คำสั่ง `update-source` ใน BGP configuration มีความสำคัญมากครับ เพราะ:

1. หน้าที่หลักของ update-source:
   - เป็นการบอกให้ BGP ใช้ IP address ของ interface ที่ระบุ (ในที่นี้คือ loopback) เป็น source IP ในการสร้าง BGP session
   - ถ้าไม่ระบุ update-source BGP จะใช้ IP ของ outgoing interface เป็น source IP โดยอัตโนมัติ

2. ในตัวอย่างของคุณ:
   ```
   R1(config-router)#neighbor 2.2.2.2 update-source lo0
   R2(config-router)#neighbor 1.1.1.1 update-source lo0
   ```
   - R1 บอกว่า "เวลาจะติดต่อกับ peer 2.2.2.2 ให้ใช้ IP ของ loopback เป็น source"
   - ถ้าไม่มีคำสั่งนี้ R1 จะใช้ IP 12.1.1.1 เป็น source แทน

3. ทำไมต้องระบุ:
   - เพื่อให้ BGP session ใช้ loopback IP เป็นทั้ง source และ destination
   - ทำให้ session มีเสถียรภาพ ไม่ขึ้นกับ physical interface ใดๆ
   - หากมีหลายเส้นทางระหว่าง routers BGP session จะไม่ล่ม ตราบใดที่ยังมีเส้นทางไปถึง loopback ได้

4. สรุป flow การทำงาน:
   - Router จะมอง peer IP (1.1.1.1/2.2.2.2) เป็น destination
   - update-source บอกให้ใช้ loopback IP เป็น source
   - ทั้งสองฝั่งต้องมี route ไปหา loopback ของกันและกันผ่าน IGP หรือ static route

คำสั่ง `disable-connected-check` ใช้ในกรณีที่เราต้องการให้ BGP peers สามารถจับคู่กันได้ แม้ว่าจะไม่ได้อยู่ใน directly connected network เดียวกัน

ในกรณีปกติ BGP จะตรวจสอบว่า:
1. peers ต้องอยู่ใน directly connected network หรือ
2. มี route ไปถึง next-hop address ของ peer

แต่เมื่อใช้ `disable-connected-check`:
- BGP จะข้ามการตรวจสอบนี้ไป
- ทำให้สามารถสร้าง BGP session ได้แม้ peers จะอยู่คนละ network

- มักใช้ในกรณีที่เรา peer ผ่าน loopback interfaces

เหตุผลที่ต้องใช้ในตัวอย่างนี้:
- R1 พยายาม peer กับ 2.2.2.2 (loopback ของ R2)
- R2 พยายาม peer กับ 1.1.1.1 (loopback ของ R1)
- IP เหล่านี้ไม่ได้อยู่ใน directly connected network
- จึงต้องใช้ `disable-connected-check` เพื่อให้ BGP session สามารถขึ้นได้

ดังนั้นคำสั่งนี้มักจะใช้คู่กับการ peer ผ่าน loopback interfaces เพื่อให้ BGP session สามารถสร้างได้สำเร็จ

จากภาพอธิบายให้เห็นประเด็นสำคัญเกี่ยวกับการใช้ ebgp-multihop ดังนี้:

1. ปัญหาที่เกิดขึ้น:
   - เมื่อลบ disable-connected-check ออกและใช้ ebgp-multihop 1
   - BGP session ไม่สามารถขึ้นได้ แสดง error "no route to peer"
   - เพราะ router ยังคงทำ connected check อยู่

2. พฤติกรรมพิเศษของ ebgp-multihop:
   - เมื่อตั้งค่า ebgp-multihop = 1: router ยังทำ connected check
   - เมื่อตั้งค่า ebgp-multihop > 1: router จะปิด connected check โดยอัตโนมัติ
   - นี่เป็น hidden feature ที่สำคัญของคำสั่ง ebgp-multihop

3. การแก้ไข:
   ```
   R1(config-router)#neighbor 2.2.2.2 ebgp-multihop 2
   R2(config-router)#neighbor 1.1.1.1 ebgp-multihop 2
   ```
   - เมื่อเปลี่ยนค่าเป็น 2 BGP session สามารถขึ้นได้
   - สังเกตจาก packet capture จะเห็น TTL = 2 ตามที่กำหนด

4. ข้อสรุปสำคัญ:
   - การตั้ง ebgp-multihop > 1 ไม่ได้เกี่ยวกับระยะทางจริงระหว่าง peer
   - แต่เป็นการบอก BGP ว่าให้ยอมรับ peer ที่ไม่ได้ connected โดยตรง
   - และเป็นการกำหนด TTL ของ BGP packets ด้วย

นี่เป็นเทคนิคที่สำคัญในการ config BGP เมื่อต้องการ peer ผ่าน loopback โดยไม่ต้องใช้ disable-connected-check

อธิบายให้ชัดขึ้นนะครับ เมื่อเรา peer BGP ผ่าน loopback มีสองวิธีที่จะทำให้ session ขึ้นได้:

1. วิธีที่ 1: ใช้ disable-connected-check
```
R1(config)#router bgp 100
R1(config-router)#neighbor 2.2.2.2 remote-as 200
R1(config-router)#neighbor 2.2.2.2 update-source lo0
R1(config-router)#neighbor 2.2.2.2 disable-connected-check
```

2. วิธีที่ 2: ใช้ ebgp-multihop ที่มีค่ามากกว่า 1
```
R1(config)#router bgp 100
R1(config-router)#neighbor 2.2.2.2 remote-as 200
R1(config-router)#neighbor 2.2.2.2 update-source lo0
R1(config-router)#neighbor 2.2.2.2 ebgp-multihop 2
```

ทั้งสองวิธีใช้ได้เหมือนกัน เพราะ:
- วิธีที่ 1: สั่งปิด connected check ตรงๆ
- วิธีที่ 2: เมื่อตั้ง ebgp-multihop > 1 มันจะปิด connected check ให้โดยอัตโนมัติ

แต่ถ้าเราใช้แค่:
```
R1(config-router)#neighbor 2.2.2.2 remote-as 200
R1(config-router)#neighbor 2.2.2.2 update-source lo0
```
Session จะไม่ขึ้น เพราะ BGP ยังทำ connected check และเห็นว่า peer address (loopback) ไม่ได้อยู่ใน directly connected network

ต้องใช้วิธีใดวิธีหนึ่งจาก 2 วิธีข้างต้นเพื่อปิด connected check ครับ

อธิบายแบบเข้าใจง่ายๆ ว่าทำไม GTSM ถึงช่วยเรื่อง security ได้:

1. ปกติการโจมตี BGP:
   - แฮกเกอร์สามารถปลอมแปลง IP source address เป็น BGP peer ได้

- ส่ง fake BGP packets มาเพื่อรบกวนหรือพยายามสร้าง BGP session
- ถ้าส่งมาเยอะๆ จะทำให้ router's CPU ทำงานหนัก (DoS attack)

2. GTSM ป้องกันโดยใช้หลักการของ TTL:
- Router ที่เป็น BGP peer ตัวจริง: ส่ง packet มาด้วย TTL=255
- เมื่อ packet ผ่าน router แต่ละตัว TTL จะลดลง 1
- ถ้าเป็น direct neighbor ที่อยู่ห่าง 1 hop, TTL จะเหลือ 254
- ถ้าอยู่ห่าง 2 hops, TTL จะเหลือ 253

3. ทำไมป้องกันการโจมตีได้:
- แฮกเกอร์ที่อยู่ไกลออกไปในอินเทอร์เน็ต:
  * ไม่สามารถส่ง packet ที่มี TTL=255 มาถึง router ได้
  * เพราะระหว่างทางต้องผ่าน router หลายตัว TTL จะลดลงเรื่อยๆ
  * พอมาถึง router เป้าหมาย TTL จะต่ำกว่า 253
  * packet จะถูก drop ทันที

4. ข้อดี:
- ป้องกันการโจมตีจากระยะไกล
- packet ที่ไม่ถูกต้องถูก drop ที่ hardware level (ไม่ต้องใช้ CPU)
- แฮกเกอร์ต้องอยู่ใกล้ๆ router ถึงจะโจมตีได้ (ซึ่งยากในความเป็นจริง)

ง่ายๆ คือ ใช้ระยะทาง (วัดจาก TTL) เป็นตัวกรองว่า packet นั้นมาจาก peer ตัวจริงหรือเปล่า เหมือนเราบอกว่า "ถ้าไม่ได้อยู่ใกล้ๆ กัน ไม่ต้องมายุ่งกับ BGP ของฉัน"

จากโจทย์ Task 5 เป็นการแก้ปัญหา BGP peering โดยใช้ PPPoE แทนวิธีการเดิมๆ สรุปขั้นตอนได้ดังนี้:

1. ปัญหาที่พบในการ BGP peer ผ่าน loopback:
- Route ไป loopback เป็น OSPF route (O) ไม่ใช่ Connected route (C)
- ทำให้ BGP connected check ไม่ผ่าน
- วิธีแก้แบบเดิมคือใช้ disable-connected-check หรือ ebgp-multihop หรือ ttl-security

2. วิธีแก้แบบใหม่โดยใช้ PPPoE:
- ลบ BGP config เดิมออกก่อน

- ตั้งค่า R1 เป็น PPPoE Server:
  * สร้าง virtual-template ที่ใช้ IP จาก loopback
  * สร้าง bba-group แล้วผูกกับ virtual-template
  * ใส่ PPPoE enable บน interface
- ตั้งค่า R2 เป็น PPPoE Client:
  * สร้าง dialer interface ที่ใช้ IP จาก loopback
  * ตั้งค่า PPPoE client บน physical interface

3. ผลลัพธ์:
- PPPoE สร้าง direct connection ระหว่าง R1-R2
- BGP สามารถ peer ได้โดยไม่ต้องใช้คำสั่งพิเศษ
- มี warning เรื่อง OSPF MTU mismatch ระหว่าง virtual-access (1492) กับ dialer (1500)
- แก้โดยใช้ ip ospf mtu-ignore

วิธีนี้ทำให้ BGP peer ผ่าน loopback ได้โดยไม่ต้องแก้ไข BGP behavior ปกติ เพราะ PPPoE ทำให้เกิด direct connection

PPPoE (Point-to-Point Protocol over Ethernet) คือโปรโตคอลที่ใช้สำหรับการเชื่อมต่อแบบ point-to-point ผ่านเครือข่าย Ethernet มีลักษณะสำคัญดังนี้:

1. องค์ประกอบหลัก:
- PPPoE Server: ทำหน้าที่ให้บริการและจัดการการเชื่อมต่อ
- PPPoE Client: อุปกรณ์ที่ต้องการเชื่อมต่อกับ server
- PPPoE Session: การเชื่อมต่อระหว่าง client และ server

2. การทำงาน:
- Discovery Stage: Client ค้นหา server ที่พร้อมให้บริการ
- Session Stage: สร้างการเชื่อมต่อ point-to-point ระหว่าง client กับ server

3. การใช้งานทั่วไป:
- บริการ ADSL/DSL: ISP ใช้ PPPoE เพื่อ:
  * ควบคุมการเชื่อมต่อของลูกค้า
  * จัดการ IP addressing

* เก็บข้อมูลการใช้งานเพื่อเรียกเก็บเงิน

4. ข้อดี:
- สร้าง direct connection ระหว่างจุดสองจุด
- รองรับการ authentication
- สามารถจัดการ bandwidth และ QoS
- ง่ายต่อการติดตามการใช้งาน

5. ในกรณี BGP:
- ใช้เพื่อสร้าง direct connection ระหว่าง router
- ทำให้ BGP มองว่าเป็น directly connected neighbors
- แก้ปัญหา connected check โดยไม่ต้องใช้คำสั่งพิเศษของ BGP

ง่ายๆ คือ PPPoE ช่วยให้อุปกรณ์สองตัวเชื่อมต่อกันโดยตรงผ่าน Ethernet เหมือนต่อสายตรงถึงกัน แม้จะอยู่คนละเครือข่าย

Lab 6-7
จากภาพอธิบายปัญหาและการแก้ไขได้ดังนี้:

1. ปัญหาที่เกิดขึ้น:
- R1 พยายาม peer กับ R3 ผ่าน R2 (2 hops)
- เมื่อ R1 ส่ง TCP SYN packet ไปหา R3:
  * Packet มี TTL=1 (ค่า default)
  * เมื่อถึง R2, TTL ลดเหลือ 0
  * R2 จึงส่ง ICMP time exceeded กลับมา
- ทำให้ BGP session ไม่สามารถสร้างได้

2. วิธีแก้มี 2 วิธี:
* วิธีที่ 1: ใช้ ebgp-multihop 2
  - ตั้ง TTL เริ่มต้นเป็น 2
  - R2 ลด TTL เหลือ 1
  - R3 รับ packet ได้พอดี

* วิธีที่ 2: ใช้ ttl-security hops 2

- ตั้ง TTL เริ่มต้นเป็น 255
- R2 ลดเหลือ 254
- R3 รับที่ 253

3. เหตุผลที่เลือกใช้ ebgp-multihop แทน ttl-security:
- ebgp-multihop เรียบง่ายกว่า - แค่ตั้ง TTL ให้พอดีกับจำนวน hop
- ttl-security มีความซับซ้อนเพิ่ม:
  * ต้องตั้งค่าทั้งสองฝั่งเหมือนกัน
  * มีการตรวจสอบ minimum TTL
  * เหมาะกับกรณีที่ต้องการความปลอดภัยสูง
- ในกรณีนี้แค่ต้องการให้ packet ไปถึง peer ที่อยู่ห่าง 2 hops
  * ebgp-multihop 2 ก็เพียงพอแล้ว
  * ไม่จำเป็นต้องใช้ security feature ที่ซับซ้อนกว่า

สรุป: เลือก ebgp-multihop เพราะตรงไปตรงมา และเพียงพอสำหรับความต้องการในการทำ multihop BGP peering โดยไม่ต้องการ security feature เพิ่มเติม