แผนบริหารการสอนประจำบท
บทที่ 8 ความปลอดภัยระบบเครือข่าย

วัตถุประสงค์

1. บอกถึงประโยชน์ของ ACL ได้
2. บอกถึงการนำ ACL ไปใช้งานได้
3. อธิบายการ Wildcard Mask ได้

เนื้อหา

1. ประโยชน์ของ Access Control List (ACL)
2. พฤติกรรมของ ACL
3. ลักษณะการบังคับใช้ ACL
4. หลักการและข้อควรคำนึงอื่น ๆ เกี่ยวกับ ACL
5. Wildcard Mask
6. Standard Access Control Lists
7. Extended Access Control Lists
8. Named Access Control Lists
9. การใช้ ACL เพื่อควบคุมการข้าถึง Line VTY
10. เปรียบเทียบ ACL กับไฟล์วอลล์

กิจกรรมการเรียนการสอน

1. ผู้สอนอธิบายวัตถุประสงค์ ความคิดรวบยอด ขอบเขตเนื้อหา วิธีการเรียน และกิจกรรมการเรียนการสอนประจำบทเรียน
2. ผู้สอนใช้สไลน์และเอกสารประกอบการสอนในรูปแบบไฟล์อิเล็กทรอนิกส์ประเภท PPT ประกอบการบรรยายเนื้อหาประเด็นสำคัญ
3. ผู้สอนบรรยายสรุปเนื้อหาและประเด็นสำคัญประจำบทเรียน
4. ผู้เรียนทำแบบฝึกหัด เพื่อเป็นการทำทวนความรู้ความเข้าใจเนื้อหาประจำบท และประเมินผลเป็นคะแนนระหว่างเรียน

5. ผู้เรียนทำงานตามที่ได้รับมอบหมายประจำบทเรียน โดยให้ผู้เรียนส่งงานในรูปแบบต่าง ๆ ตามที่ผู้สอนกำหนด

สื่อการเรียนการสอน

1. เอกสารประกอบการสอน รายวิชาความรู้เบื้องต้นเกี่ยวกับเครือข่ายคอมพิวเตอร์ ซึ่งเรียบเรียงโดยอาจารย์สุลัยมาน เภอโส๊ะ สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์เทคโนโลยีและการเกษตร
2. สไลน์ประกอบการสอน รายวิชาความรู้เบื้องต้นเกี่ยวกับเครือข่ายคอมพิวเตอร์ ซึ่งเผยแพร่ไว้บนเว็บไซต์อีเลิร์นนิ่งของมหาวิทยาลัยราชภัฏยะลา โดยมีที่อยู่ของเว็บไซต์ คือ http://elearning.yru.ac.th

การวัดผลและการประเมินผบ

1. วัดและประเมินผลจากคะแนนแบบฝึกหัด และให้คะแนนตามเกณฑ์ที่กำหนดไว้ล่วงหน้า แล้วบันทึกเป็นคะแนนระหว่างเรียนของผู้เรียนแต่ละคน
2. ประเมินผลงานหรือการบ้านที่ผู้สอนมอบหมายให้ผู้เรียนปฏิบัติประจำบทเรียน และให้คะแนนตามเกณฑ์ที่กำหนดไว้ล่วงหน้า แล้วบันทึกเป็นคะแนนระหว่างเรียนของผู้เรียนแต่ละคน

บทที่ 8
ความปลอดภัยระบบเครือข่าย

ในบทนี้จะกล่าวถึงการรักษาความปลอดภัยโดย Access Control Lists (ACL) ซึ่งเป็นฟีเจอร์หนึ่งบน IOS ของซิสโก้ที่ช่วยในการสร้างความปลอดภัยกับระบบเน็ตเวิร์กและเร้าเตอร์

8.1 ประโยชน์ของ Access Control List (ACL)

ACL ใน IOS ของซิสโก้ นอกจากจะใช้เพื่อสร้างความปลอดภัยให้กับเน็ตเวิร์กได้แล้ว ยังมีประโยชน์อื่น ๆ อีกมากมายในแทบจะทุกเรื่องของการเซตคอนฟิกูเรชัน เช่น

1) สร้างความปลอดภัยโดยการควบคุมประเภทของทราฟฟิกที่ผ่านมาเข้าออกได้ ตัวอย่างเช่น บนเร้าเตอร์ตัวริมที่ต่อกับอินเทอร์เน็ตดดยตรง เราสามารถเซต ACL ขึ้นมาเพื่อป้องกันทราฟฟิกอันตราย เช่น ทราฟฟิกที่ต้องการเข้าถึงพอร์ต 135,445,139 ไม่ให้สมารถเข้ามายังเน็ตเวิร์กภายในได้

2) เซตประเภทของทราฟฟิกที่จะสามารถกระตุ้นให้ ISDN Line ทำงานได้ ดังในตัวอย่างบทที่ 17 โดย ACL จะถูกใช้คู่กับคำสั่ง dialer –list (และคำสั่ง dialer –list จะถูกนำไปจับคู่กับคำสั่ง dialer –group ภายใต้อินเตอร์เฟซ BRI อีกที)

3) เซตประเภทของทราฟฟิกที่จะถูกนำไปใช้ในฟีเจอร์เกี่ยวกับเร้าติ้งโปรโตคอล เช่น นำไปใช้งานร่วมกับ Route – Map, การทำ Redistribution , การกำหนดเร้าติ้งเอ็นทรีที่จะประกาศ (advertise) ออกไปยังเร้าเตอร์เพื่อนบ้าน (distribute list) เป็นต้น

8.2 พฤติกรรมของ ACL

ACL จะถูกไล่เปรียบเทียบจากบรรทัดบนลงบรรทัดล่างทีละบรรทัดละบรรทัด จนกว่าจะพบบรรทัดที่มีเงื่อนไขสอดคล้องกับแพ็กเก็ตที่วิ่งเข้ามาให้ตรวจเช็กในขณะนั้น เมื่อพบแล้ว เร้าเตอร์จะดูว่า “action” ที่ตั้งไว้เป็น permit หรือ deny หากเป็น PERMIT เร้าเตอร์จะอนุญาตให้ทราฟฟิกนั้นวิ่งผ่านไปได้ แต่หากเป็น DENY ทราฟฟิกนั้นจะถูกโยงทิ้ง (Drop) ไป

ลำดับบรรทัดของ Access Control Entry (ACE) แต่ละบรรทัดที่ถูกสร้างลงไปใน ACL จะมีความสำคัญมาก เพราะอย่างที่ได้กล่าวไปในข้อที่แล้วว่า ทราฟฟิกจะถูกเช็กจากเงื่อนไขใน ACE จากบนลง

ล่างไปเรื่อย ๆ ดังนั้น ACE ที่อยู่ในบรรทัดแรกๆ ที่คีย์ลงไปควรจะมีเงื่อนไขที่ละเอียดและเจาะจง (more specific) มากกว่าเงื่อนไขของ ACE ที่อยู่บรรทัดท้าย ๆ ตัวอย่างเช่น หากใน ACE แรกๆ เซตไว้ว่า access-list 101 permit ip any และ ACE ท้าย ๆ เซตไว้ว่า access-list 101 deny tcp any any eq 135 ผลที่ได้ก็คือ ทราฟฟิกที่ต้องการ DENY นั้นจะได้รับอนุญาตให้ผ่านไปได้ สาเหตุก็เพราะ เมื่อมีการประเมินเงื่อนไข , ACE แรก ๆ ที่เซตไว้ได้อนุญาตให้ทราฟฟิกวิ่งผ่านไปก่อนแล้วทราฟฟิกจึงไม่ถูกประเมินด้วย ACE ท้าย ๆ ตามที่ต้องการ

ทุก ๆ ACL ที่สร้างขึ้นมาจะมีเงื่อนไขสุดท้ายที่ถูกซ่อนไว้เสมอ เรานิยมเรียกว่า implicit deny all ความหมายก็คือ ทราฟฟิกใดๆ ที่ไม่สอดคล้องกับเงื่อนไขในบรรทัดต่างๆ ก่อนหน้านี้ทราฟฟิกประเภทานั้นจะถือว่าถูก “ปฏิเสธ (deny)” ไปโดยปริยายและถูกโยนทิ้งไปโดยอัตโนมัติ

8.2.1 ประเภทของ ACL

ACL แบบพื้นฐานจะมีอยู่ด้วยกัน 2 ประเภท ได้แก่ Standard ACL และ Extended ACL

8.2.1.1 Standard ACL ประเภทนี้จะตรวจเช็กได้เฉพาะหมายเลขแอดเดรสต้นทาง (Source Address) ของแพ็กเก็ตเท่านั้นว่าอยู่ในเงื่อนไขที่ต้องการหรือไม่ ACL ประเภทนี้จึงไม่สามารถแยกแยะลงไปในรายละเอียดและฟิลด์ส่วนอื่นๆ ของแพ็กเก็ตได้ เช่น หมายเลข TCP/UDP Port, Destination IP Address เป็นต้น

8.2.1.2 Extended ACLประเภทนี้สามารถประเมินค่าฟิลด์อื่นๆ ของแพ็กเก็ตได้อย่างละเอียด มันสามารถตรวจเช็กได้ทั้งฟิลดีในเลเยอร์ 3 และเลเยอร์ที่ 4 ได้แก่ ตรวจเช็กหมายเลข Source IP Address, Destination IP Address, ฟิลด์ Protocol ในส่วนแฮดเดอร์ของแพ็กเก็ต IP (คงจำได้จากบทที่ 3 ว่าฟิลด์ Protocol ใช้บ่งบอกว่าข้อมูลในเลเยอร์บนเป็น TCP หรือ UDP), หมายเลขพอร์ต TCP/UDP ทั้งพอร์ตต้นทางและพอร์ตปลายทาง ACL ประเภทนี้จึงให้เวื่อนไขในการตัดสินใจได้อย่างละเอียดมากขึ้น

โดยปกติ เราจะอ้างอิงถึง ACL ต่างๆ เหล่านี้ผ่านทางหมายเลข (number) แต่จริงๆ แล้ว เราสามารถตั้งชื่อที่เป็นสตริงตัวอักษรให้กับ ACL เหล่านี้ได้ด้วย เราจะเรียก ACL ประเภทนี้ว่า Named Access Control List ผู้เขียนจะได้กล่าวถึงต่อไป

8.3 ลักษณะการบังคับใช้ ACL

ACL สามารถบังคับใช้ทั้งในลักษณะ Inbound (ทิศทางขาเข้ามายังอินเตอร์เฟซของเร้าเตอร์) ในลักษณะ Outbound (ทิศทางขาออกจากอินเตอร์เฟซของเร้าเตอร์)

Inbound

ACL แบบ Inbound สามารถสร้างได้โดยการสร้าง access-list ขึ้นมาก่อน แล้วจากนั้นก็บังคับใช้ access-list นั้นลงไปบนอินเตอร์เฟซด้วยคำสั่ง ip access-group <หมายเลข access-list> in สังเกตว่าคีย์เวิร์กข้างท้ายจะเป็น in ซึ่งหมายถึง Inbound

แพ็กเก็ตที่ได้รับอนุญาต (permit) ด้วย ACL แบบ Inbound นี้จะถูกเร้าเตอร์นำไปประมวลผลเพื่อหาเส้นทางจากเร้าติ้งเทเบิลต่อ ว่าจะส่งผ่านออกไปทางอินเตอร์เฟซขาออกอินเตอร์เฟซใด ส่วนแพ็กเก็ตที่ถูกปฏิเสธ (deny) ด้วย ACL แบบ Inbound จะถูกโยนทิ้งไปในทันที

8.3.1 Outbound

ACL และ Outbound สามารถสร้างได้โดยการสร้าง access – list ขึ้นมาก่อน แล้วจากนั้นก็บังคับใช้ access – list นั้นลงไปบนอินเตอร์เฟวด้วยคำสั่ง ip access – group <หมายเลข access – list> out สังเกตว่าคีย์เวิร์กข้างท้ายจะเป็น out ซึ่งหมายถึง Outbound

แพ็กเก็ตที่ผ่านการหาเส้นทางจากเร้าติ้งเทเบิลมาแล้ว เมื่อเร้าเตอร์ทราบว่าต้องส่งแพ็กเก้ตนั้นออกไปทางอินเตอร์เฟซไหน สมมติว่าเป็น interface E0/0 เร้าเตอร์จะต้องเซ็กก่อนว่าที่อินเตอร์เฟซ E0/0 นี้มีการบังคับใช้ ACl ในขาออกหรือไม่ หากมี เร้าเตอร์จะต้องประเมินทราฟฟิกนั้นก่อนว่าอยู่ในเงื่อนไขที่ permit หรือ deny หากอยู่ในเงื่อนไขที่ permit แพ็กเก็ตก็จะได้รับการส่งไปโดยปกติ แต่หากอยู่ในเงื่อนไขที่ deny แพ็กเก็ตนั้นก็จะถูกโยนทิ้งไป

8.4 หลักการและข้อควรคำนึงอื่น ๆ เกี่ยวกับ ACL

บนอินเตอร์เฟซหนึ่งๆ สามารถมี ACL ที่บังคับใช้ในลักษณะ INBOUND และ OUTBOUND ได้อย่างละหนึ่ง ACL เท่านั้น เมื่อมีการเพิ่มเติม Access Control Entry (ACE) บรรทัดใหม่เข้าไปภายใต้ Access Control list (ACL) ที่มีอยู่เดิม ACE ที่เพิ่มเข้าไปนั้นจะถูกเพิ่มเข้าไปต่อจากบรรทัดสุดท้ายที่มีอยู่เดิม เราไม่สามารถแทรก ACE เข้าไปตรงกลางระหว่างบรรทัดที่มีอยู่เดิมได้

เมื่อมีการยกเลิก Access Control list (ACL) ออกไป เราต้องยกเลิกทั้งหมดด้วยสั่ง no access-list <หมายเลข ACL> เราไม่สามารถยกเลิกเฉพาะบาง ACE ที่มีอยู่ภายใน ACL นั้นๆ ได้ (เป็นข้อมูลล่าสุดที่ IOS เวอร์ชัน 12.0 เช่นกันในอนาคตอาจเป็นไปได้ใน IOS เวอร์ชันใหม่ๆ เราสามารถยกเลิกเฉพาะบาง ACE ได้

วิธีการแก้ไข ACL ที่ดีที่สุด (ณ IOS เวอร์ชัน 12.0) ก็คือ การ copy ACL ทั้งหมดไปไว้ใน Notepad แล้วทำการแก้ไขให้เรียบร้อย จากนั้นให้ยกเลิก ACL เดิม (ด้วยคำสั่ง no access – list) แล้วเซต ACL เดิมลงไปอีกครั้ง

ให้สร้าง ACL ขึ้นมาก่อนด้วยคำสั่ง access – list แล้วค่อยบังคับใช้ลงไปที่อินเตอร์เฟซด้วยคำสั่ง ip access – group

ACL ใช้ในการฟิลเตอร์ทราฟฟิกที่วิ่งผ่านเข้าออกเร้าเตอร์เท่านั้น ไม่ได้ใช้ฟิลเตอร์ทราฟฟิกที่เร้าเตอร์เป็นผู้ส่งออกเองโดยตรง

ควรวาง Standard ACL ไว้ที่อินเตอร์เฟซของเร้าเตอร์ที่อยู่ใกล้กับเป้าหมายปลายทาง (Destination) มากที่สุด สาเหตุก็เพราะ Standard ACL ตรวจเช็กได้เฉพาะ Source Address อย่างเดียว หากนำมาไว้ใกล้ต้นทาง แพ็กเก็ตอาจถูกฟิลเตอร์ทิ้งเร็วเกินไป

ควรวาง Extended ACL ไว้ที่อินเตอร์เฟซของเร้าเตอร์ที่อยู่ใกล้กับโฮสต์หรือซับเน็ตต้นทาง (Source) มากที่สุด สาเหตุก็เพราะ Extended ACL ตรวจเช็กได้ละเอียด มันจึงควรเช็กแพ็กเก็ตตั้งแต่เนิ่นๆ หากต้องการฟิลเตอร์ทิ้งจะได้ฟิลเตอร์แต่เนิ่น ๆ เลยไม่ต้องให้เร้าเตอร์หลาย ๆ ตัวมาเสียเวลาประมวลผลแพ็กเก็ตนั้น ๆ

ในการสร้าง ACE แต่ละบรรทัดที่อยู่ภายใน ACL มีหลักการคิด 2 แบบ แบบแรกคือ ให้ PERMT ทราฟฟิกที่ต้องการไว้ก่อน แล้วค่อย DENY ในบรรทัดสุดท้ายหรือจะใช้ Implicit all ทำงานให้ก็ได้ และแบบที่สองคือ ให้ DENY ทราฟฟิกที่ต้องการทราฟฟิกเตอร์ทิ้งแน่นอนออกไปก่อน อย่างเช่น ให้ฟิลเตอร์ทราฟฟิกที่เกี่ยวข้องกับการโจมตีระบบปฏิบัติการของไมโครซอฟท์ เช่น พอร์ต 135,139,455 เป็นต้น แล้วในบรรทัดทุดท้ายของ ACL ให้ใส่ ACE ที่ PERMIT ทราฟฟิกทั้งหมด (permit ip any any) สำหรับแบบที่สองนี้อย่าลืมใส่ Ace ที่ PERMIT ทราฟฟิกทั้งหมดในบรรทัดท้ายด้วยเพราะถ้าหากลืม นั่นจะเท่ากับเรากำลัง DENY ทุกๆ ทราฟฟิกด้วยของ Implicit deny all ท้ายสุด

ถึงแม้ในตอนท้ายของ ACL ทุก ACL จะถือว่าเสมอนมี Implicit deny all อยู่ก็ตาม แต่การใส่ deny ip any any ไว้ที่ตอนท้ายของ ACL จะช่วยให้เราทราบ Hit Count ซึ่งเป็นปริมาณแพ็กเก็ตที่

ถูก Deny ได้ เพราะเมื่อใช้คำสั่ง show access – list ขึ้นมา เราจะเห็น Hit Count ของบรรทัด demy ip any any

8.5 Wildcard Mask

Wildcard Mask เป็นเครื่องมือที่ใช้ในการ “เมนต์ (match)” บิตต่างๆ ในหมายเลขแอดเดรสตามที่ต้องการ โดยค่าบิตที่เป็น 0 ใน Wildcard Mask จะหมายถึงให้ “เมนต์” กับค่าบิตในแอดเดรส ส่วนค่าบิตที่เป็น 1 ใน Wildcard Mask จะหมายถึง ไม่ต้องสนใจ (don’t card) ค่าบิตนั้นในแอดเดรสที่กำลังเปรียบเทียบอยู่

ตัวอย่างเช่น

192.168.30.1 0.0.0.0 จะหมายความว่า เราต้องการ “เมนต์” แอดเดรสทุกแอดเดรสที่มีค่าบิตเท่ากับ 192.168.30.1 ซึ่งก็คือการ “เมนต์” เฉพาะโฮสต์แอดเดรส 192.168.30.1 เพียงแอดเดรสเดียวเท่านั้น

หรืออีกตัวอย่างหนึ่ง

192.168.30.0 0.0.0.255 จะหมายความว่า เราต้องการ “เมนต์” IP Address ทุกแอดเดรสที่มี 3 ไบต์ แรกขึ้นต้นด้วย 192.168.30 ส่วนไบต์สุดท้ายจะเป็นอะไรก็ได้ไม่สนใจ (กล่าวอีกอย่างหนึ่งคือต้องการ “เมนต์”ซับเน็ตแอดเดรส 192.168.30.0/24

Wildcard Mask จะถูกนำมาใช้ใน ACL เพื่อการเมนต์แอดเดรสที่ต้องการ การเมนต์แอดเดรสนี้สำหรับ Extended ACL แล้วจะทำที่แอดเดรสต้นทางหรือปลายทางก็ได้ แต่สำหรับ Standard ACL จะสามารถทำได้เฉพาะแอดเดรสต้นทางเท่านั้น

ลองพิจารณาสักตัวอย่างหนึ่งเพื่อประกอบความเข้าใจ โดยพิจารณาจาก Standard ACL ที่ฟิลเตอร์ได้เฉพาะ Source Address ต้นทางเท่านั้น ดังในตัวอย่างถัดไป

```
HQ (config) #access – list 10 deny 192.168.30.0 0.0.0.255
HQ (config) #int fa0/0
HQ (config-if) #ip access – group 10 in
```

ความหมายของ access – list ข้างต้นก็คือ ให้แพ็กเก็ต IP ทุกแพ็กเก็ตที่วิ่งเข้ามาหาอินเตอร์เฟซ fa0/0 ว่าแพ็กเก็ตไหนบ้างที่แอดเดรสต้นทาง (Source Address) 3 ไบต์ แรกขึ้นต้นด้วย 192.168.30

ส่วนไบต์สุดท้ายเป็นอะไรก็ได้ ถ้าแพ็กเก็ตที่มีลักษณะดังกล่าว ให้เร้าเตอร์ทำการปฏิเสธ (DENY) หรือฟิลเตอร์แพ็กเก็ตดังกล่าวไม่ให้วิ่งผ่าน fa0/0 เข้าไป แพ้กเก็ตนั้นๆ ก้จะถูกโยนทิ้ง (Drop) ไปในที่สุด

มีอีกกรณีหนึ่งที่ควรทำความเข้าใจเพิ่มเติม นั่นคือ การเขียน Wildcard Mask ให้ครอบคลุมแอดเดรสทั้งหมดที่ต้องการ permit หรือ deny โดยเฉพากรณีที่ Wildcard Mask นั้นจำเป็นต้องครอบคลุมซับเน็ตแอดเดรสหลายๆ แอดเดรสพร้อมๆ กัน ตัวอย่างเช่น ต้องการเขียน Wildcard Mask ใน ACL ที่มีการ DENY ซับเน็ตแอดเดรสตั้งแต่ 172.16.16.0 ถึง 172.16.19.0 Wildcard Mask ที่เขียนได้จะเป็นดังนี้

HQ (config) #access – list 20 deny 172.16.16.0 0.0.3.255

ทำไมถึงเป็น 172.16.16.0 0.0.3.255 วิธีคิดแบบละเอียดก็คือ ให้เขียน 172.16.16.0, 172.16.17.0, 172.16.18.0 และ 172.16.19.0 ออกมาในรูปแบบของเลขฐานสอง (ไบนารี) ทั้งหมดแล้วไล่เช็ดูว่าบิตไหนของ 4 ซับเน็ตแอดเดรสนี้ที่ตรงกันทั้งหมด ก็ให้แทนที่ตำแหน่งบิตนั้นใน Wildcard Mask ด้วยเลข 0 ส่วนบิตไหนที่มีค่าบิตไม่ตรงกันทั้งหมดก็ให้แทนที่ตำแหน่งบิตนั้นด้วยเลข 1 แล้วค่อยอ่าน Wildcard Mask ออกมาเป็นเลขฐานสิบ

8.5.1 วิธีสังเกตแบบรวดเร็วที่ง่ายกว่านั้นก็คือ

8.5.1.1 ตำแหน่งของไบต์ที่ 1 และ 2 นั้นจะต้องมี Wildcard Mask เท่ากับ 0 แน่นอนอยู่แล้ว เพราะค่าของทั้ง 2 ไบต์นี้ของทุก ๆ ซับเน็ตแอดเดรสมีค่าเท่ากันคือ 172.16

8.5.1.2 ส่วนตำแหน่งของไบต์ที่ 4 นั้นจะต้องมี Wildcard Mask เท่ากับ 1 เพราะค่าของไบต์ที่ 4 ในทุก ๆ ซับเน็ตแอดเดรสจะเป็นอะไรก็ได้ (don’t care)

8.5.1.3 สำหรับตำแหน่งของไบต์ที่ 3 นั้นให้คิดดังนี้ ให้จำตัวเลข 2,4,8,16,32,64,128 ไว้เลขนี้จะถือเป็นเลข “ระบุขนาดของกลุ่มซับเน็ตแอดเดรส”

8.5.1.4 พิจารณาต่อไปว่า “ขนาดของกลุ่มซับเน็ตแอดเดรส” นี้เป็นเท่าไหร่ จากตัวอย่างข้างต้น ซับเน็ตแอดเดรสทั้งหมดมีขาดเท่ากับ 4 (นับจาก 172.16.16.0 17.0, 18.0, 19.0 เท่ากับ 4 พอดี) ถ้า “ขนาดของกลุ่มซับเน็ตแอดเดรส” มีค่าเป็นอื่นๆ ที่ไม่ใช่ตัวเลขที่ลิสต์ไว้ในข้อที่ 3 ให้ปัดขึ้น เช่น ถ้าขนาดเท่ากับ 20 ให้ถือว่าขนาดของกลุ่มเท่ากับ 32 แล้วค่อยพิจารณาในข้อถัดไป

8.5.1.5 ตัวเลขใน Wildcard Mask ตำแหน่งไบต์ที่สามจะมีค่าเท่ากับ “ขนาดของกลุ่ม” ลบด้วย 1,4 – 1=3 ดังนั้น ค่าของ Wildcard Mask ทั้งหมดที่คำนวณได้จะเท่ากับ 0.0.3.255 และถ้าเขียนเต็มๆ ใน access – list จะเป็น ซับเน็ตแอดเดรสเริ่มต้นแล้วตามด้วยค่า Wildcard Mask ที่คำนวณได้ข้างต้น นั่นคือ 172.16.16.0 0.0.3.255

ลองพิจารณาตัวอย่างต่อไปนี้

1) HQ (config) #access – list 20 deny 172.16.16.0 0.0.7.255 ตัวอย่างข้างต้นเป็นการสั่งให้ DENY ซับเน็ตแอดเดรส 172.16.16.0 – 172.16.23.0 (16.0,17.0,18.0,19.0,20.0,21.0,22.0,23.0) สังเกตว่า “ขนาดของกลุ่มซับเน็ตแอดเดรส” จะเป็น 8 ไบต์ที่สามจึงเป็น 8 – 1 = 7 สังเกตว่า ซับเน็ตแอดเดรสทั้งหมดนี้จะมีจำนวนบิตที่เหมือนกันทั้งหมด 21 บิต

2) HQ (config) #access – list 30 deny 172.16.32.0 0.0.31.255 ตัวอย่างข้างต้นเป็นการสั่งให้ DENY ซับเน็ตแอดเดรส 172.16.32.0 – 172.16.63.0 สังเกตว่า “ขนาดของกลุ่มซับเน็ตแอดเดรส” จะเป็น 32 (วิธีการหา ให้นำเอา 63 ลบด้วย 32 แล้ว บวกอีก 1 ได้เท่ากับ 32 ไบต์ที่สามจึงเป็น 32 – 1 = 31 สังเกตว่า ซับเน็ตแอดเดรสทั้งหมดนี้จจะมีจำนวนบิตที่เหมือนกันทั้งหมด 19 บิต

3) HQ (config) #access – list 40 deny 172.16.64.0 0.0.63.255 ตัวอย่างข้างต้นเป็นการสั่งให้ DENY ซับเน็ตแอดเดรส 172.16.64.0 – 172.16.127.0 สังเกตว่า “ขนาดของกลุ่มซับแอดเดรส” จะเป็น 64 ไบต์ที่สามจึงเป็น 64 – 1 = 63 สังเกตว่า ซับเน็ตแอดเดรสทั้งหมดนี้จะมีจำนวนบิตที่เหมือนกันทั้งหมด 18 บิต

8.6 Standard Access Control Lists

Standard ACL สามารถอนุญาตหรือฟิลเตอร์ทราฟฟิกได้โดยการพิจารณาจาก Source IP Address ในแพ็กเก็ต วิธีการสร้าง Standard ACL ก็คือ การสร้าง access – list ที่มีหมายเลขประจำ ACL เป็นค่าตัวเลขระหว่าง 1 -99 หรือ 1300 – 1999 หมายเลขของ ACL จะเป็นตัวบ่งบอกว่า ACL ที่ถูกสร้างขึ้นมาจัดอยู่ในประเภทไหน หลังจาก ที่ระบุหมายเลข ACL ลงไปเป็น 1 – 99 หรือ 1300 – 1999 , IOS จะรู้ได้ในทันที่ว่าเรากำลังสร้าง Standard ACL อยู่

ลองดูรูปแบบการระบุคำสั่งและพารามิเตอร์ต่างๆ ใน Standard ACL

```
HQ(config) #access – list ?
<1 -99> IP standard access list
<100 - 199> IP extended access list
<1000 - 1099> IPX SAP access list
<1100 - 1199> Extended 48 – bit MAC address access list
<1200 -1299> IPX summary address access list
```

<1300 - 1990> IP standard access list (expanded range)

<200 - 299> Protocol type – code access list

<2000 - 2699> IP extended access list (expanded range)

<300 - 399> DECnet access list

<400 – 499> XNS standard access list

<500 – 599> XNS extended access list

<600 – 699> Appletalk access list

<700 – 799> 48 – bit MAC address access list

<800 – 899> IPX standard access list

<900 – 999> IPX extended access list

Dynamic – extended Extend the dynamic ACL absolute timer

Rate – limit simple rate – limit specific access list

ลองคีย์เลข 10 (อยู่ในช่วง 1 – 99) แล้วพิมพ์เครื่องหมายคำถาม (?) , IOS จะรู้ทันทีว่าเรากำลังสร้าง Standard ACL พารามิเตอร์ถัดไปจึงเป็นพารามิเตอร์ที่เหมาะสมสำหรับ Standard ACL

HQ(config) #access – list 10 ?

Deny specify packets to reject ระบุว่าต้องการ DENY แพ็กเก็ตที่อยู่ในเงื่อนไขที่ตามหลัง

Permit Specify packets to forward ระบุว่าต้องการ PERMIT แพ็กเก็ตที่อยู่ในเงื่อนไขที่ตามหลัง

Remark Access list entry comment ต้องการใส่คำอธิบายไปใน access – list ที่สร้างขึ้น

ลองเซตคีย์เวิร์ดเป็น Deny

HQ(config) #access – list 10 deny ?

Hostname ro A.B.C.D Address to match สามารถระบุซับเน็ตแอดเดรสต้นทางพร้อมทั้ง Wildcard Mask ได้ในที่นี้

Any Any source host ระบุว่าแอดเดรสต้นทางเป็นอะไรก็ได้

Host A single host address ระบุแอดเดรสต้นทางเป็นโฮสต์แอดเดรส

ลองดูตัวอย่างถัดไป

1) HQ(config)#access – list 10 deny host 192.168.30.2 เป็นการ DENY โฮสต์แอดเดรส 192.168.30.2
2) HQ(config)#access – list 20 prmit any เป็นการ PERMIT ทุก ๆ แอดเดรส

3) HQ(config)#access – list 30 deny 192.168.0.0 0.0.255.255 เป็นการ DENY ทุกๆ แพ็กเก็ตที่มี Source IP Address 2 ไบต์แรกขึ้นต้นด้วย 192.168 ส่วนอีก 2 ไบต์สุดท้ายเป็นอะไรก็ได้

8.7 Extended Access Control Lists

Extended ACL เป็น ACL ที่เปิดโอกาสให้ผู้ใช้งานสามารถระบุเงื่อนไขได้อย่างละเอียดมากขึ้น การ PERNIT หรือ DENY แพ็กเก็ต ได้แก่

1) Source Address เป็นได้ทั้งซับเน็ตแอดเดรส, ซับเน็ตแอดเดรสพร้อมด้วย Wildcard Mask, โฮสต์แอดเดรส

2) Destination Address เป็นได้ทั้งซับเน็ตแอดเดรส, ซับเน็ตแอดเดรสพร้อมด้วย Wildcard Mask, โฮสต์แอดเดรส

3) Protocol Field เป็นได้หลายแบบทั้ง eigrp, gre, icmp, igmp,grp,ip,ospf,udp

4) Source Port เป็นหมายเลขพอร์ตต้นทางที่อยู่ในส่วนแฮดเดอร์ของเลเยอร์ 4 (TCP/UDP) ในแพ็กเก็ต IP

5) Destination Port เป็นหมายเลขพอร์ตปลายทางอยู่ในส่วนแฮดเดอร์ของสเลเยอร์ 4 (TCP/UDP) ในแพ็กเก็ต IP

6) IP Type of Service (TOS)

7) IP Precedence

8) แฟลกของ TCP เช่น SYN, ACK

สำหรับ Extended ACL หมายเลข ACL จะเป็นค่าตั้งแต่ 100 -199 และ 2000 – 2699 ในขณะที่เราคีย์คำสั่ง access – list แล้วตามด้วยตัวเลขในช่วงข้างดังกล่าว IOS จะรู้โดยทันทีว่าเรากำลังสร้าง Extended ACL อยู่ ดังในตัวอย่างถัดไป

HQ(config) #access – list 100 ?

Deny specify packets to reject

Dynamic Specify a DYNAMIC list of PERMITs or DENYs

Permit Specify packets to forward

Remark Access list entry comment

ในเอาต์พุตถัดไป ลองพิมพ์ access – list 100 deny แล้วตามด้วยเครื่องหมายคำถาม พารามิเตอร์ถัดไปที่พบคือ ฟิลด์ Protocol ซึ่งอยู่ในส่วนแฮดเดอร์ของแพ้กเก็ต IP

HQ (config) #access – list 100 deny ?

<0 – 255> An IP protocol number

Ahp Authentication Header Protocol

Eigrp Cisco's GRE EIGRP routing protocol

Esp Encapsulation Security Payload

Gre Cisco's GRE tunneling

I cmp Internet Control Massage Protocol

I gmp Internet Gateway Massage Protocol

Igrp Cisco's IGRP routing Protocol

Ip Any Internet Protocol

Ipinip IP in IP tunneling

Nos KA9Q NOS compatible IP over IP tunneling

Ospf OSPF routing protocol

Pcp Payload compression Protocol

Pim Protocol Independent Multicast

Tcp Transmission control Protocol

Udp User datagram Protocol

ในเอาต์พุตถัดไป เลือกฟิลด์ Protocol เป็น TCP แล้วลองคีย์เครื่องหมาย ? เพื่อสำรวจดูพารามิเตอร์ถัดไป ซึ่งเป็น Source Address

HQ (config) #access – list 100 deny tcp ?

A.B.C.D Source address

Any Any Source host

Host A single source host

ในเอาต์พุตถัดไป ผู้เขียนใช้ 192.168.30.0 0.0.0.255 เป็น Source Address ต้นทางและคีย์เครื่องหมาย ? เพื่อสำรวจดูพารามิเตอร์ถัดไป ซึ่งเป็นได้ทั้งหมายเลขพอร์ตต้นทาง (Source Port) หรือพารามิเตอร์ Destination Address ปลายทาง

HQ (config)#access – list 100 deny tcp 192.168.30.0 0.0.0.255 ?

A.B.C.D Destination address

Any Any Destination host

Eq Match only packets on a given port number

Gt Match only packets with a greater port number

Host A single Destination host

It Match only packets with a lower port number

Nep Match only packets not on a given port number

Range Match only packets in the range of port numbers

ในเอาต์พุตถัดไป ผู้เขียนใช้เป็น Destination Address ซึ่งเท่ากับ any พารามิเตอร์ any หมายถึงแอดเดรสใดๆ ก็ได้ แล้วพิมพ์เครื่องหมาย ? เพื่อสำรวจดูพารามิเตอร์ถัดไปซึ่งมีอยู่ด้วยกันหลายรูปแบบ

HQ (config)#access – list 100 deny tcp 192.168.30.0 0.0.0.255 any ?

Ack Match on the ACK bit

Dscp Match packets with given dscp value

Eq Match only packets on a given port number

Established Match established connections

Fin Match on the FIN bit

Fragments check non – initial fragments

Gt Match only packets with a greater port number

Log Log matches against this entry

Log – input Lon matches against this entry, including input interface

It 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

Psh Match on the PSH bit range Match only packets in the range of port numbers

Rst Match on the RST bit

Syn Match on the SYN bit

Time – range specify a time – range

Tos Match packets with given TOS value

Urg Match on the URG bit

ในเอาต์พุตถัดไป ในที่นี้เราสนใจก็คือ คีย์เวิร์ก eq ซึ่งใช้ระบุหมายเลขพอร์ตปลายทาง (destination port) ลองพิมพ์ eq แล้วพิมพ์เครื่องหมาย ? เพื่อสำรวจดูว่ามีหมายเลขพอร์ตใดบ้าง เราจะพบว่าหมายเลขพอร์ตที่ใส่ได้จำนวนมาก

HQ (congig) #access – list 100 deny tcp 192.168.30.0 0.0.0255 any eq?

<0 – 65535> Port number

Bgp Borer Gateway Protocol (179)

Chargen Character generator (19)

Cmd Remote commands (rcmd, 514)

Daytime Daytime (13)

Discard Discard (9)

Domain Domain Name Service (53)

Echo Echo (7)

Exec Exec (rsh, 512)

Finger Finger (79)

ftp File Transfer Protocol (21)

ftp – data FTP data connections (20)

gopher Gopher (70)

hostname NTC hostname server (101)

ident Ident Protocol (113)

irc Internet Relay Chat (194)

klogin Kerberos login (543)

kshe11 kerberos shell (544)

lonig Login (rlogin, 513)

lpd Printer service (515)

nntp Network News Transport Protocol (119)

pim – auto – rp PIM Auto – RP (496)

(ตัดเอาต์พุต)

ในที่นี้ ใช้ eq 80 ซึ่งหมายถึงหมายเลขพอร์ตปลายทางเป็น 80 ดังแสดง

HQ (config) #access – list 100 deny tcp 192.168.30.0 0.0.0.255 any eq 80

Access – list ข้างต้นเป็นการฟิลเตอร์หรือปฏิเสธทราฟฟิกที่มีแอดเดรสต้นทางขึ้นต้นด้วย 192.168.30 (ส่วนไบต์ที่ 4 เป็นอะไรก็ได้) และส่งไปยังแอดเดรสปลายทางเป็นอะไรก็ได้ที่มีพอร์ตปลายทางเป็นหมายเลข 80 (พอร์ตของ www)

ถัดจากนั้น ผู้เขียนเพิ่ม ACE อื่นๆ เข้าไปใน access – list 100 เพิ่มเติม ดังนี้

HQ(config)#access – list 100 permit tcp 192.168.30.0 0.0.0.255 any eq 110

HQ(config)#access – list 100 permit tcp 192.168.30.00.0.0.255 any eq 25

HQ(config)#access – list 100 permark Used to allow mail traffic

ทั้งสองบรรทัดเป็นการอนุญาตทราฟฟิกกจากซับเน็ตแอดเดรส 192.168.30.0 ไปยังปลายทางใด ๆ ก็ได้โดยมีพอร์ตปลายทางเท่ากับ 100 (pop3) และเท่ากับ 25 (smtp) ในที่นี้ผู้เขียนใส่คำอธิบายของ ACL นี้ไว้ด้วยโดยใช้คำสั่ง access – list remark

ส่วนทราฟฟิกประเภทอื่นๆ จะถูกฟิตเตอร์ทิ้งไปโดยดีฟอล์ ด้วยผลของ Implicit deny all ที่อยู่ท้าย ACL ทุกๆ ACL ถึงแม้เราจะไม่ได้สร้างไว้ก็ตาม

8.8 Named Access Control Lists

หลักในการทำงานของ Named ACL นั้นเหมือนกันทุกประการกับ Standard ACL และ Extended ACL ต่างกันตรงที่เราสามารถตั้งชื่อให้กับ ACL ได้ ข้อดีของ Named ACL ได้แก่

1) สะดวกและง่ายต่อการจดจำ
2) สามารถลบเฉพาะบาง ACE ที่ต้องการได้

คำสั่งที่ใช้ในการสร้าง Named ACL แบบ Standard คือ ip access – list standard ส่วนคำสั่งที่ใช้ในการสร้าง Named ACL แบบ Extended คือ ip access – list standard ดังแสดงในตัวอย่างถัดไป และคำสั่งที่ใช้ในการบังคับใช้ Named ACL ที่สร้างไว้ก็คือ ip access – group <ชื่อของ Named ACL > <in/out>

ตัวอย่างถัดไป แสดงการสร้างและบังคับใช้ Named ACL แบบ Extended สังเกตว่าภายใต้โหมดของการสร้าง Named ACL เราพิมพ์ ACE ที่ต้องการโดยเริ่มต้นด้วย permit หรือ deny ได้เลย โดยไม่ต้องพิมพ์คำว่า access – list ขึ้นต้นเหมือนอย่างที่ผ่านมา

HQ(config) #ip access – list ?

Extended Extended Access List

Log – update control access list log updates

Loging control access list logging

Standard standard Access List

HQ(config) #ip access – list extended BlockWorm พิมพ์ ip accress – list extended ตามด้วยชื่อและเคาะคีย์ enter

HQ(config –ext – nacl) #deny tcp any any eq 135 ใส่ deny หรือ permit ได้เลยแล้วตามด้วยพารามิเตอร์แบบเดิม

HQ(config –ext – nacl) #deny tcp any any eq 139

HQ(config –ext – nacl) #deny tcp any any eq 445

HQ(config –ext – nacl) #permit ip any any บรรทัดสุดท้ายจะ permit ทราฟฟิกอื่นๆ ทั้งหมด

HQ(config –ext – nacl) #exit

HQ(config)#int s0/1

HQ(config – if) #ip access – group BlockWorm in บังคับใช้ Named ACL บนอินเตอร์เฟซ S0/1 ในทิศทางขาเข้า (inbound)

ในลักษณะข้างต้น ผู้เขียนได้เซต DENY ทราฟฟิกที่ไม่ต้องต้องการออกไปก่อน แล้วจากนั้นจึงค่อย PERMIT ทราฟฟิกอื่น ๆ ที่เหลือ จากนั้นให้บังคับใช้ Named ACI ดังกล่าวบนอินเตอร์เฟซ S0/1 ในทิศทาง inbound

HQ#sh access – list BlockWorm

Extended IP access list BlockWorm

deny tcp any any eq 135

deny tcp any any eq 139

deny tcp any any eq 445

permit ip any any

หากต้องการยกเลิกบาง ACE ออกไปจาก Named ACL ชื่อ BlockWorm ให้เข้าสู่โหมดของการสร้างอีกครั้งแล้วพิมพ์คำสั่ง NO ตามด้วย ACE เดิมเช่น

HQ(config)#ip access – list extended BlockWorm

HQ(congig – ext – nacl) #no deny tcp any any eq 139

HQ(congig – ext – nacl) #exit

HQ(congig) #exit

HQ#sh access – list BlockWorm

Extended IP access list BlockWorm

Deny tcp any any eq 135

Deny tcp any any eq 445 บรรทัด deny tcp any any eq 139 ถูกยกเลิกออกไปแล้ว

Permit ip any any

ตัวอย่างถัดไป แสดงการสร้างและบังคับใช้ Named ACL แบบ Standard

HQ(config)#ip access – list standard onlyPrivate

HQ (comfig – std – nacl)#permit ?

Hostname or A.B.C.D Address to match

Any Any source host

Host A single host address

HQ(config – std – nacl)#permit 192.168.0.0 0.0.255.255

HQ(config – std – nacl)#exit

HQ(config)#int fa0/0

HQ(config)#ip access – group OnlyPrivate in

คอนฟิกูเรชันข้างต้นเป็นการสร้าง Named ACL แบบ Standard เพื่อบังคับใช้อินเตอร์เฟซ FA0/0 ในทิศทางขาเข้า โดยอนุญาตเฉพาะแพ็กเก็ตที่มี IP Address ต้นทางขึ้นต้นด้วย 192.168 เท่านั้น

คำสั่ง SHOW ที่เกี่ยวข้องกับ ACL

SHOW ACCESS – LIST

HQ#sh access – list

Standard IP access list 10

Deny 192.168.30.2

Deny 192.168.30.0, wildcard bits 0.0.0.255 (779 matches)

Standard IP access list 40

Deny 172.16.64.0, wildcard bits 0.0.63.255

Extended IP access list 100

Deny tcp 192.168.30.0 0.0..255 any eq www

Permit tcp 192.168.30.0 0.0.0.255 any eq pop3

Permit tcp 192.168.30.0 0.0.0.255 any eq smtp

Extended IP access list 101

Permit ip any any (15 matches)

Permit tcp any any eq www

SHOW ACCESS – LIST <ACL Number>

ใช้ในการแสดงเฉพาะรายละเอียดของ ACL ที่สนใจ ทั้งคำสั่ง show access – list และ show access – list <number> จะมีประโยชน์อีกอย่างหนึ่งก็คือ มันจะแสดงค่าสถิติที่เรียกว่า hit count ขึ้นมาด้วย ซึ่งเป็นค่าที่บ่งบอกว่ามีปริมาณแพ็กเก็ตที่สอดคล้องกับเงื่อนไขในแต่ละ ACE มากน้อยเพียงใด

SHOW IP INTERFACE <interface Number>

หากต้องการดูว่า ขณะนี้บนอินเตอร์เฟซที่สนใจมี access – list บังคับใช้อยู่หรือไม่ คำสั่งหนึ่งที่ใช้ได้คือ show run แล้วตามด้วยอินเตอร์เฟซนั้นๆ เช่น sh run int fa0/0

นอกจากนั้น ยังมีอีกคำสั่งหนึ่งที่ใช้ได้ก็คือ show ip interface <interface number>

HQ#sh ip int fa0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 192.168.40.1/24

Broadcast address is 255.255.255.255

Address determined by setup command

MTU is 1500 bytes

Helper address is not set

Directed broadcast forwarding is disabled

Multicast reserved groups joined : 224.0.0.5 244.0.0.6

Outgoing access list is not set

Inbound access list is 10

8.9 การใช้ ACL เพื่อควบคุมการข้าถึง Line VTY

ACL สามารถถุกนำมาใช้เพื่อควบคุมการเทลเน็ตเข้าสู่เร้าเตอร์เองผ่านทาง Line VTY ได้ด้วยการกำหนดหมายเลข IP Address หรือซับเน็ตแอดเดรสต้นทางที่สามารถเทลเน็ตเข้าสู่เร้าเตอร์ได้ ขั้นตอนมีดังนี้

1) สร้าง Standard ACL ขึ้นมาเพื่ออนุญาต (permit) เฉพาะ IP Address ที่เหมาะสม
2) บังคับใช้ Standard ACL ข้างต้นเข้ากับ Line VTY ด้วยคำสั่ง access – class in

HQ(config)#access – list 20 permit 192.168.30.2 อนุญาตเฉพาะ IP Address ที่เหมาะสม

HQ(config)#access – list 20 permit 192.168.30.3 อนุญาตเฉพาะ IP Address ทีเหมาะสม

HQ(config)#line vty 0 4

HQ(config - line)#access – class 20 in แมป ACL หมายเลข 20 ข้างต้นเข้ากับ line VTY ผ่านทาง

คำสั่ง access – class <acl number> in เพื่ออนุญาตเฉพาะ 2 แอดเดรสข้าต้นเท่านั้นให้เทลเน็ตเข้ามาได้

8.10 เปรียบเทียบ ACL กับไฟร์วอลล์

ถึงแม้ ACL จะเป็นเครื่องมือที่ดีในการสร้ามาตรการรักษาความปลอดภัย ถ้าหากเน็ตเวิร์กของเราต้องการอุปกรณ์สำหรับทำหน้าที่ในการครบคุมทราฟฟิกที่วิ่งผ่านเข้าออกอย่างจริงจังโดยเฉพาะทราฟฟิกที่รับส่งกับเครือข่ายอินเทอร์เน็ตและต้องป้องกันเซิร์ฟเวอร์สาธารณะที่เปิดให้บริการภายนอกด้วยไฟร์วอลล์น่าจะเป็นอีกทางเลือกที่เหมาะสมกว่า

ACL บนเร้าเตอร์ควรถูกนำมาใช้เป็นมาตรการที่ช่วยเสริมความปลอดภัยอีกชั้นหนึ่ง คือช่วยกรองแพ็กเก็ตบางจำพวกออกไปก่อน ก่อนที่แพ็กเก็ตเหล่านั้นจะผ่านมายังไฟร์วอลล์ อละควรถูกใช้ในเน็ตเวิร์กภายในที่ต้องการการควบคุมทราฟฟิกระหว่างซับเน็ตต่างๆ หรือระหว่าง VLAN ที่อธิบานได้ข้างต้น มันอาจไม่เหมาะนักหากจะนำเอา “เร้าเตอร์” มอคอนฟิก ACL แล้วให้ทำหน้าที่เสมือนหนึ่งเป็น “ไฟร์วอลล์”

เพราะหน้าที่หลักจริงๆ ของเร้าเตอร์คือการทำงานด้านเร้าติ้ง อย่างไรก็ดี เร้าเตอร์รุ่นใหม่และ IOS ใหม่ ๆ ของซิสโก้ในปัจจุบันได้ผนวกรวมเอาฟีเจอร์ด้านความปลอดภัยต่างๆ มากมายนอกเหนือจาก ACL เข้ามาไว้ภายในตัวเสร็จสรรพ เพื่อช่วยเสริมสร้างความปลอดภัยให้กับโครงสร้างพื้นฐานด้านเครือข่ายขององค์กรธุรกิจทั่วไปให้แข็งแกร่งยิ่งขึ้น

บทสรุป

Access Control Lists (ACL) ซึ่งเป็นฟีเจอร์หนึ่งบน IOS ของซิสโก้ที่ช่วยในการสร้างความปลอดภัยกับระบบเน็ตเวิร์กและเร้าเตอร์ Wildcard Mask เป็นเครื่องมือที่ใช้ในการ “เมนต์ (match)” บิตต่าง ๆ ในหมายเลขแอดเดรสตามที่ต้องการ

ACL สามารถบังคับใช้ทั้งในลักษณะ Inbound และในลักษณะ Outbound

Standard ACL สามารถอนุญาตหรือฟิลเตอร์ทราฟฟิกได้โดยการพิจารณาจาก Source IP Address ในแพ็กเก็ต

Extended ACL เป็น ACL ที่เปิดโอกาสให้ผู้ใช้งานสามารถระบุเงื่อนไขได้อย่างละเอียดมากขึ้น การ PERNIT หรือ DENY แพ็กเก็ต

หลักในการทำงานของ Named ACL นั้นเหมือนกันทุกประการกับ Standard ACL และ Extended ACL ต่างกันตรงที่เราสามารถตั้งชื่อให้กับ ACL ได้

แบบฝึกหัด

จงตอบคำถามต่อไปนี้มาพอสังเขป

1. อธิบายประโยชน์ที่ได้รับจากการทำ ACL?
2. ACL สามารถใช้งานอย่างไรได้บ้าง?
3. ACLและ Wildcard Mask มีความสัมพันธ์กันอย่างไร?
4. ไฟล์วอลล์คืออะไร?
5. ACL กับ ไฟล์วอลล์แตกต่างกันอย่างไร?

อ้างอิง

เอกสิทธิ์ วิริยจารี .(2548). **เรียนรู้ระบบเครือข่ายจากอุปกรณ์ของ Cisco ภาคปฏิบัติ** กรุงเทพฯ .: ซีเอ็ดยูเคชั่น

จตุชัย แพงจันทร์ (2555). **เจาะระบบ Network 3nd Edition**. นนทบุรี: ไอดีซีฯ

ซิสโก้ .(2548). **Cisco Networking Academy Program CCNA 2**. กรุงเทพฯ : เพียร์สัน เอ็ดดูเคชั่นอินโดไซน่า

jodoi. (n.d). **ACL on CISSO Router**. Retrieved Jun 23, 2017, from http://jodoi.org/ACL.html

บรรณานุกรม

My reading room. (n.d). **Components of data communication.** Retrieved Nov 20, 2016, from http://www.myreadingroom.co.in/notes-and-studymaterial/68-dcn/675-components-of-data-communication.html

Cisco. (n.d). **Networking Fundamentals.** Retrieved Nov 20, 2016, from http://www.cisco.com/c/dam/global/fi_fi/assets/docs/SMB_University_120307_Networking_Fundamentals.pdf

Fossbytes. (n.d). **Network Topology Explained.** Retrieved Nov 20, 2016, from https://fossbytes.com/what-is-ring-topology-advantages-and-disadvantages-of-ring-topology/

Cerna Glenn. (n.d). **Physical topology**. Retrieved Nov 20, 2016, from http://it11delacernagjd.blogspot.com/p/topology-there-are-two-basic-categories.html

Ciao Systems. (n.d). **Setting Up a Small Business Network in 5 Simple Steps**. Retrieved Feb 20, 2017, from http://ciaosystems.com/setting-small-business-network-5-simple-steps/

จตุชัย แพงจันทร์) .2555). **เจาะระบบ Network 3nd Edition**. นนทบุรี: ไอดีซีฯ

สุชาติ คุ้มมะณี .ธวัชชัย ชมศิริ ,(2550). **เรียนรู้เครือข่ายและอุปกรณ์ Cisco ด้วยโปรแกรม Simulation**. กรุงเทพฯ : โปรวิชั่น

เอกสิทธิ์ วิริยจารี .(2548). **เรียนรู้ระบบเครือข่ายจากอุปกรณ์ของ Cisco ภาคปฏิบัติ** กรุงเทพฯ .: ซีเอ็ดยูเคชั่น

ซิสโก้ .(2548). **Cisco Networking Academy Program CCNA 2**. กรุงเทพฯ : เพียร์สัน เอ็ดดูเคชั่น อินโดไซน่า

จีเนียสย์ ดีเวลลอป.(ม.ป.ป.). **ความแตกต่างของ SWITCH และ HUB**. สืบค้นเมื่อ 10 ธันวาคม 2558, สืบค้นจาก https://www.ez-genius.com/index.php/presets/network-security/80-switch-and-hub

บรรณานุกรม (ต่อ)

QRZ Now (n.d). **50 OHM PROFESSIONAL COAXIAL CABLES.** Retrieved Feb 10, 2016, from http://qrznow.com/50-ohm-professional-coaxial-cables/

Powerclub Thailand. (ม.ป.ป.). **Hub กับ Switch แตกต่างกันอย่างไร?.** สืบค้นเมื่อ 10 ธันวาคม 2558, สืบค้นจาก http://powerclub-thailand.com/contents/Articles/hub-switch/hub-switch.html

Emily Gagne (n.d). **Most Commonly Used Computer Networking Cables.** Retrieved Feb 10, 2016, from http://letsblogblogger.blogspot.com/2011/07/most-commonly-used-computer-networking.html

เทเลพาร์ท คอร์ปอเรชั่น ซัพพลาย. (ม.ป.ป.). **Fiber Optic Cables.** สืบค้นเมื่อ 10 ธันวาคม 2558, สืบค้นจาก http://www.telepart.net/ไฟเบอร์ออฟติกFiber-Optic

Orbitco. (2558). **What is CIDR? Explained with Examples.** Retrieved Jun 23, 2017, from http://www.orbit-computer-solutions.com/classless-interdomain-routing-cidr-explained/

Sysnet center. (ม.ป.ป.). **การ Config อุปกรณ์ Air-Live WL-5460AP ใน Mode Access Point (AP)**. สืบค้นเมื่อ 10 ธันวาคม 2558, สืบค้นจาก http://www.sysnetcenter.com/board /index.php? topic=216.0

Stretch. (2553). **Basic Private VLAN Configuration**. Retrieved Jun 23, 2017, from http://packetlife.net/blog/2010/aug/30/basic-private-vlan-configuration/

Lisans.cosum. (n.d). **Frame Relay**. Retrieved Jun 23, 2017, from http://lisans.cozum.info.tr/networking/bilgisayar_aglari/www.protocols.com/Protocol%20Directory%20-%20Frame%20Relay.htm

jodoi. (n.d). **ACL on CISSO Router**. Retrieved Jun 23, 2017, from http://jodoi.org/ACL.html