การพัฒนาระบบควบคุมเครือข่ายจากภายในสู่ภายนอกสำหรับเครือข่ายแบบหลายจุด
Development of an Egress NAC System for Multi-Hop Networks

อรรถพล สุวรรณษา[1], สมนึก พ่วงพรพิทักษ์[1]
Atthapol Suwannasa[1], Somnuk Puangpronpitag[1]
Received: 31 August 2015; Accepted: 30 November 2015

บทคัดย่อ
หลากหลายอุปกรณ์เครือข่ายต้องยืนยันตัวตนกับระบบควบคุมเครือข่ายจากภายในสู่ภายนอก (Egress NAC) เพื่อรับสิทธิ์ในการเข้าถึงเครือข่าย แต่ยังมีสามปัจจัยหลักที่ต้องคำนึงในการเลือกใช้ระบบเพื่อควบคุมอุปกรณ์เหล่านี้ในเครือข่ายขนาดใหญ่แบบหลายจุด (Multi-Hop) ประกอบไปด้วย ความสามารถในการให้บริการ (Capacity) ความยืดหยุ่นในการใช้งาน (Flexibility) และความถูกต้องของระบบ (Validity) ทั้งนี้หลายระบบที่เป็น Open source ยังมีข้อจำกัดในเรื่องความสามารถในการให้บริการในการบริการผู้ใช้จำนวนมาก นอกจากนี้หลายระบบที่ถูกพัฒนามาแตกต่างกันของแต่ละผู้ผลิตยังมีความไม่ยืดหยุ่นในการใช้งาน และระบบส่วนใหญ่ยังมีความบกพร่องในการเก็บข้อมูล MAC address ของผู้ใช้ตามข้อกำหนดของ พรบ.ว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. 2550 งานวิจัยนี้จึงนำเสนอการพัฒนาระบบ Egress NAC ที่สนับสนุนการทำงานของทั้งสามปัจจัย ผลการพัฒนาแสดงให้เห็นว่าระบบมีประสิทธิภาพดีในการทำงานบนเครือข่ายแบบหลายจุดขนาดใหญ่ที่มีการเชื่อมต่อจำนวนมาก อีกทั้งมีความยืดหยุ่นในการติดตั้งใช้งานโดยที่ไม่ต้องเปลี่ยนแปลงระบบเครือข่ายใหม่ และยังสามารถเก็บข้อมูลผู้ใช้ได้อย่างถูกต้อง

คำสำคัญ: การยืนยันตัวตน ระบบควบคุมการเข้าถึงเครือข่าย เครือข่ายแบบหลายจุด

Abstract
Several different end-user devices have to authenticate to an egress NAC system prior to granting network access. Nevertheless, enforcing the system to control these devices in a large-scale multi-hop network presents a number of challenges in three main factors including capacity, flexibility, and validity. Firstly, many open source systems provide insufficient capacity to serve the large amount of users. Secondly, several proprietary solutions may reduce the flexibility of the enforcement. Lastly, due to the Thailand Computer Crime Act of 2007, the failure of collecting valid user MAC addresses into log files is still the notable problem in most systems. Hence, this paper proposes the development of an egress NAC system to enhance the aforementioned factors. The experimental results have indicated that our system can effectively provide the capacity under the huge number of concurrent connections. The development outcomes have demonstrated that the system is flexible especially in achieving complete enforcement without disrupting the network. Furthermore, the system can also store valid log data of users.

Keywords: Authentication, Egress NAC, Multi-Hop Networks

[1] *อาจารย์, ภาควิชาวิทยาการคอมพิวเตอร์, คณะวิทยาการสารสนเทศ, มหาวิทยาลัยมหาสารคาม อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150*

[1] *Lecturer, Department of Computer Science, Faculty of Informatics, Mahasarakham University, Kantharawichai District, Maha Sarakham 44150, Thailand*

บทนำ

ในเครือข่ายขนาดใหญ่ที่มีการเชื่อมต่อแบบ Multi-Hop ผู้ใช้ (Client) สามารถใช้อุปกรณ์เครือข่ายต่างๆ ร้องขอการใช้งานจากหลากหลายจุด ทำให้เกิดความเสี่ยงต่อความมั่นคงและการจัดการที่ยุ่งยาก จึงมีการติดตั้งระบบควบคุมเครือข่ายจากภายในสู่ภายนอก (Egress Network Access Control) หรือ Egress NAC เพื่อควบคุมผู้ใช้และอุปกรณ์เครือข่าย แต่การเลือกใช้ระบบ Egress NAC สำหรับเครือข่าย Multi-Hop จะต้องคำนึงถึง 3 ปัจจัยหลักประกอบด้วย 1. Capacity คือความสามารถในการให้บริการผู้ใช้จำนวนมาก 2. Flexibility คือความยืดหยุ่นในการติดตั้งและใช้งาน และ 3. Validity คือความถูกต้องตามความต้องการพื้นฐาน[1] ที่ระบบ Egress NAC ต้องมีแต่ระบบในปัจจุบันยังมีปัญหาในทั้ง 3 ปัจจัยดังนี้

1) หลายระบบ Egress NAC แบบ Open source พัฒนาบนพื้นฐานของ Firewall ของระบบปฏิบัติการ Linux (Iptables[2]) ที่ยังมีปัญหาในการตรวจสอบ Packet จำนวนมากในเครือข่าย Multi-Hop ขนาดใหญ่ ทำให้ขาด Capacity ในการให้บริการ

2) ระบบ Egress NAC แบบผลิตภัณฑ์เชิงพาณิชย์จากผู้ผลิตรายใหญ่หลายรายที่เป็น Client/Server based ต้องมีการติดตั้งและตั้งค่าซอฟต์แวร์ฝั่งผู้ใช้ จึงขาด Flexibility ในการใช้งาน และ Open source ส่วนใหญ่ไม่ได้ออกแบบมาเพื่อ Multi-Hop ทำให้ต้องดัดแปลงระบบใหม่หรือเปลี่ยนโครงสร้างของระบบเดิม ซึ่งสร้างความซับซ้อนและอาจมีปัญหาด้านประสิทธิภาพที่ลดลง

3) นอกจากนี้การจัดการผู้ใช้ หรือ Accounting ถือเป็นส่วนสำคัญส่วนหนึ่งของความต้องการพื้นฐาน[1] ของระบบ Egress NAC โดยระบบต้องสามารถตรวจสอบบัญชีผู้ใช้ได้และต้องเก็บบันทึกข้อมูลการใช้งานผู้ใช้ (Log) ให้ถูกต้อง ซึ่งจากประกาศกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร เรื่องหลักเกณฑ์การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ของผู้ให้บริการ พ.ศ. ๒๕๕๐[3] กำหนดให้ต้องมีการเก็บ MAC address ของเครื่องผู้ใช้ แต่ Egress NAC ในเครือข่าย Multi-Hop ส่วนใหญ่ล้วนล้มเหลวต่อ Validity ในการเก็บข้อมูลดังกล่าว เนื่องจาก MAC address จะถูกเปลี่ยนไปในแต่ละ Hop ตามหลักการพื้นฐานของการสื่อสารแบบ Multi-Hop

จากปัญหาในปัจจัยหลักทั้ง 3 ปัจจัย งานวิจัยนี้จึงเสนอการพัฒนาระบบ Egress NAC สำหรับ Multi-Hop networks ด้วย Open source โดยมีการทดสอบ Capacity ถึง 32,766 การเชื่อมต่อ อีกทั้งทำงานแบบ Bridge[4] เพื่อ Flexibility ในการติดตั้ง และประยุกต์ GetMACAddress[5] module เพื่อ Validity ในการเก็บ Log ของผู้ใช้

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

Multi-Hop networks

หากอุปกรณ์เครือข่ายทุกชิ้นเชื่อมต่อเข้าเพียง 1 Local Area Network (LAN) การสื่อสารจะเป็นการส่งข้อมูลแบบ Single-Hop โดยที่ข้อมูลที่ส่งภายใน LAN จะถูกส่งจากเครื่องสู่เครื่องโดยตรง ซึ่ง Gateway Router หรือ Egress NAC Server มักติดตั้งภายใต้เครือข่ายเดียวกับเครื่องลูกข่ายดัง (Figure 1) แต่หากมีการส่งข้อมูลจากต้นทางถึงปลายทางผ่าน LAN มากกว่า 1 LAN ขึ้นไป จะเป็นระบบเครือข่ายแบบ Multi-Hop ซึ่งมักเป็นเครือข่ายขนาดใหญ่ เนื่องจากมีส่วนประกอบเป็นเครือข่ายย่อยๆ หลายเครือข่าย (Multi-LAN) การส่งข้อมูลจะถูกส่งต่อแบบ Hop by Hop ผ่านอุปกรณ์เครือข่ายอย่าง Switch หรือ Router ที่ทำหน้าที่แยกเครือข่ายในแต่ละ Hop ออกจากกัน โดยสุดท้ายต้องส่งผ่าน Core layer ของระบบก่อนออกสู่เครือข่ายภายนอกดัง (Figure 2)

Egress NAC สำหรับ Multi-Hop networks

ระบบ Egress NAC ใช้ควบคุมผู้ใช้ภายในก่อนออกสู่เครือข่ายภายนอก ส่วนใหญ่ผู้ใช้ต้องผ่านการยืนยันตัวตน เช่น การใช้ Username และ Password เป็นต้น จากนั้นระบบจะจดจำค่า IP address หรือ MAC address ของเครื่องผู้ใช้เพื่อให้สิทธิ์ Packet ผู้ใช้ที่ผ่านการยืนยันตัวตนในการเข้าใช้งานเครือข่ายหรือเข้าถึงอินเทอร์เน็ต

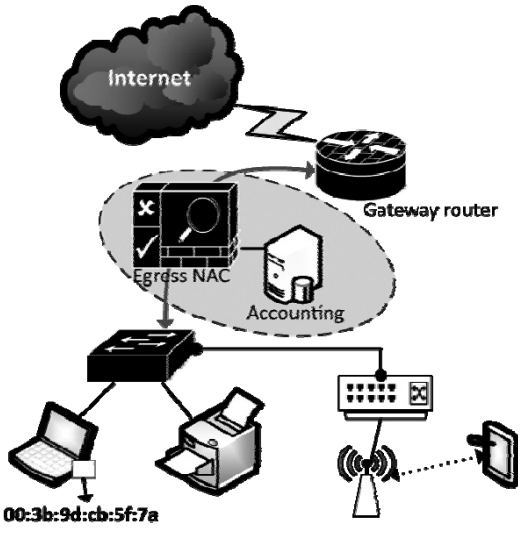

Figure 1 Single-Hop connections

ในการติดตั้งระบบ Egress NAC เพื่อใช้ควบคุมเครือข่ายขนาดใหญ่ที่มีการเชื่อมต่อแบบ Multi-Hop จำเป็นต้องคำนึงถึง 3 ปัจจัยหลัก ดังนี้

1) Capacity คือความสามารถในการรองรับการเชื่อมต่อจำนวนมาก เนื่องจากจำนวนอุปกรณ์เครือข่ายและผู้

ใช้จำนวนมากทำให้เกิดปริมาณ Packet มหาศาล ที่ Egress NAC server จะต้องมีประสิทธิภาพที่ดีในการตรวจสอบและจัดการสิทธิ์ของ Packet เหล่านั้น

2) Flexibility หรือความยืดหยุ่นในการใช้งาน ประกอบไปด้วย 1. สามารถนำ Egress NAC server ไปติดตั้งเข้ากับ Core layer ของเครือข่ายโดยที่ต้องไม่เปลี่ยนแปลงโครงสร้างของระบบเดิม 2. ลดการตั้งค่าที่ซับซ้อนแก่ผู้ใช้ และ 3. ต้องสนับสนุนอุปกรณ์เครือข่ายและระบบปฏิบัติการที่หลากหลาย

3) Validity คือต้องถูกต้องตามความต้องการพื้นฐานของ Egress NAC ประกอบด้วย Authentication, Authorization, และ Accounting หรือ AAA[1]

ปัญหาด้าน Capacity ของระบบ Egress NAC

ส่วนใหญ่มักพบในระบบ Egress NAC ที่เป็น Open source หลายระบบ เช่น Coovachilli[6], ClearOS captive portal[7], PepperSpot[8] และ NoCatAuth[9] เป็นต้น ที่อาศัย Firewall ของระบบปฏิบัติการ Linux อย่าง Iptables[2] เป็นตัวช่วยตรวจสอบสิทธิ์ของ Packet ผู้ใช้ ซึ่งหากต้องตรวจสอบ Packet ผู้ใช้จำนวนมาก ประสิทธิภาพจะลดลงจนอาจไม่สามารถให้บริการต่อได้ (ดังจะกล่าวในหัวข้อ “ทดสอบประสิทธิภาพด้าน Capacity”)

ปัญหา Flexibility ในการใช้งาน Egress NAC

การใช้มาตรฐาน IEEE 802.1x[10] ที่ผู้ใช้ต้องยืนยันตัวตนก่อนถึงจะได้รับหมายเลข IP address เพื่อใช้สื่อสาร ซึ่งหากเป็นอุปกรณ์ เช่น Printers Scanners และ IP phones เป็นต้น ที่ไม่สามารถยืนยันตันตนด้วยตัวอุปกรณ์เองได้ นั่นคือจะไม่มี IP address ใช้สื่อสาร จำเป็นต้องตั้งค่าเพื่ออนุญาตให้อุปกรณ์เหล่านี้สื่อสารได้ เพิ่มความยุ่งยากในการจัดการอุปกรณ์ลักษณะดังกล่าว

Cisco NAC[11], Microsoft NAP[12], และ TCG TNC[13] เป็นระบบ NAC แบบ Client/Server[14, 15]15</style></Display Text><record><rec-number>101</rec-number><foreign-keys><key app=”EN” db-id=”a0p9ddfv0xzztwef0f3xfpt3e00prttdrfrr”>101</key></foreign-keys><ref-type name=”Conference Proceedings”>10</ref-type><contributors><authors><author>G.J. Serrao</author></authors></contributors><titles><title>Network access control (NAC ที่พัฒนาโดยผู้ผลิตอุปกรณ์เครือข่ายรายใหญ่ ซึ่งต้องมีการจัดการ Agent ที่ฝั่งผู้ใช้ทำให้เกิดความยุ่งยากและการเปลี่ยนอุปกรณ์ต้องคำนึงถึงความเข้ากันได้ของแต่ละผู้ผลิต ส่วน Open source NAC มักถูกออกแบบมาให้ใช้งานในเครือข่ายแบบ Single-Hop เช่น Coovachilli[6], Rahunas[16], Zero-shell[17], และ ISAN-SNAC[18, 19] เป็นต้น หากจะนำไปใช้บนเครือข่ายแบบ Multi-Hop จะต้องปรับเปลี่ยนการทำงานภายในใหม่ ซึ่งไม่สะดวกต่อผู้ใช้ที่ไม่มีความสามารถในการดัดแปลงระบบ อีกทั้งส่วนใหญ่มีระบบจ่าย IP address ในตัว ทำให้ต้องเปลี่ยนโครงสร้างเครือข่าย สร้างความยุ่งยากต่อผู้ดูแลระบบในการจัดการระบบ IP address ใหม่

ปัญหา Validity ของระบบ Egress NAC

ระบบจะต้องทำงานถูกต้องตามความต้องการพื้นฐาน 3 อย่าง คือ Authentication, Authorization, และ Accounting หรือ AAA ซึ่งทุกระบบ Egress NAC ล้วนแต่สามารถ Authentication และ Authorization ผู้ใช้ได้ แต่ส่วนใหญ่ยังมีปัญหาส่วน Accounting ที่ยังเก็บบันทึกข้อมูลการเข้าใช้งานผู้ใช้ (Log) ไม่ถูกต้อง

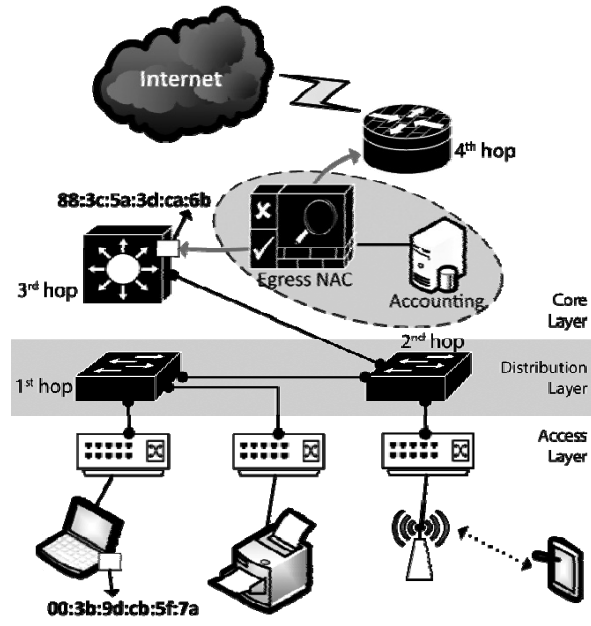

Figure 2 Multi-Hop connections

ในการสื่อสารแบบ Multi-Hop หมายเลข MAC address จะมีความผิดพลาดเนื่องจากถูกเปลี่ยนแปลงในแต่ละ Hop (เห็นได้จาก MAC address ที่ติดต่อกับ Egress NAC ไม่ใช่ของ Client แต่เป็นของ Core Switch ซึ่งเป็น Hop ล่าสุดที่เชื่อมต่อกับ Egress NAC ดัง (Figure 2) จากการสำรวจระบบ Egress NAC ที่ทำงานบน Multi-Hop networks ที่เป็นผลิตภัณฑ์เชิงพาณิชย์ เช่น Sangfor IAM[20] (กำลังใช้งานในมหาวิทยาลัยมหาสารคาม) Forescout[21] และ ระบบอื่นๆ ที่ไม่สามารถเปิดเผยข้อมูลได้ พบว่าเก็บข้อมูล MAC address ผู้ใช้ผิดพลาด หลายระบบเลือกที่จะไม่เก็บ MAC Address ทำให้ไม่สามารถนำ Log ไปใช้เป็นหลักฐานตามข้อกำหนดของ

หลักเกณฑ์การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ของผู้ให้บริการ พ.ศ. ๒๕๕๐[3] ได้ แม้ว่าระบบ Egress NAC ที่ใช้ Module getMACAddress[5] สามารถเก็บ MAC address ในเครือข่าย Multi-Hop ได้ถูกต้อง แต่ยังไม่มีการทดสอบ Capacity ในระบบเครือข่ายขนาดใหญ่ ซึ่งถือเป็นหนึ่งในปัจจัยสำคัญของระบบ Egress NAC สำหรับ Multi-Hop networks

วัตถุประสงค์ของการพัฒนาระบบ

ในการพัฒนาระบบได้คำนึงถึง 3 ปัจจัยหลักที่กล่าวในข้างต้น โดยมีวัตถุประสงค์ดังต่อไปนี้

1) เพื่อสนับสนุน Capacity ของเครือข่ายขนาดใหญ่ ซึ่งระบบได้เลือกใช้ Ipset[22] เป็นตัวกรอง Packet ผู้ใช้เพื่อลดข้อจำกัดของระบบ Egress NAC หลายระบบที่ทำงานโดยอาศัย Iptables[2] เป็นตัวกรอง Packet ที่มีการตรวจสอบกฎแบบ Linear order จากบนสู่ล่าง ทำให้อาจมีปัญหาต่อการตรวจสอบ Packet จำนวนมากภายใต้เครือข่ายขนาดใหญ่ (กรณี Worst case คือ Packet ที่ตรวจสอบไม่ตรงกับกฎใดเลย ทำให้เสียเวลาในการเปรียบเทียบกับกฎที่มีอยู่)

2) เพื่อให้ระบบมี Flexibility ในการนำไปใช้งานในเครือข่ายขนาดใหญ่ที่ต้องติดตั้ง Egress NAC server ที่ Core layer ของระบบ จึงพัฒนาระบบด้ายการเชื่อมต่อแบบ Bridge[4] โดยเสมือนว่า 3rd hop ใน (Figure 2) เชื่อมกับ 4th hop โดยตรง ที่ Egress NAC server ต้องทำงานแบบ Transparent นอกจากนี้จะต้องไม่มีการตั้งค่าใดๆ ในฝั่งผู้ใช้ และสามารถใช้กับระบบ IP address เดิมของเครือข่ายได้ทันที

3) เพื่อความ Validity ของการจัดการผู้ใช้ (Accounting) ในงานวิจัยนี้ได้ประยุกต์ Module getMACAddress[5] เพื่อการเก็บข้อมูล Log ที่ถูกต้อง

การพัฒนาระบบ

งานวิจัยนี้ได้พัฒนาระบบแบ่งออกเป็น 4 ส่วนหลักๆ ดังนี้

ส่วนที่ 1 Core service ใช้ควบคุมสิทธิ์ผู้ใช้ (Authorization) ซึ่งการตรวจสอบสิทธิ์ของ Packet ผู้ใช้ว่าผ่านการยืนยันตัวตนแล้วหรือไม่ ใช้การตรวจสอบกฎ Firewall ของ Ipset 6.2[22] ที่มีประสิทธิภาพในการ Matching packet เนื่องด้วยใช้วิธีการ Hash จึงมีความเร็วในการ Matching โดยสร้าง Whitelist เพื่อเก็บค่า IP address ผู้ใช้ที่ผ่านการยืนยันตัวตนเพื่อให้สิทธิ์ในการเข้าถึงเครือข่าย ซึ่งการเพิ่มหรือลบ IP address ผู้ใช้ใน Whitelist ด้วย Ipset ที่ Core Service จะทำการ Execute คำสั่งต่อไปนี้

```
# Add IP address to Whitelist
/usr/sbin/ipset add whitelist IP_client
# Delete IP address from Whitelist
/usr/sbin/ipset del whitelist IP_client
```

ส่วนที่ 2 Accounting ใช้เชื่อมต่อกับ Egress NAC ดัง (Figure 2) ทำเป็น Database server (ติดตั้งเป็น MySQL version 14.14) เพื่อใช้ในการเก็บและตรวจสอบข้อมูลผู้ใช้ที่ส่งมาจาก Core service และติดตั้งเป็น Freeradius-2 เพื่อใช้ตรวจสอบผู้ใช้และเก็บ Log

ส่วนที่ 3 User interface ติดตั้ง Apache 2.0 web server บน Egress NAC และพัฒนา login.php ด้วย PHP เพื่อตรวจสอบการยืนยันตัวตน (Authentication) กับ Server ที่จัดเก็บข้อมูลผู้ใช้ (Accounting) โดยผู้ใช้ต้องป้อน Username และ Password เพื่อยืนยันตัวตน อีกทั้งควบคุม Session ผู้ใช้ (ประกอบไปด้วย Login, Logout, Session timed out) ในส่วนการติดต่อระหว่าง Core service และ PHP ใช้ XML-RPC[23] เป็นตัวเชื่อม

ส่วนที่ 4 Get MAC address ใช้เก็บค่า MAC address ที่ถูกต้องจาก Client โดยใช้ GetMACAddress[5] ซึ่งเครื่อง Client ต้องติดตั้ง Agent เพื่อดึงค่า MAC address แต่ตัว Agent จะแฝงอยู่ใน login.php และจะติดตั้งโดยอัตโนมัติเมื่อผู้ใช้ผ่านการยืนยันตัวตน ซึ่งลดปัญหาความยุ่งยากในการจัดการ Agent

Egress NAC พัฒนาบนระบบปฏิบัติการ CentOS 6.7 kernel 2.6 และทำงานแบบ Bridge ทำให้ไม่ต้องเปลี่ยนโครงสร้างของเครือข่าย สามารถติดตั้งที่ Core layer ของระบบได้ทันที เช่น ติดตั้งระหว่าง Core switch และ Gateway router ดัง (Figure 2) โดยที่ไม่ต้องเปลี่ยนแปลงรูปแบบการเชื่อมต่อใหม่

Algorithm 1.1 Egress NAC Operation – Authentication

```
1:  procedure AUTHENTICATION (Usr_client, Pwd_client, IP_client)
2:    if Usr_client = Usr_radius && Pwd_client = Pwd_radius then
3:     mac = getMACAddress()
4:     log ← Usr_client, IP_client, mac
5:     login = ture
6:     go to procedure CORESERVICE (IP_client, login)
7:    end if
8:  end procedure
```

การทำงานของระบบ

ผู้ใช้จะต้องทำการยืนยันตัวตนก่อน โดยระบบจะทำการเปลี่ยนปลายทางของผู้ใช้ไปที่ login.php (ส่วนที่ 3) เพราะผู้ใช้ไม่ได้อยู่ใน Whitelist จากนั้นเริ่มป้อนข้อมูล Username (Usr_{client}) และ Password (Pwd_{client}) ส่วน IP address เครื่องผู้ใช้ (IP_{client}) จะถูกส่งมายัง Egress NAC server ด้วย login.php โดยอัตโนมัติ จากนั้นก็จะเข้าสู่ Procedure Authentication ใน Algorithm 1.1 ซึ่ง Server จะนำข้อมูล Usr_{client} และ Pwd_{client} ไปตรวจสอบกับ Accounting server หากตรงกับข้อมูลที่มีในระบบแสดงว่าผู้ใช้มีตัวตนอยู่จริง ที่หน้า login.php ก็จะทำการเรียก getMACAddress (ส่วนที่ 4) ที่พัฒนาด้วย JAVA Applet (ติดตั้งโดยอัตโนมัติผ่าน Web browser) เพื่อดึงค่า MAC address ที่แท้จริงของเครื่อง Client จากนั้นนำข้อมูลผู้ใช้ไปเก็บไว้ใน Log files ที่ Accounting server

เมื่อยืนยันตัวตนเสร็จ ข้อมูล IP_{client} และสถานะการ Login ของผู้ใช้ (ซึ่งค่าจะเป็น True เมื่อผู้ใช้ทำการ Login และค่าจะเป็น False เมื่อผู้ใช้ทำการ Logout หรือ Session timed out) จะถูกส่งไปที่ Procedure Core service ตาม Algorithm 1.2 โดยระบบจะสร้าง Whitelist ด้วย Ipset โดยเก็บค่า IP address เพื่อใช้กรอง Packet ผู้ใช้ของผู้ใช้ที่ผ่านการยืนยันตัวตน และ IP address จะถูกลบออกจาก Whitelist เมื่อผู้ใช้ Logout หรือ Session timed out

---

Algorithm 1.2 Egress NAC Operation – Core service

```
1:  procedure CORESERVICE (IP_client, login)
2:   start = ture
3:   while start do
4:    if IP_client in whitelist && login = false then
5:     whitelist → IP_client
6:    else if login = true then
7:     whitelist ← IP_client
8:    end if
9:    If start = false then  /* Close service */
10:    go to 3
11:   end if
12:  end while
13: end procedure
```

ซึ่งถ้า IP_{client} อยู่ใน Whitelist ระบบจะยอมปล่อย Packet ผู้ใช้ผ่าน แต่หากไม่อยู่และผู้ใช้ผ่านการยืนยันตัวตน (login=true) ค่า IP address ผู้ใช้ก็จะถูกนำไปเก็บใน Whitelist แต่หาก IP ใดไม่อยู่ใน Whitelist ซึ่งเกิดจากผู้ใช้ไม่ผ่านการยืนยันตัวตน ผู้ใช้ Logout หรือ Session timed out ระบบจะ Redirect ไปยังหน้า login.php เพื่อให้ผู้ใช้ยืนยันตัวตน หากผู้ดูแลระบบทำการปิดการทำงานของ Core Service (start=false) ตัว Service ก็จะไม่สามารถให้บริการในการตรวจสอบ Whitelist ได้ นั่นคือปิดการทำงานของระบบ Egress NAC

การเชื่อมต่อเครือข่าย

เพื่อทดสอบระบบที่พัฒนา ได้เชื่อมต่อแบบระบบปิดเพื่อควบคุมปัจจัยอื่นๆ ที่อาจมีผลต่อการวัดประสิทธิภาพของระบบ ดัง (Figure 2) โดยจำลองการเชื่อมต่อแบบเครือข่าย Multi-Hop ที่ใช้อยู่จริงทั่วไป และใช้อุปกรณ์เครือข่ายจริง ซึ่งออกแบบการเชื่อมต่อทั้งหมด 4 hops แต่ละ hop มีรายละเอียดอุปกรณ์ที่เชื่อมต่อแตกต่างกัน ดังนี้

1st hop – เป็นส่วน Distribution layer ใช้ Switch Cisco 2960 เชื่อมต่อกับ Switch ใน Access Layer (3Com baseline switch 2824) เพื่อกระจายการเชื่อมต่อของ Endpoint อย่าง PC หรือ Laptop เป็นต้น โดยมี PC เป็นเครื่องผู้ใช้ที่นำมาทดสอบซึ่งมีคุณสมบัติ คือ Intel Core 2 Duo 2.66GHz 2GB of RAM 500GB Hard disk และเชื่อมต่อด้วย Intel PRO/1000 Network Interface Card

2nd hop – Switch (Cisco 2960) เชื่อมต่อกับ Access Point เพื่อกระจายการเชื่อมต่อไปยังอุปกรณ์ไร้สายเช่น Tablets หรือ Smart phones เป็นต้น

3rd hop – เป็น Core Switch (Cisco 2960) ติดตั้งที่ Core layer ของระบบ เพื่อเชื่อม Gateway router และ Switch ที่อยู่ใน Distribution layer

4th hop – เป็นส่วนของ Gateway router ที่เชื่อมต่ออินเทอร์เน็ต โดยติดตั้งเป็น Server ที่มีคุณสมบัติ คือ Intel Core i5 3.30GHz 4GB of RAM 1TB Hard disk และเชื่อมต่อด้วย 10BaseT/100BaseT/1000BaseT Network Interface Card ระบบปฏิบัติการที่ใช้ คือ CentOS 6.7 และติดตั้ง Apache 2.0 เป็น Web server

เครื่อง Egress NAC มีคุณสมบัติ คือ Intel Xeon 2 processors 2.0 GHz 4GB of RAM Hard disk 1TB และ มี 2 Network Interface Cards ที่มีคุณสมบัติคือ 10BASE-T/100BASE-TX/1000BASE-T โดย Card ที่ 1 เชื่อมต่อกับ Core switch ส่วน Card ที่ 2 เชื่อมต่อกับ Gateway router และติดตั้งระบบปฏิบัติการ CentOS 6.7 ร่วมกับ Iptables-1.4.7 Ipset-6.11 Apache-2.0 PHP5 และ MySQL-14.14 เพื่อ Run ซอฟต์แวร์ Egress NAC ที่พัฒนา อีกทั้งเชื่อมต่อกับ Accounting server เพื่อตรวจสอบและเก็บ Log ข้อมูลผู้ใช้

Egress NAC ถูกพัฒนาให้ทำงานแบบ Bridge สามารถติดตั้งที่ Core layer โดยไม่ต้องแก้ไขโครงสร้างของเครือข่ายที่มีอยู่ก่อนหน้า ดังจะเห็นได้จากตำแน่งที่ติดตั้งระหว่าง 3rd hop กับ 4th hop ใน (Figure 2) เพียงแต่ต้องตั้งค่า IP address ของ Egress NAC ให้สอดคล้องกับเครือข่ายนั่นคือต้องมี IP ให้ Egress NAC ในการ Redirect packet ผู้ใช้ที่ยังไม่ผ่านการยืนยันตัวตนไปยัง login.php ที่ Egress NAC server

ในการเชื่อมต่อแบบ Bridge จะต้องเชื่อม Egress NAC server เพื่อขวางทาง Packet ก่อนที่จะออกสู่อินเทอร์เน็ต เช่น ขวางระหว่าง Core switch และ Gateway router (ดัง Figure 2) แต่ระบบจะต้องมีการตั้งค่าให้สนับสนุนการทำงานแบบ Bridge ด้วย โดยที่ Egress NAC server จะต้องแก้ไข Configuration files บนระบบปฏิบัติการ CentOS ที่ /etc/sysconfig/network-scripts/ ประกอบไปด้วย 3 files (Ifcfg-eth0, Ifcfg-eth1, และ Ifcfg-br0) ดังต่อไปนี้

Ifcfg-eth0	**Ifcfg-eth1**
DEVICE=eth0	DEVICE=eth1
TYPE=ETHER	TYPE=ETHER
BRIDGE=br0	BRIDGE=br0

จากการเชื่อมต่อใน (Figure 5) ไฟล์ ifcfg-eth0 เป็นไฟล์ที่ตั้งค่าให้กับ Network Interface Card ที่ชื่อ eth0 (หมายเลข 1) และไฟล์ ifcfg-eth1 ตั้งค่าให้กับ eth1 (หมายเลข 4) จะเห็นว่ามีชนิดเป็น Ether (Ethernet) ทั้งสอง Cards ที่สำคัญ Bridge=br0 คือการเชื่อมทั้งสอง Cards เข้าหากันด้วย Bridge ที่ชื่อ br0 ซึ่งมีการตั้งค่าดังในไฟล์ ifcfg-br0 ดังนี้

Ifcfg-br0
DEVICE=br0
TYPE=Bridge
IPADDR=10.99.90.2
GATEWAY=10.99.90.1
NETMASK=255.255.255.248

จากไฟล์ ifcfg-br0 เป็นการเชื่อม eth0 กับ eth1 โดยมีการตั้งค่า IP address ให้อยู่ใน LAN เดียวกับ Core layer เพื่อให้ระบบสามารถ Redirect packet ของผู้ใช้มายังหน้า login.php ที่ Egress NAC server ได้ และตั้งค่า Gateway และ Subnet mask ไปยัง Gateway router เพื่อเชื่อมต่ออินเทอร์เน็ตต่อไป

ทดสอบประสิทธิภาพด้าน Capacity

งานวิจัยนี้ได้ทำการทดสอบประสิทธิภาพ ดังนี้

1) เนื่องด้วยหลายระบบ Egress NAC แบบ Open source ตรวจสอบ Packet ผู้ใช้ด้วย Iptables งานวิจัยนี้จึงได้ทดสอบ Iptables-based egress NAC ด้วยจำนวนการเชื่อมต่อที่ 254 อัตราร้องขอการเชื่อมต่ออยู่ที่ 1000 req/s (requests per second) และเพิ่มกฎ Iptables จาก 0 ถึง 30,000 กฎ เพื่อทดสอบว่าระหว่างระบบที่มีกฎน้อยๆ กับกฎจำนวนมาก จะมีผลต่อการตรวจสอบ Packet ที่ผ่านเข้าออกระบบมากน้อยเพียงใด โดยทำการทดสอบ 30 ครั้ง ในแต่ละการเพิ่มกฎ และเชื่อมต่อแบบ Multi-Hop ดัง (Figure 2)

2) ทดสอบระบบ Egress NAC ที่พัฒนา โดยทดลองแบบเดียวกับที่ทดลองในระบบ Egress NAC แบบ Iptables-based เพื่อหาว่าระบบที่พัฒนาจะมีความสามารถในการตรวจสอบ Packet จำนวนมากในเครือข่ายแบบ Multi-Hop มากน้อยเพียงใด

นอกจากนี้เพื่อทดสอบ Capacity ที่อาจต้องเผชิญในเครือข่ายที่มีขนาดใหญ่มาก จึงได้ทดสอบเพิ่ม Request rate ที่ 5,000 20,000 และ 40,000 (req/s) และในแต่ละ Rate ทดสอบจาก 510 ถึง 32,766 connections และเพิ่มกฎ Firewall จาก 0 ถึง 30,000 โดยทดสอบ 30 ครั้ง ในแต่ละการเพิ่มกฎ

เครื่องมือที่ใช้ทดสอบ

การวัดประสิทธิภาพได้ใช้ซอฟต์แวร์ทดสอบประสิทธิภาพ httperf-0.9.0[24] เพื่อทดสอบการรับส่ง Packet ผ่านตัว Egress NAC ว่ามีประสิทธิภาพในการส่งต่อ Packet ของผู้ใช้ที่ผ่านการยืนยันตัวตนแล้ว หรือ การปฏิเสธ (Reject) Packet ของผู้ใช้ที่ยังไม่ผ่านการยืนยันตัวตนมากน้อยเพียงใด โดย httperf จะส่ง Packet ข้อมูลจาก Client ทั้งข้อมูลที่กระทบกฎแล้วยอมรับให้ปล่อยผ่านไปยัง Gateway router ได้ และข้อมูลที่ไม่กระทบกฎใดเลย ซึ่ง Packet จะถูก Drop ทิ้งทันที ซึ่งได้ทำการวัดค่าหลักๆ ดังนี้

- Connection time เป็นค่าเวลา (ms) ที่ใช้ในการส่ง Packet จากเครื่องผู้ใช้ผ่าน hop ต่างๆ และผ่าน Egress NAC ไปยัง Gateway router
- Response time เป็นค่าเวลา (ms) ที่ส่งจาก Gateway router ผ่าน Egress NAC และ hop ต่างๆ กลับมายังเครื่องผู้ใช้

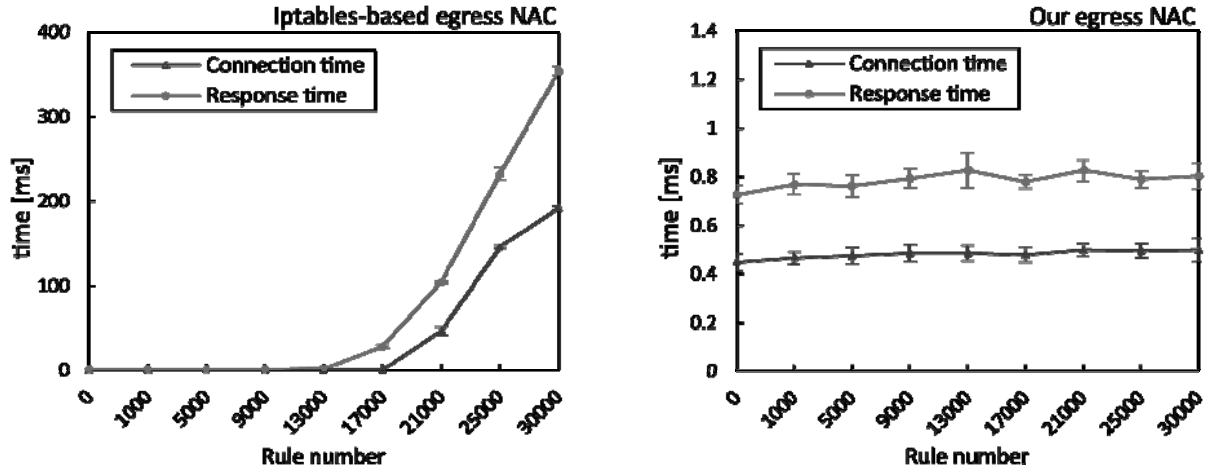

Figure 3 Comparison of the iptables-based egress NAC and our egress NAC

ผลการทดลอง

ผลการวัดประสิทธิภาพระบบ Egress NAC แบบ Open source ที่ทำงานแบบ Iptables-based ผลปรากฏดัง (Figure 3) จากรูปจะเห็นว่าหากจำนวนกฎน้อย (0-13,000) ค่า Connection time และค่า Response time ของ Iptables-based มีค่าน้อยมาก โดยค่าเวลาที่มากที่สุดที่ช่วงความเชื่อมั่น 95% คือ 1.34±1.15 ms และ 2.60±1.16 ms ตามลำดับ นั่นคือระบบยังมีประสิทธิภาพในการตรวจสอบ Packet ของผู้ใช้ได้เป็นอย่างดี แต่เมื่อจำนวนกฎมีมากขึ้น (17,000-30,000) ยิ่งเพิ่มจำนวนกฎค่าเวลาที่ได้ยิ่งเพิ่มขึ้นเป็นอย่างมาก โดยค่า Connection time และค่า Response time ที่มากที่สุดคือ 192.23±1.83 ms และ 354.25±5.49 ms ตามลำดับ ซึ่งส่งผลต่อประสิทธิภาพที่ลดลงเมื่อเทียบกับระบบที่งานวิจัยนี้พัฒนา ซึ่งจะเห็นว่าแม้ว่าจะเพิ่มจำนวนกฎมากขึ้นก็ไม่ส่งผลต่อประสิทธิภาพการตรวจสอบที่ลดลง โดยค่า Connection time และค่า Response time ที่มากที่สุดคือ 0.50±0.046 ms และ 0.82±0.044 ms ตามลำดับ ซึ่งเป็นค่าเวลาที่น้อยมาก

และเนื่องจาก Packet ที่ Egress NAC จะต้องตรวจสอบกับกฎ Firewall ที่มีอยู่ อีกทั้งต้องนำข้อมูลไปประมวลผลจึงจะส่ง Packet ตอบกลับผู้ใช้ได้ ทำให้ค่า Response time มีค่ามากกว่า Connection time อย่างไรก็ตามจะเห็นว่าระบบที่พัฒนามีประสิทธิภาพในการตรวจสอบ Packet ผู้ใช้มากกว่า

นอกจากการทดสอบ Capacity ในข้างต้นแล้ว งานวิจัยนี้ยังทำการวัดประสิทธิภาพของระบบที่ต้องเผชิญกับเครือข่ายที่มีขนาดใหญ่มากในสถานการณ์ต่างๆ ตามที่กล่าวมา ผลที่ได้แสดงดัง (Figure 4)

ซึ่งพบว่า Request rate และ จำนวน Connection ที่มากขึ้นย่อมส่งผลต่อค่าเวลาที่เพิ่มขึ้นเพราะมี Traffic มากขึ้น อย่างไรก็ตามรูปแบบของค่าที่ได้คล้ายกัน โดยที่ 5,000 และ 20,000 Request rate แม้ว่าจะมีการเพิ่มขึ้นและลดลงของ Connection time และ Response time อยู่ตลอด (ทั้งนี้ก็เนื่องจากขึ้นอยู่กับจังหวะความคับคั่ง และความสามารถในการประมวลผลของ Egress NAC ณ เวลานั้นๆ) แต่การเพิ่ม Connection และจำนวนกฎไม่ได้ทำให้ค่าเวลาแตกต่างกันอย่างสิ้นเชิง ค่าที่ได้ยังคงใกล้เคียงกันในแต่ละการเพิ่ม Connection

ที่ 40,000 Request rate ลักษณะของค่าที่ได้ยังคงเป็นเช่นเดิม คือไม่ได้แตกต่างกันอย่างสิ้นเชิงในแต่ละการเพิ่ม Connection แต่จะสังเกตว่าค่า Response time ที่ 510 connections ยังอยู่ในระดับต่ำกว่า 300 ms เนื่องจากจำนวน Connection ยังไม่มาก แม้มี Request rate ที่มากแต่หากระบบยังสามารถประมวลผลจัดการ Packet ได้ ค่าที่ได้ก็จะไม่เพิ่มมากนัก แต่เมื่อจำนวน Connection มากขึ้นผนวกกับ Request rate ที่มาก ค่าก็จะมากขึ้นโดยขึ้นอยู่กับความคับคั่งและการประมวลผล Packet ที่ Egress NAC ณ ช่วงเวลานั้นๆ

อย่างไรก็ตามแม้ว่าจะเพิ่ม Request rate ไปที่ 40,000 req/s จำนวน Connection มากถึง 32,766 และมีกฎ Ipset ถึง 30,000 กฎ ค่า Connection time และค่า Response time ที่มากที่สุดยังอยู่ในระดับที่น้อยและไม่แตกต่างอย่างสิ้นเชิงกับจำนวนกฎที่น้อยกว่า นั่นแสดงว่า ระบบ Egress NAC ที่พัฒนาสามารถให้บริการเครือข่าย Multi-Hop ขนาดใหญ่ได้อย่างดี และหากใช้ Hardware ที่มีประสิทธิภาพมากกว่านี้ก็อาจได้ค่าเวลาที่น้อยลง

ทดสอบ Flexibility ของการใช้งานระบบ

เพื่อความยืดหยุ่นในการติดตั้ง ได้พัฒนาระบบให้สามารถเชื่อมต่อที่ Core layer โดยที่สามารถใช้ระบบ IP address ของเครือข่ายเดิม ซึ่งไม่สามารถทำได้กับระบบ Open source หลายระบบที่กล่าวมา

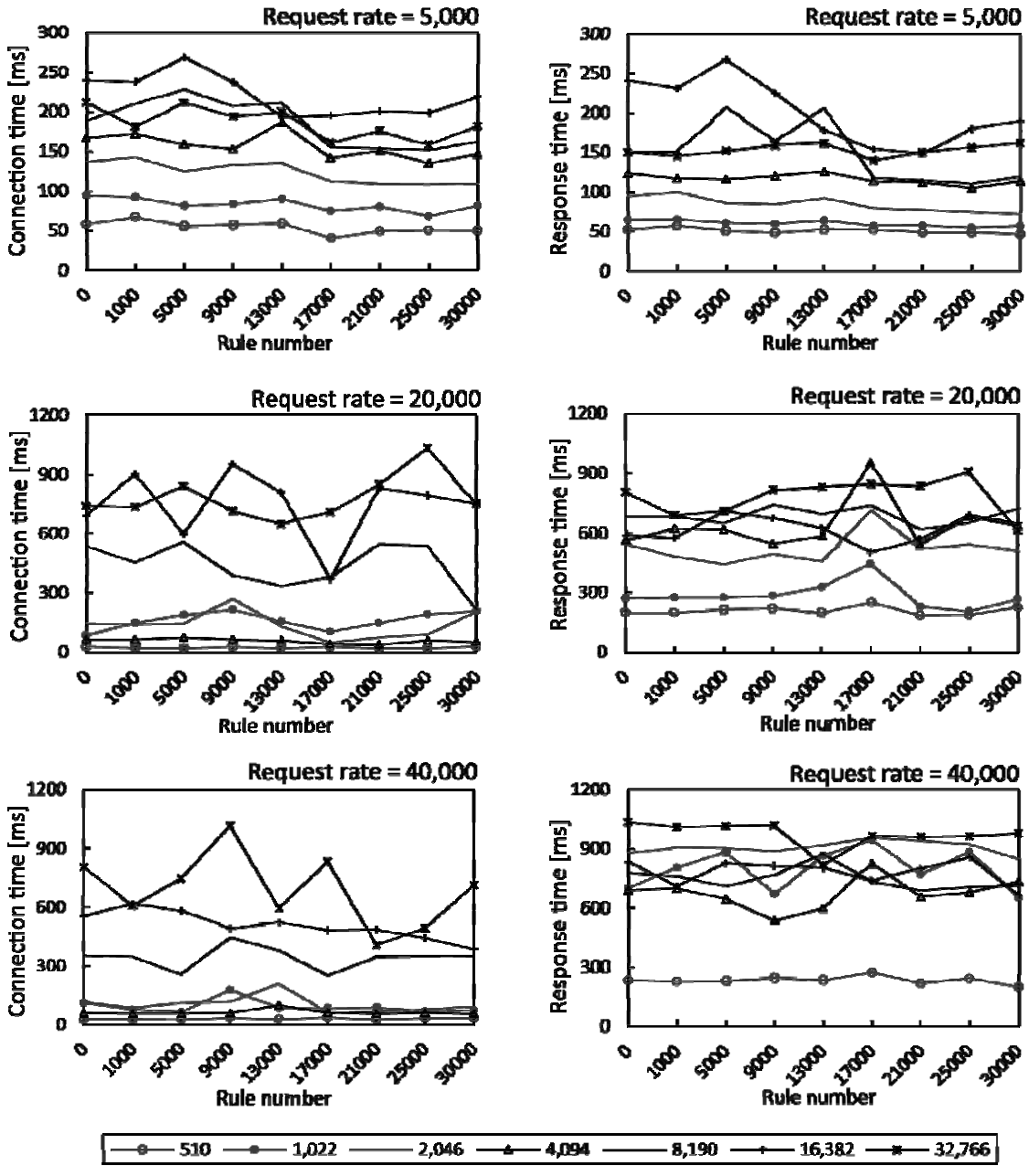

Figure 4 Our egress NAC in different request rates, connections, and rule numbers

จาก (Figure 5) เป็นการเชื่อมต่อแบบ Bridge ที่ Core layer ของระบบ เริ่มจากเชื่อม Egress NAC ที่หมายเลข 1 เข้ากับ Core switch ที่หมายเลข 2 และที่หมายเลข 3 ก็จะเชื่อมไปยัง Switch ที่ Distribution layer ต่อไป ในส่วนการเชื่อมต่ออินเทอร์เน็ต หมายเลข 4 ที่ Egress NAC เชื่อมต่อไปยังหมายเลข 5 ที่ Gateway router และหมายเลข 6 เชื่อมต่ออินเทอร์เน็ตต่อไป จากนั้นตั้งค่า Configuration files ตามที่กล่าวมาแล้วในหัวข้อ “การเชื่อมต่อเครือข่าย”

ระบบเดิมที่ไม่มี Egress NAC จะเชื่อมต่อ Core switch จากหมายเลข 2 ไปยังหมายเลข 5 ที่ Gateway router โดยตรง แต่เมื่อติดตั้งระบบ Egress NAC ก็เพียงนำมาแทรกกลาง จึงไม่ต้องแก้โครงสร้างของระบบเดิม เพิ่ม Flexibility ในการติดตั้งอย่างมาก นอกจากนี้ระบบยังไม่ต้องตั้งค่าใดๆ ฝั่ง Client ทำให้ไม่มีข้อจำกัดในการใช้งานระบบจาก Client ทุกอุปกรณ์และทุกระบบปฏิบัติการ

ทดสอบ Validity ในการเก็บข้อมูลผู้ใช้

เมื่อผู้ใช้ทำการ Login เข้าสู่ระบบผ่านหน้า login.php หากข้อมูลผู้ใช้ถูกต้องก็จะนำข้อมูลผู้ใช้ตามข้อกำหนดของประกาศกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร เรื่อง หลักเกณฑ์การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ของผู้ให้บริการ พ.ศ. 2550 ไปเก็บไว้ใน Log files ที่ Accounting server

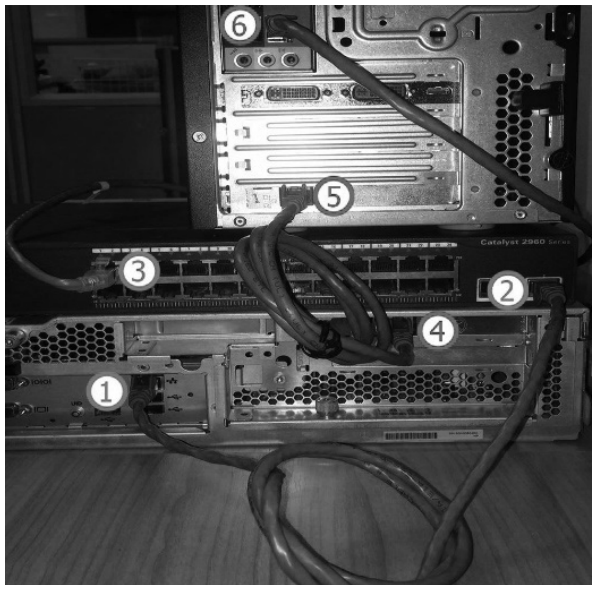

Figure 5 Core layer network set up

```
Tue Nov   10 09:56:14 2015
        Acct-Status-Type = Start
        Acct-Authentic = Local
        User-Name = "alice"
        Framed-IP-Address = 10.31.7.30
        Calling-Station-Id = "00-3B-9D-CB-5F-7A"
        NAS-Identifier = "Egress NAC"
        NAS-IP-Address = 10.99.90.2
```

Figure 6 Authentication log

ในการตรวจสอบ Validity ของการเก็บข้อมูลสำคัญอย่าง MAC address ที่แท้จริงของเครื่องผู้ใช้ใน Log files ผลการเก็บข้อมูลปรากฏดัง (Figure 6) จะเห็นว่าแม้ว่าเครื่อง Client จะอยู่ต่าง Hop กับ Egress NAC server (Figure 2) ระบบยังสามารถเก็บ MAC address (00-3B-9D-CB-5F-7A) ที่เป็นหมายเลขจริงของเครื่องผู้ใช้ที่ใช้ในการทดลองและข้อมูลอื่นๆ ตามข้อกำหนดของหลักเกณฑ์การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ของผู้ให้บริการ พ.ศ. 2550 ได้อย่างถูกต้อง

สรุปผล

เครือข่าย Multi-Hop มักเป็นเครือข่ายขนาดใหญ่ที่ต้องมีการเชื่อมต่ออุปกรณ์เครือข่ายหลายชนิด การเลือกระบบ Egress NAC จึงต้องคำนึงถึง 3 ปัจจัยหลัก คือ Capacity Flexibility และ Validity ซึ่งระบบ Open source ส่วนใหญ่มักมีปัญหาบนเครือข่ายแบบ Multi-Hop ที่ต้องแก้ไขระบบเครือข่ายใหม่ให้สนับสนุนระบบที่เลือกมาติดตั้ง อีกทั้งมี Capacity ที่น้อย หรืออีกหลายระบบที่ต้องตั้งค่า Agent ฝั่ง Client อาจมีปัญหา Flexibility ในการปรับเปลี่ยนอุปกรณ์เครือข่าย และความเข้ากันได้ของซอฟต์แวร์ที่ทำงานแตกต่างระบบปฏิบัติการ และระบบที่ไม่ต้องตั้งค่าฝั่ง Client ก็ยังคงมีปัญหา Validity ในการเก็บข้อมูล Log ของผู้ใช้ที่ผิดพลาด

งานวิจัยนี้จึงพัฒนาระบบ Egress NAC เพื่อสนับสนุนการทำงานทั้ง 3 ปัจจัยหลัก โดยมีทั้งประสิทธิภาพในการให้บริการ ความยืดหยุ่นในการติดตั้งใช้งาน และความถูกต้องในการเก็บข้อมูลผู้ใช้

แต่ระบบยังต้องพัฒนาต่อในส่วนความสามารถในการให้บริการในกรณีที่ระบบมีความขัดข้อง (เนื่องจากติดตั้งที่ Core layer ของเครือข่ายซึ่งเป็นจุดสำคัญที่สุดของระบบ) โดยจะต้องส่งผลน้อยมากต่อการใช้งานตามปกติของเครือข่าย

กิตติกรรมประกาศ

งานวิจัยฉบับนี้ได้รับทุนอุดหนุนการวิจัย ปี 2558 จากหลักสูตรวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม และขอขอบคุณข้อมูลระบบ Egress NAC จากสำนักคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม

เอกสารอ้างอิง

1. Aboba B, Wood J. Authentication, Authorization and Accounting (AAA) Transport Profile. IETF RFC 3539, 2003;
2. Netfilter. The netfilter.org iptables project. Available from http://www.netfilter.org/projects/iptables. Accessed July 10, 2015.
3. ประกาศกระทรวงเทคโนโลยรสารสนเทศและการสื่อสาร เรื่องหลักเกณฑ์การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ของผู้ให้บริการ พ.ศ. 2550. ราชกิจจานุเบกษา
4. Linux Foundation. Bridge. Available from http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge. Accessed October 29, 2015.
5. Suwannasa A, Puangpronpitag S. The Enhancement of Egress NAC Systems for Valid Logging. Journal of Science and Technology Mahasarakham University 2015; 34[3]: 270-276.
6. Coovachilli. Open Source Captive Portal Access Controller and RADIUS Software. Available from http://coova.org/. Accessed November 1, 2015.
7. ClearOS. Captive Portal with Dansguardian. Available from http://www.clearos.com/resources/documentation/clearos/content:en_us:kb_howtos_captive_portal_with_dansguardian. Accessed November 1, 2015.
8. Vincent S, Vancon T. PepperSpot The Next Generation Captive Portal. Available from http://pepperspot.source forge.net. Accessed November 1, 2015.

9. NoCatAuth. Available from http://sourceforge.net/projects/nocatauth. Accessed November 1, 2015.
10. IEEE SA-Standards Board. IEEE Std 802.1X-2004 - Port Based Network Access Control. IEEE; 2004;
11. Cisco. Network Admission Control. Available from http://www.cisco.com/c/en/us/solutions/enterprise-networks/network-admission-control.Accessed October 19, 2015.
12. Microsoft. Networking and Access Technologies. Available from https://technet.microsoft.com/en-us/network/bb545879. Accessed October 29, 2015.
13. TCG. Trusted Network Communications. Available from http://www.trustedcomputinggroup.org/developers/trusted_network_communications. Accessed October 29, 2015.
14. Serrao GJ. Network access control (NAC): An Open Source Analysis of Architectures and Requirements. IEEE International Carnahan Conference on Security Technology (ICCST); 2010; 94-102.
15. Czarny B. Network Access Control Technologies. 2008;
16. Soutmun N, Soutmun S. Rahunas - Durable Captive Portal Solution. Available from http://authen.rahunas.org. Accessed November 1, 2015.
17. Ricciardi F. Hotspot router for authenticated network access. Available from http://www.zeroshell.org. Accessed November 1, 2015.
18. Puangpronpitag S, Suwannasa A. A design of egress NAC using an authentication visa checking mechanism to protect against MAC address spoofing attacks. 8th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON); 2011; 300-303.
19. Puangpronpitag S, Suwannasa A. A Lightweight Agent-based Egress NAC on Wireless LAN. International Conference on Computer & Information Science (ICCIS); Kuala Lumpur, Malaysia. IEEE 2012;
20. SangforIAM. Internet Access Management. Available from http://www.sangfor.com/product/Internet AccessManagement/outline.html. Accessed July 2, 2014.
21. Forescout Technologies. Network Access Control & NAC Solutions. Available from http://www.forescout.com. Accessed November 1, 2015.
22. Kadlecsik J. The netfilter.org ipset project. Available from http://www.netfilter.org/projects/ipset. Accessed June 16, 2015.
23. Kidd E. XML-RPC for C and C++ A lightweight RPC library based on XML and HTTP. Available from http://xmlrpc-c.sourceforge.net. Accessed November 1, 2015.
24. Httperf. A tool for measuring web server performance. Available from http://www.hpl.hp.com/research/linux/httperf. Accessed October 29, 2015.