access Control List **ตอนที่** 2

สำหรับ ACL ตอนที่ 2 เป็นการยกตัวอย่างการตั้งค่า Config ทั้งแบบที่เป็น Standard ACL และ Extended ACL รวมถึง Name ACL ด้วย

Standard ACL มีความสามารถเฉพาะในการกรองหรือเลือก Traffic ได้แค่เพียง Source Address เท่านั้น ไม่สามารถที่ระบุ Protocol และ Destination Address หรือ IP ปลายทางได้

ตัวอย่าง: ใน Diagram รูปด้านล่าง ต้องการให้ User IP 172.16.1.3 เพียงเครื่องเดียว ที่ไม่สามารถติดต่อหรือ Ping ไปยัง User ที่อยู่ในเครือข่ายวง 172.16.2.0/24 ได้



ทำการเขียน ACL แบบ Standard ตามเงื่อนไขที่ต้องการ โดยทำการ Config บน Router R1

```
R1(config)#access-list 1 deny   172.16.1.3
R1(config)#access-list 1 permit 172.16.1.0 0.0.0.255
```

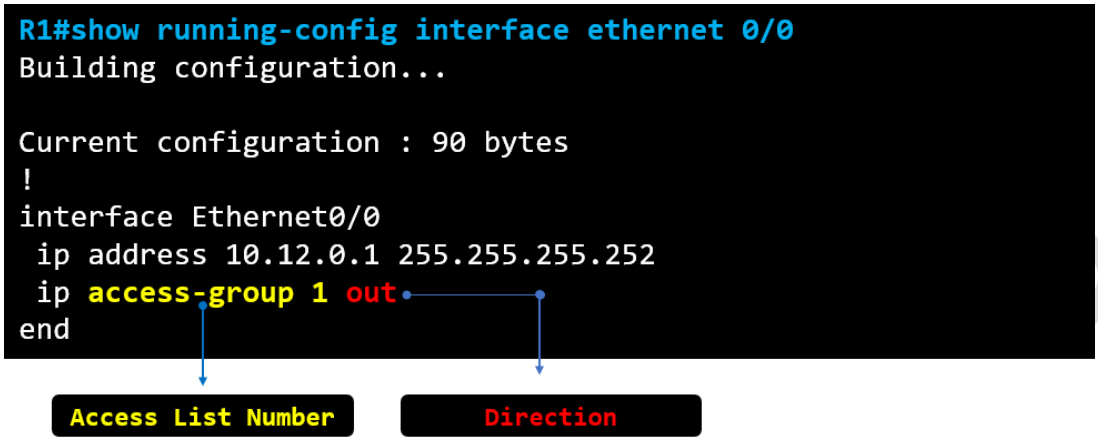
ใช้คำสั่ง “show ip access-list [Access-List Number]” เพื่อทำการตรวจสอบ Statement หลังจาก Config จะเห็นได้ว่ามี 2 Statement คือ 10 และ 20 ตามลำดับ

```
R1#show ip access-lists 1
Standard IP access list 1
    10 deny   172.16.1.3
    20 permit 172.16.1.0, wildcard bits 0.0.0.255
```

Statement ที่ 10 Host ที่มี IP 172.16.1.3 เท่านั้นที่ไม่อนุญาตให้ผ่านได้

Statement ที่ 20 ให้ทุก Host ที่มี IP 172.16.1.0-255 สามารถผ่านได้

หลังจากที่กำหนดเงื่อนไขของ ACL แล้ว ลำดับถัดไปนำ ALC ไป Apply ที่ Interface สำหรับตัวอย่างนี้จะนำ ACL Standard 1 ไป Apply บน Interface Eth 0/0 ซึ่งเป็น Interface ที่เชื่อมต่อระหว่าง Router R1 และ R2 โดยให้มีทิศทางเป็น Out (Outbound)



ทำการทดสอบหลังจาก Apply ACL ลงบน Interface เป็นไปตามเงื่อนไขที่ต้องการหรือไม่ โดยการ Ping จาก R1-PC3 ที่มี IP 172.16.1.3 ไป R2-PC3 จะเห็นได้ว่าไม่สามารถ Ping ไปยังปลายทางได้

```
R1-PC3> show ip

NAME        : R1-PC3[1]
IP/MASK     : 172.16.1.3/24
GATEWAY     : 172.16.1.1
DNS         :
MAC         : 00:50:79:66:68:03
LPORT       : 20000
RHOST:PORT  : 127.0.0.1:30000
MTU         : 1500

R1-PC3> ping 172.16.2.3

*172.16.1.1 icmp_seq=1 ttl=255 time=2.368 ms (ICMP type:3, code:13, Communication administratively prohibited)
*172.16.1.1 icmp_seq=2 ttl=255 time=1.593 ms (ICMP type:3, code:13, Communication administratively prohibited)
*172.16.1.1 icmp_seq=3 ttl=255 time=2.200 ms (ICMP type:3, code:13, Communication administratively prohibited)
*172.16.1.1 icmp_seq=4 ttl=255 time=1.252 ms (ICMP type:3, code:13, Communication administratively prohibited)
*172.16.1.1 icmp_seq=5 ttl=255 time=1.800 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

ทดสอบ Ping จาก R1-PC2 ที่มี IP 172.16.1.2 ไป R2-PC3 จะเห็นได้ว่าสามารถ Ping ไปยังปลายทางได้

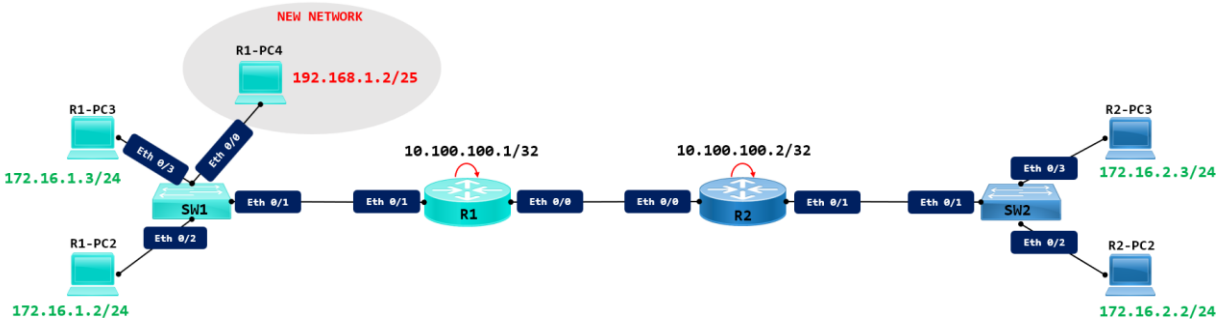
```
R1-PC2> show ip

NAME        : R1-PC2[1]
IP/MASK     : 172.16.1.2/24
GATEWAY     : 172.16.1.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 20000
RHOST:PORT  : 127.0.0.1:30000
MTU         : 1500

R1-PC2> ping 172.16.2.3

84 bytes from 172.16.2.3 icmp_seq=1 ttl=62 time=2.580 ms
84 bytes from 172.16.2.3 icmp_seq=2 ttl=62 time=2.647 ms
84 bytes from 172.16.2.3 icmp_seq=3 ttl=62 time=2.883 ms
84 bytes from 172.16.2.3 icmp_seq=4 ttl=62 time=2.119 ms
84 bytes from 172.16.2.3 icmp_seq=5 ttl=62 time=2.621 ms
```

เมื่อตรวจสอบที่ Router R1 โดยใช้คำสั่ง “show ip access-list [Number]” จะเห็นได้ว่า มี Packet ที่เข้าเงื่อนไขของ ACL ด้วยเช่นกัน ซึ่งสังเกตได้จากคำว่า “matches”

```
R1#show ip access-lists 1
Standard IP access list 1
    10 deny   172.16.1.3 (10 matches)
    20 permit 172.16.1.0, wildcard bits 0.0.0.255 (5 matches)
```

จากผลการทดสอบข้างต้นจะเห็นได้ว่า ACL Standard 1 ที่ Config บน Router R1 และ Apply บน Interface Eth 0/0 ทิศทาง Out นั้นมีผลตามที่ต้องการ

แล้วถ้าหากที่ Router R1 มีเครือข่ายหรือ Network LAN เพิ่มขึ้นมาอีก 1 Network เป็น IP 192.168.1.0/25 จะสามารถ Ping จาก Network วงใหม่ไปยัง R2-PC3 ได้หรือไม่




```
R1-PC4> show ip

NAME        : R1-PC4[1]
IP/MASK     : 192.168.1.2/25
GATEWAY     : 192.168.1.1
DNS         :
MAC         : 00:50:79:66:68:07
LPORT       : 20000
RHOST:PORT  : 127.0.0.1:30000
MTU         : 1500

R1-PC4> ping 172.16.2.3

*192.168.1.1 icmp_seq=1 ttl=255 time=1.621 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.1.1 icmp_seq=2 ttl=255 time=1.601 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.1.1 icmp_seq=3 ttl=255 time=1.270 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.1.1 icmp_seq=4 ttl=255 time=1.771 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.1.1 icmp_seq=5 ttl=255 time=1.827 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

จะเห็นได้ว่า R1-PC4 IP 192.168.1.2/25 ไม่สามารถ Ping ไปหา R2-PC3 ได้ นั่นก็เพราะว่าไม่เข้าข่ายกับเงื่อนไขที่ระบุไว้บน ALC ที่มีอยู่เดิม เมื่อ Packet ที่ไม่เข้าข่ายกับ ACL ก็จะถูกปฏิเสธหรือ Deny โดย “implicit deny all” หรือ “deny any” ซึ่ง implicit deny all จะเป็น Statement สุดท้ายถูกซ่อนเอาไว้และไม่แสดงบน Configuration ของ ACL จึงทำให้ R1-PC4 ไม่สามารถติดต่อกับ R2-PC3 ได้นั่นเอง

```
R1#show ip access-lists 1
Standard IP access list 1
    10 deny   172.16.1.3
    20 permit 172.16.1.0, wildcard bits 0.0.0.255
       --Implicit deny all --
```

ถ้าหากต้องการให้ User 192.168.1.2/25 สามารถติดต่อไปยัง R2-PC3 ได้ ก็ต้องสร้าง Statement ขึ้นมาเพิ่มเพื่อรองรับ Network วงใหม่ โดยปกติแล้ว ค่า Sequence ของ Statement เริ่มต้นอยู่ที่ 10 และ

ระหว่าง Statement จะห่างกันเท่ากับ 10 เช่นกัน ในตัวอย่าง Statement แรกจะเริ่มต้นที่ 10 และลำดับถัดไป คือ 20 แต่ในที่นี้ต้องการให้ Network วงใหม่อยู่บนสุดจะมีวิธีการอย่างไรบ้าง

- ตรวจสอบ ACL เดิมที่ได้ทำการ Config ไว้ “show ip access-list [Number]”

```
R1#show ip access-lists 1
Standard IP access list 1
    10 deny   172.16.1.3
    20 permit 172.16.1.0, wildcard bits 0.0.0.255
```

การแทรก Statement เพื่อสร้างเงื่อนไขสำหรับ Network วงใหม่ โดยจะให้เป็น Statement ที่อยู่บนสุด หรือเป็นอันดับแรก ซึ่งสามารถกำหนดหมายเลขได้ ตั้งแต่ 1-9 ในตัวอย่างได้กำหนดหมายเลขเป็น 5

```
R1(config)#ip access-list standard 1 ⟶ (Number)
                          (standard/extended)
R1(config-std-nacl)#5 permit 192.168.1.0 0.0.0.128
  (Sequence Number)
```

```
R1#show ip access-lists 1                                  New Statement
Standard IP access list 1
    5  permit 192.168.1.0, wildcard bits 0.0.0.127
    10 deny   172.16.1.3
    20 permit 172.16.1.0, wildcard bits 0.0.0.255
```

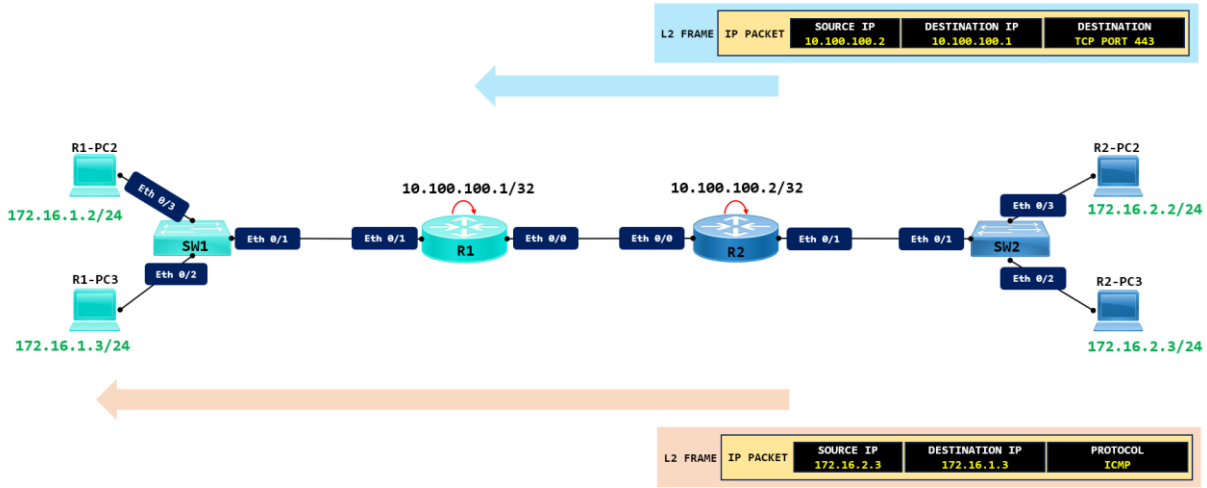
หลังจากที่ทำการเพิ่ม Statement ไปแล้ว ทำการทดสอบ Ping จาก R1-PC4 ไปยัง R2-PC3 จะเห็นได้ว่า R1-PC4 สามารถ Ping ไปยัง R2-PC3 ได้แล้ว

```
R1-PC4> ping 172.16.2.3

84 bytes from 172.16.2.3 icmp_seq=1 ttl=62 time=2.304 ms
84 bytes from 172.16.2.3 icmp_seq=2 ttl=62 time=1.025 ms
84 bytes from 172.16.2.3 icmp_seq=3 ttl=62 time=1.330 ms
84 bytes from 172.16.2.3 icmp_seq=4 ttl=62 time=1.285 ms
84 bytes from 172.16.2.3 icmp_seq=5 ttl=62 time=1.099 ms
```

Extended ACL

Extended ACL สามารถสร้างเงื่อนไขที่มีความยืดหยุ่นหรือระบุรายละเอียดได้มากกว่า Standard ACL อย่างเช่น สามารถระบุ IP ต้นทางและปลายทาง พร้อมทั้ง Protocol ที่ต้องการได้ อย่างเช่น UDP/TCP ICMP เป็นต้น

ในตัวอย่างด้านล่างต้องการให้ Router R2 สามารถ Remote เข้า Router R1 ได้โดยผ่าน https และต้องการให้ R1-PC3 Ping ไปยัง R2-PC3 จากความต้องการนี้ เราสามารถ Config ACL Extended บน Router R1 อย่างไรได้บ้าง? เพื่อให้ตรงกับเงื่อนไขที่ต้องการ



รูปแบบคำสั่งที่ใช้ Config ACL Extended

```
R1(config)#access-list [100-199 or 2000-2699] [permit/deny] [Protocol] [Source-IP] [Source Port Number] [Destination IP] [Destination Port]
```

Training

ทำการ Config Extended ACL และ Apply บน Interface ที่ Router R1

```
R1(config)#access-list 100 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443

R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
```

- ใน Statement แรก เป็นการอนุญาตให้ IP ต้นทาง 10.100.100.2 ไปยัง IP ปลายทาง 10.100.100.1 ที่เป็น TCP Port 443 (https) ได้ และที่ Router R1 ได้เปิดใช้งาน “ip http secure-server” ไว้เพื่อเป็นการจำลองการ Access เข้า Router โดยใช้ https หรือหน้า Web Browser

```
R1#show running-config interface ethernet 0/0
Building configuration...

Current configuration : 91 bytes
!
interface Ethernet0/0
 ip address 10.12.0.1 255.255.255.252
 ip access-group 100 in
end
```

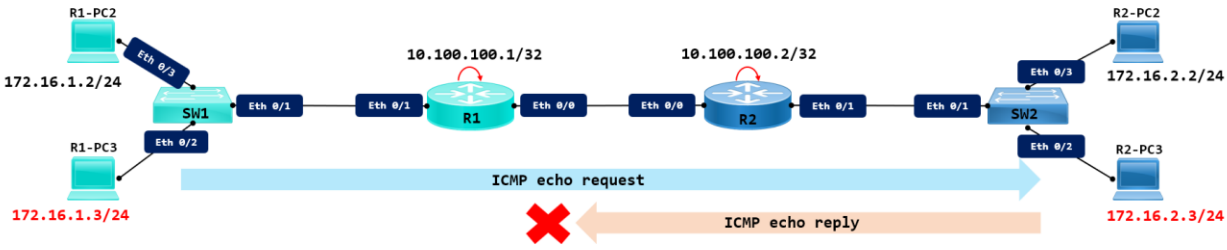
- หลังจากที่สร้างเงื่อนไข ACL และ Apply บน Interface แล้ว ก็ทำการทดสอบผลเป็นไปตามเงื่อนไขที่ต้องการหรือไม่

```
R2#telnet 10.100.100.1 443 /source-interface loopback 0
  Destination IP      Destination TCP Port Number        Source IP = 10.100.100.2

Trying 10.100.100.1, 443 ... Open
```

ผลทดสอบการเชื่อมต่อจะเห็นได้ว่า Router R2 สามารถ Access Router R1 TCP Port 443 ได้ แต่อีกเงื่อนไขหนึ่งคือ R1-PC3 ยังไม่สามารถ Ping ไปยัง R2-PC3 ได้ เพราะอะไร?

```
R1-PC3> ping 172.16.2.3

172.16.2.3 icmp_seq=1 timeout
172.16.2.3 icmp_seq=2 timeout
172.16.2.3 icmp_seq=3 timeout
172.16.2.3 icmp_seq=4 timeout
172.16.2.3 icmp_seq=5 timeout
```

การทำงานของ Ping จะใช้ Packet ขนาดเล็ก ICMP echo request และ ICMP echo reply ในตัวอย่างนี้ R1-PC3 ได้ส่ง ICMP echo request ไปยัง R2-PC3 แล้ว แต่ตัว Router R1 ไม่เงื่อนไขที่อนุญาตให้ ICMP ผ่านได้ จึงเป็นเหตุให้ ICMP echo reply จาก R2-PC3 ถูก Filter ทิ้งไปที่ Router R1 Interface Eth0/0 ทำให้ R1-PC3 และ R2-PC3 Ping กันไม่เจอ นั่นเอง



ทำการสร้างเงื่อนไขเพิ่มอีก 1 Statement เพื่อให้ R1-PC3 172.16.1.3/24 สามารถ Ping ไปยัง R2-PC3 172.16.2.3/24 ได้

```
R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443


R1(config)#ip access-list extended 100
R1(config-ext-nacl)#20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
   Add New Sequence          Protocol     Source IP Address      Destination IP Address
R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443 (8 matches)
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255 (5 matches)
```

ทดสอบ Ping จาก R1-PC3 ไปยัง R2-PC3 หลังจากเพิ่ม Statement เพื่อรองรับ Protocol ICMP แล้ว ซึ่งผลการทดสอบ สามารถ Ping จาก R1-PC3 ไปยัง R2-PC3 ตามความต้องการได้

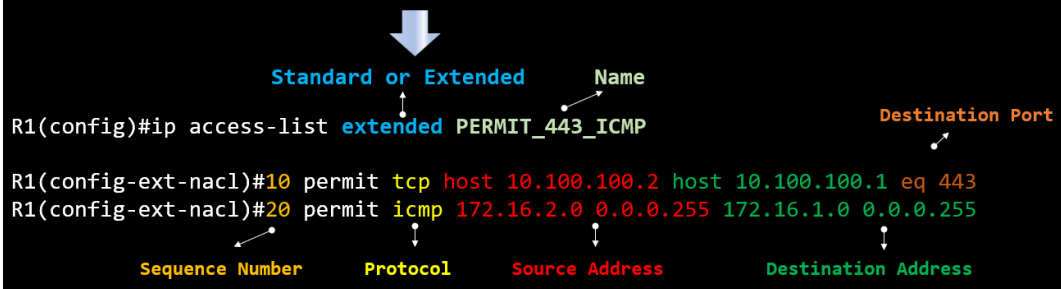
```
R1-PC3> ping 172.16.2.3

84 bytes from 172.16.2.3 icmp_seq=1 ttl=62 time=3.068 ms
84 bytes from 172.16.2.3 icmp_seq=2 ttl=62 time=1.184 ms
84 bytes from 172.16.2.3 icmp_seq=3 ttl=62 time=1.081 ms
84 bytes from 172.16.2.3 icmp_seq=4 ttl=62 time=1.382 ms
84 bytes from 172.16.2.3 icmp_seq=5 ttl=62 time=1.648 ms
```

Name ACL

Name ACL เป็นการใช้ชื่อแทนที่การกำหนดหมายเลขในการระบุประเภทของ ACL ข้อดีคือสามารถจดจำได้ง่าย รองรับทั้ง Standard และ Extended ACL

ในตัวอย่างที่แสดงต่อไปนี้ จะขอนำตัวอย่างจาก Extended ACL ด้านบน มา Config ในรูปแบบเป็น Name ACL

```
R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
```



ส่วนการนำไป Apply บน Interface มีรูปแบบเดียวกันกับ Standard และ Extended โดยเปลี่ยนจากหมายเลขของ ACL เป็นชื่อของ ACL ตามรูปที่ได้แสดงด้านล่าง

```
R1#show running-config interface ethernet 0/0
Building configuration...

Current configuration : 91 bytes
!
interface Ethernet0/0
 ip address 10.12.0.1 255.255.255.252
 ip access-group PERMIT_443_ICMP in
end
```

ข้อควรระวังในการลบ-แทรก Access List

```
R1#show running-config | include access-list
access-list 100 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
access-list 100 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
```


```
R1(config)#no access-list 100permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255


R1#show running-config | include access-list
```

จาก Configuration ด้านบน ถ้าหากต้องการลบเงื่อนไขที่เป็นตัวอักษรสีแดง จะไม่สามารถใช้คำสั่ง “no access-list 100 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255” ได้ เพราะจะทำให้ทุกเงื่อนไขใน ACL Extended 100 ถูกลบทั้งหมด

ถ้าหากต้องการเพิ่มหรือลบ ให้ใช้คำสั่ง “show ip access-list [Number | Name]” เพื่อดูลำดับ Sequence ของ Statement ก่อนทำการแก้ไข ในตัวอย่างด้านล่างจะทำการแทรกเงื่อนไขระหว่าง Statement Sequence ที่10 และ 20 โดยใช้เลข Sequence เป็น 15 หลังจากนั้นจะแสดงวิธีการลบ Statement ซึ่งสามารถ Config ได้ตามรูปที่แสดงด้านล่าง

- การเพิ่มหรือแทรกเงื่อนไข

```
R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255


R1(config)#ip access-list extended 100
R1(config-ext-nacl)#15 deny ip any any
R1(config-ext-nacl)#end

R1#show ip access-lists
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
    15 deny ip any any
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
```

Before Add Statement

Add Statement

After Add Statement

- การลบเงื่อนไข

```
R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
    15 deny ip any any
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255

R1(config)#ip access-list extended 100
R1(config-ext-nacl)#no 15

R1#show ip access-lists 100
Extended IP access list 100
    10 permit tcp host 10.100.100.2 host 10.100.100.1 eq 443
    20 permit icmp 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
```

Before Delete Statement

Delete Statement

After Delete Statement

แต่ละหน่วยงานหรือแต่ละ Network ก็จะมีต้องการที่จะเข้าถึง Traffic ประเภทต่างๆ แตกต่างกันออกไป หลังจากการสร้างและลบ Statement ใน ACL ก็ควรจะทดสอบ เพื่อให้ตรงกับความต้องการที่ใช้งานด้วยเช่นกัน