ระบบรักษาความปลอดภัยบนอินเตอร์เน็ตโดยใช้ไฟร์วอลล์

Internet security Using Firewall

โดย

นางสาวกวิตา ฉายกฤดาธิการ

นายฉัตรชัย ตันติเวชยานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

ระบบรักษาความปลอดภัยบนอินเตอร์เน็ตโดยใช้ไฟร์วอลล์

Internet security Using Firewall

โดย

นางสาวกวิตา ฉายกฤดาธิการ 37014010

นายฉัตรชัย ตันติเวชยานนท์ 37014072

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยธำรง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรักษาความปลอดภัยบนอินเตอร์เน็ตโดยใช้ไฟร์วอลล์

ผู้จัดทำ

1. นางสาวกวิตา ฉายกฤดาธิการ
2. นายฉัตรชัย ตันติเวชยานนท์

.....อาจารย์ที่ปรึกษา

(ผศ. บรรจง ปิยธำรง)

ระบบรักษาความปลอดภัยบนอินเตอร์เน็ตโดยใช้ไฟร์วอลล์

นางสาวกวิตา ฉายกฤดาธิการ
นายฉัตรชัย ตันติเวชยานนท์

ผศ. บรรจง ปิยธำรง
ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของโครงงานวิจัยเพื่อศึกษาระบบรักษาความปลอดภัยของเครือข่าย โดยอาศัยไฟร์วอลล์ (firewall) ทำหน้าที่ป้องกันเครือข่ายภายใน (internal network) ซึ่งเป็นเครือข่ายที่ต้องการป้องกันไม่ให้มีการบุกรุกจากภายนอกเข้ามาทำอันตรายใด ๆ กับระบบ โดยใช้สถาปัตยกรรมแบบดูอัลโฮมโฮสต์ (Dual-home host architecture) ซึ่งมีการทำงานอยู่ในรูปแบบของบริการพรอกซี (proxy service) ทำการควบคุมบริการสำคัญ ๆ ที่มีการส่งต่อระหว่างเครือข่ายภายในและเครือข่ายภายนอก (external network) เช่น เอฟทีพี (FTP) , เทลเน็ต (TELNET)

Internet Security Using Firewall

Miss Kawita Chaikerdatikarn

Mr. Chatchai Tantiwatchayanon

Asst . Prof . Banjong Piyatamrong

1997

Abstract

This thesis deals with the study of network security by using firewall to protect internal network. The internal network protected network from intrusion that wants to harm our system.

Using dual-home host architecture which operated in proxy service. A proxy control the important services that have transmission between internal network and external network such as FTP , TELNET , etc.

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในสังคมทั่ว ๆ ไปมีผู้ที่ใช้อินเตอร์เน็ต (Internet) ประมาณ 30 – 40 ล้านคน ซึ่งมีผู้คนจำนวนน้อยที่ถูกทำอันตราย ถึงแม้ว่าจำนวนของผู้ใช้ที่ทำอันตราย (milicious user) จะไม่ถึง 1% ของคนทั้งหมดในสังคม แต่ก็เป็นจำนวนที่มากเพียงพอที่จำเป็นจะต้องทำการป้องกัน

จำนวนของความปลอดภัยที่ศูนย์ตอบสนองเหตุฉุกเฉินทางคอมพิวเตอร์ (CERT Coordination Center) ที่รายงานออกมามีสถิติที่เพิ่มขึ้นทุก ๆ ปี (ในปี 1989 มีน้อยกว่า 200 , ปี 1991 มี 400 , ปี 1993 มี 1,400 , ปี 1994 มี 2,241 และในปี 1995 มีประมาณ 3,000) ซึ่งส่วนใหญ่จะเกิดขึ้นกับรัฐบาล , กองทัพ , บริษัทหลักทรัพย์ , มหาวิทยาลัย ในบางเหตุการณ์จะเกี่ยวข้องกับแอคเคาท์เดี่ยว (single account) บนระบบเดียว (single system) หรือในบางครั้งจะเกี่ยวข้องกับระบบเป็นแสน ๆ ระบบ ซึ่งตัวเลขนี้เป็นเพียงส่วนเล็กน้อยเท่านั้น การบุกรุก (intrusion) ส่วนใหญ่จะไม่ถูกบันทึกไว้ จริง ๆ แล้วที่การบุกรุกไม่ได้ถูกบันทึกไว้ทั้งหมด เพราะองค์กรที่เป็นเหยื่อของการบุกรุกต้องการที่จะรักษาชื่อเสียงของตนเองไว้ หรือองค์กรอาจไม่สนใจต่อการบุกรุกนั้น ส่วนอีกเหตุผลหนึ่งก็คือ ไม่พบเห็นการบุกรุก

ไม่มีผู้ใดทราบถึงสถิติที่ถูกต้องเกี่ยวกับจำนวนของการโจมตี (attack) ที่ถูกค้นพบโดยองค์กรที่ถูกโจมตี แต่คนส่วนใหญ่ในสังคมที่ปลอดภัย (security community) เห็นพ้องต้องกันว่ามีเปอร์เซนต์ที่เกิดขึ้นน้อยมาก มีบางสถิติจะกล่าวถึงดังต่อไปนี้ กลุ่มที่ตอบสนองการบุกรุกเครือข่าย (network intrusion service) ให้กับลูกค้า ซึ่งกลุ่มนี้พยายามที่จะทะลุผ่านเข้าไปในระบบ โดยใช้วิธีการเดียวกันกับที่ผู้บุกรุก (intruder) ใช้ในการโจมตีของตนเองเมื่อลูกค้าอนุญาต กลุ่มนี้พบไซต์ (site) ที่พยายามจะทะลุผ่านเข้าไปเพียง 4% ซึ่งเป็นตัวเลขที่น่ากลัวมาก บิล เชลวิกสท์ (Bill Cheswict) แห่งเอทีแอนด์ทีเบลล์แล็บ (AT&T Bell Labs) เชื่อว่าการโจมตีที่ประสบผลสำเร็จมีอย่างน้อย 40% ของการโจมตีทั้งหมดที่สามารถเข้าถึงรูท (root) ได้ ซึ่งไม่เฉพาะจำนวนเหตุการณ์จะเพิ่มขึ้นเท่านั้น แต่ยังเป็นการชักนำให้เกิดวิธีการต่าง ๆ ในการโจมตีอีกด้วย เมื่อมีการสร้างอินเตอร์เน็ตเวิร์ม (Internet worm) ขึ้นในปี 1988 การโจมตีจึงได้แบ่งเป็น 2 ประเภท คือ การเดารหัสผ่าน (password) และการแสวงหาประโยชน์จากข้อบกพร่องของระบบรักษาความปลอดภัยในระบบปฏิบัติการ (operating system) และโปรแกรมระบบ (system program) ถึงแม้ว่าไซต์ (site) จำนวนมากจะตกเป็นเหยื่อของการโจมตี แต่เราจะพิจารณาถึงเทคนิคซับซ้อนที่เพิ่มขึ้นในเหตุการณ์ใหม่ ๆ ที่เกิดขึ้น ในบางระดับอาจเป็นเหตุผลที่ทำให้ความรอบคอบของผู้ใช้และผู้บริหารระบบ (administrator) เพิ่มขึ้น โดยผู้ใช้อาจจะตั้งรหัสผ่านของตนเองให้ดีขึ้นยากต่อการเดา และผู้บริหารระบบจะแก้ไขระบบได้เร็วขึ้น อย่างไรก็ตามความรอบคอบของระบบรักษาความปลอดภัยที่เพิ่มขึ้นมานี้ก็ไม่ได้ทำให้การโจมตีระบบรักษาความปลอดภัยลดลง แต่กลับทำให้ผู้โจมตีต้องพยายามค้นหาเทคนิคในการโจมตีใหม่ ๆ ขึ้นมา การโจมตีในปัจจุบันจะใช้การปลอมแปลงเลขไอพี (IP address) คือ ผู้บุกรุกจะเดาลำดับของตัวเลขซึ่งเกี่ยวข้องกับการเชื่อมต่อเครือข่าย และการตอบรับ (acknowledgements) ระหว่างเครื่อง , การหาประโยชน์จากแหล่งของทางเลือกในการจัดหาเส้นทาง

(source routing option) ของไอพีแพ็กเก็ต (IP packet) บนระบบยูนิกซ์ (UNIX) และการปล้นเทอร์มินัล (terminal) ที่เปิด หรือล็อกอินเซสชั่น (login session)

ผู้ใช้และผู้บริหารระบบส่วนมากไม่ได้ศึกษาเกี่ยวกับการโจมตีในรูปแบบเก่า ๆ ซึ่งสิ่งนี้เป็นสิ่งที่สำคัญมาก และมีความจำเป็นที่จะต้องให้พวกผู้จัดการ (manager) ศึกษาด้วยจะได้ตั้งงบประมาณให้มีการฝึกฝนเกี่ยวกับระบบรักษาความปลอดภัย (security training) ซึ่งมีผู้คนตระหนักถึงเรื่องนี้น้อยมาก มีหนทางที่ซึ่งการเจริญเติบโตของอินเตอร์เน็ต (Internet) จะทำอันตรายให้กับเรา อินเตอร์เน็ตในปัจจุบันนี้ ไซต์ (site) ที่เชื่อมต่อกับเครือข่ายมีชุดของซอฟท์แวร์ และฮาร์ดแวร์ ซึ่งในปัจจุบันการที่เชื่อมต่อกับอินเตอร์เน็ตนั้นที่ไซต์นั้น ๆ มักจะลืมที่จะนำเอาเทคนิคที่ทำให้การเชื่อมต่อมีความปลอดภัยรวมเข้าไปด้วย

ถึงแม้ว่าจำนวนของความปลอดภัยที่อาจเกิดขึ้นได้มีเพิ่มขึ้น และประเภทของการโจมตีก็ล้ำสมัยขึ้น จึงเป็นข่าวดีสำหรับผู้ที่สนใจทางด้านความปลอดภัย (security) ทั้งหมดนี้เราได้เห็นถึงการเจริญเติบโตอย่างมากมายในการรับรู้ถึงอันตรายของการเชื่อมต่อกับอินเตอร์เน็ต และมีกิจกรรมจำนวนมากในส่วนที่มีความปลอดภัย สิ่งหนึ่งคือการเจริญเติบโตของเฟิร์สท (FIRST) ซึ่งนำไปสู่การอยู่ร่วมกันของหลาย ๆ กลุ่มที่ตอบสนองต่อความปลอดภัยของคอมพิวเตอร์จากรัฐบาล , วงการค้า และองค์กรอื่น ๆ ดังนั้นความเปิดกว้างจะเป็นการคงอยู่ของเครื่องมือทางด้านความปลอดภัยที่ดีขึ้น ซึ่งแต่ละชุมชนสามารถนำไปใช้ได้สะดวก การจัดการทางด้านคอมพิวเตอร์ (computer operation) , การตรวจสอบ (audit) และเทคโนโลยีทางด้านความปลอดภัย (security technology) เป็นศูนย์กลางสำหรับการรวมกลุ่ม และทดสอบเครื่องมือต่าง ๆ เหล่านี้ ท้ายที่สุดความเป็นสาธารณะของตำรา และเอกสารที่เกี่ยวข้องกับความปลอดภัยบนอินเตอร์เน็ต (Internet security) ทำให้มีการตระหนักถึงระบบรักษาความปลอดภัยมากขึ้น

ในเรื่องที่เกี่ยวกับความอันตรายของระบบคอมพิวเตอร์ ไฟร์วอลล์ (firewalls) เป็นทางเลือกที่ดีที่สุดที่จะทำให้ไซต์ (site) ปลอดภัย ถึงแม้ว่าจะได้รับการรวมระบบรักษาความปลอดภัยประเภทอื่นผสมกันไป แต่ถ้าจะเน้นทางด้านการเชื่อมต่อกับอินเตอร์เน็ต ไฟร์วอลล์ (firewalls) ควรจะเป็นหลักสำคัญที่สุดในการวางแผนที่จะตั้งระบบรักษาความปลอดภัย

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาและทำความเข้าใจการทำงานของระบบเครือข่ายที่ใช้โปรโตคอลทีซีพีไอพี (TCP/IP)
2. ทำความเข้าใจความหมายและความจำเป็นของระบบเครือข่ายที่มีความปลอดภัย
3. ทำความเข้าใจระบบไฟร์วอลล์ และแต่ละรูปแบบการทำงาน
4. ศึกษาข้อบกพร่องของเครือข่ายต้นแบบที่จะนำไฟร์วอลล์มาใช้งานด้วย และกำหนดคุณสมบัติต่าง ๆ อันเป็นเป้าหมายในการดำเนินงานของไฟร์วอลล์ในเครือข่ายนั้น
5. สามารถสร้างระบบไฟร์วอลล์ที่เหมาะสมกับเครือข่ายของภาควิชา ฯ โดยที่บริการเดิมของเครือข่ายยังคงใช้งานได้ดังเดิม แต่มีความปลอดภัยสูงขึ้น

1.3 ขอบเขตของโครงงาน

โครงงานนี้จะทำการศึกษาเครือข่ายของภาควิชา ฯ เพื่อดูว่าในปัจจุบันมีแอพพลิเคชั่นใดใช้งานอยู่บ้าง และมีระดับของความปลอดภัยที่ใช้งานเป็นอย่างไร จากนั้นจะทำการออกแบบระบบไฟร์วอลล์ที่เหมาะสมกับเครือข่ายของภาควิชา ฯ มาป้องกันแอพพลิเคชั่นแต่ละตัว โดยที่แอพพลิเคชั่นแต่ละตัวนั้นต้องทำการบริการได้เหมือนเดิม มีระดับของความปลอดภัยตามที่ต้องการ และยังคงเข้ากับเครือข่ายของภาควิชา ฯ ได้เหมือนเดิม ซึ่งจะมีการกำหนดคุณสมบัติของไฟร์วอลล์ให้เป็นไปตามวัตถุประสงค์ที่กำหนดไว้

1.4 วิธีการดำเนินงาน

ในโครงงานนี้จะเริ่มต้นด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับโครงงาน ซึ่งประกอบไปด้วยเรื่องหลัก ๆ 2 เรื่องดังนี้

1 สิ่งต่าง ๆ ที่เกี่ยวข้องกับการป้องกัน (protect) ของไฟร์วอลล์ (firewall)

2 สถาปัตยกรรมแบบต่าง ๆ ของไฟร์วอลล์ (firewall)

ซึ่งรายละเอียดจะอธิบายในบทที่ 2, 3, 4, 5 และ 6 จากนั้นก็จะศึกษาตัวอย่างของไฟร์วอลล์ที่สามารถหามาได้ ซึ่งไฟร์วอลล์ที่นำมาศึกษามี 3 ตัวด้วยกัน คือ ไฟร์วอลล์วัน (Firewall-1) ซ็อคส์ (Socks) และทีไอเอสไฟร์วอลล์ทูลคิท (TIS FWTK) รวมทั้งมีการศึกษาบริการอื่น ๆ ที่ทำให้ระดับของความปลอดภัยของเครือข่ายเกิดขึ้นสูงสุด และศึกษาเครือข่ายของภาควิชา ฯ รวมทั้งแอพพลิเคชั่นต่าง ๆ ที่สามารถให้บริการได้ เพื่อนำมากำหนดคุณสมบัติของไฟร์วอลล์ และส่วนรายละเอียดของขั้นตอนการดำเนินงานจะอธิบายในบทที่ 7 ส่วนการทดสอบจะอธิบายในบทที่ 8 มีการทำสคริปต์เพื่ออำนวยความสะดวกให้ผู้ดูแลระบบสามารถกำหนดคุณสมบัติของไฟร์วอล์ มีรายละเอียดอยู่ในบทที่ 9

บทที่2

คำแนะนำสำหรับอินเตอร์เน็ตไฟร์วอลล์ (Internet firewall)

ทุก ๆ ครั้งที่องค์กรใดองค์กรหนึ่งทำการเชื่อมต่อระหว่าง ระบบคอมพิวเตอร์ภายในองค์กรนั้นไปยังอินเตอร์เน็ต ก็เท่ากับเป็นการสร้างความเสี่ยงที่จะเกิดความเสียหายขึ้นได้ เพราะความเปิดกว้างของอินเตอร์เน็ตทำให้เครือข่าย (network) ของทุก ๆ องค์กรที่เชื่อมต่อกับอินเตอร์เน็ตมีความเสี่ยงต่อการถูกจู่โจมได้ ซึ่งตามทฤษฎีแล้วแฮกเกอร์ (hacker) บนอินเตอร์เน็ตมีความสามารถที่จะเจาะเข้าไปยังเครือข่ายขององค์กรแล้วทำให้เกิดความเสียหายได้หลาย ๆ วิธีด้วยกัน เช่น ขโมยหรือทำลายข้อมูลที่สำคัญ , ทำลายคอมพิวเตอร์บางเครื่องหรือเครือข่ายทั้งหมดให้ทำงานไม่ได้ , ใช้ทรัพยากรคอมพิวเตอร์ขององค์กรไปในทางที่ไม่เกี่ยวข้องหรือเหมาะสม หรือรวมถึงการใช้เครือข่ายหรือทรัพยากรต่าง ๆ ขององค์กรโดยทำตัวเสมือนหนึ่งเป็นพนักงานของบริษัทนั้น ๆ อันอาจสร้างความเข้าใจผิดให้กับบุคคลภายนอกที่พบเห็นหรือได้รับการติดต่อ

การแก้ไขปัญหานี้ไม่ใช่ว่าจะต้องตัดการติดต่อระหว่างเครือข่ายขององค์กรบนอินเตอร์เน็ต แต่องค์กรสามารถสร้างกลไกที่เรียกว่าไฟร์วอลล์ (firewall) ขึ้นมาสำหรับป้องกันเครือข่ายขององค์กรจากการถูกจู่โจมได้ ไฟร์วอลล์ (firewall) เหล่านี้ยังคงยอมให้พนักงานในองค์กรสามารถออกไปใช้อินเตอร์เน็ต (Internet) แต่จะยับยั้งพวกแฮกเกอร์ (hacker) หรือคนอื่น ๆ บนอินเตอร์เน็ตไม่ให้เข้ามาใช้เครือข่ายขององค์กรแล้วสร้างความเสียหายให้เกิดขึ้นได้

2.1 สิ่งที่ต้องการจะป้องกัน

ในการที่คุณจะสร้างไฟร์วอลล์ (firewall) จะต้องคำนึงถึงสิ่งที่ต้องการจะป้องกัน ซึ่งประกอบไปด้วย 3 สิ่งที่มีอัตราความเสี่ยงสูง ดังนี้

- ข้อมูล (your data) คือ ข่าวสารที่คุณเก็บไว้ในคอมพิวเตอร์
- ทรัพยากร (your resources) คือ คอมพิวเตอร์ของเรา
- ชื่อเสียง (your reputation)

2.1.1 ข้อมูล (your data)

ข้อมูลของคุณที่ต้องการจะป้องกัน (protect) แบ่งออกเป็น 3 ลักษณะใหญ่ ๆ ดังนี้

- มีความเป็นความลับ คือ ไม่ต้องการให้ผู้อื่นรับรู้ข้อมูลที่ต้องการจะป้องกัน
- มีความครบถ้วนสมบูรณ์ คือ ไม่ต้องการให้ผู้อื่นมาเปลี่ยนแปลงข้อมูล
- สามารถนำมาใช้ได้ คือ ถ้าต้องการที่จะใช้ข้อมูลต้องสามารถใช้ได้ตลอดเวลา

ในหลาย ๆ องค์กรมักจะมีข้อมูลที่สำคัญที่เป็นความลับอยู่ในคอมพิวเตอร์ เช่น การออกแบบสำหรับผลิตภัณฑ์ , การซื้อขาย

ถ้าข้อมูลไม่ถูกทำให้เป็นความลับ อาจจะต้องได้รับผลเสียที่เกิดจากการที่ข้อมูลถูกเปลี่ยนแปลงหรือถูกทำลาย สิ่งนี้ถ้าเป็นของที่มีราคา ต้องเสียเงินและเวลาในการจะสร้างขึ้นมาใหม่ หรือถ้าต้องการจะขายข้อมูล อาจจะเสียการขายได้ เช่น เสียข้อมูลที่เป็นโค้ด (code) ของซอฟท์แวร์ (software) ทำให้มีคนมาซื้อน้อยลง และที่สำคัญที่สุดจะทำให้ขาดการได้รับความไว้วางใจ (ความไว้วางใจของผู้ใช้ , ลูกค้า , ทีมงาน , ฯลฯ) ในระบบและข้อมูล และยังทำให้ขาดความไว้วางใจในองค์กรของคุณด้วย

2.1.2 ทรัพยากร (your resources)

ถ้าหากมีคนที่ต้องการจะใช้คอมพิวเตอร์ของคุณ คุณควรที่จะได้ประโยชน์จากการที่ให้ผู้อื่นใช้ คนส่วนมากต้องการที่จะใช้คอมพิวเตอร์ของตัวเอง หรือต้องการที่จะคิดค่าบริการจากผู้อื่นที่ต้องการเข้ามาใช้ คนที่ให้ใช้คอมพิวเตอร์และเนื้อที่เก็บข้อมูลมักจะคาดหวังผลตอบแทนบางอย่าง แต่เราไม่สามารถคาดหวังได้จากผู้บุกรุก คุณต้องเสียเวลาและเงินบนคอมพิวเตอร์ของคุณ และคุณควรที่จะสามารถควบคุมให้ใครมาใช้ได้อย่างไร

2.1.3 ชื่อเสียง (your reputation)

ผู้บุกรุกที่ปรากฏบนอินเตอร์เน็ต (Internet) ของส่วนของคุณ ทุก ๆ สิ่งที่เขาทำก็จะมาจากคุณ อะไรเป็นผลการกระทำ บางครั้งผู้บุกรุกที่ไม่ชอบคุณอาจจะส่งอีเมล์ (E-Mail) หรือข่าวสารที่เป็นสิ่งร้าย ๆ ออกไปโดยอ้างว่ามาจากคุณ ซึ่งมันเป็นการทำลายชื่อเสียงของคุณ

2.2 อะไรบ้างที่พยายามจะต่อต้าน

2.2.1 ชนิดของการโจมตี

มีการโจมตีหลาย ๆ ชนิดที่เกิดขึ้นบนระบบ และมีหลาย ๆ ระดับขั้นของการโจมตี เราสามารถแบ่งแยกระดับขั้นของการโจมตีได้ คือ การบุกรุก (intrusion) , การขัดขวางการให้บริการ (denial of service) , และ การขโมยข้อมูล (information theft)

2.2.1.1 การบุกรุก (intrusion)

การโจมตีโดยทั่วไปบนระบบของคุณ คือ การบุกรุก (intrusion) ซึ่งก็คือ ผู้ที่สามารถเข้ามาใช้คอมพิวเตอร์ของคุณโดยมิได้รับอนุญาต ผู้บุกรุกส่วนมากต้องการใช้คอมพิวเตอร์ของคุณราวกับว่าเขาคือผู้ใช้ที่ถูกต้องสมควร

ผู้บุกรุกมีหลายทางที่จะเข้ามาได้ ตั้งแต่การอ้างสิทธิ์ว่าเป็นเจ้าของกับผู้บริหารระบบ (administrator) ว่าต้องการจะเปลี่ยนรหัสผ่าน (password) หรือโดยการเดารหัสผ่าน (password) หรือใช้วิธีการที่ซับซ้อนในการเข้ามาโดยไม่ต้องใช้ชื่อแอคเคาท์ (account name) และ รหัสผ่าน (password)

2.2.1.2 การขัดขวางการให้บริการ (denial of service)

การโจมตีแบบการขัดขวางการให้บริการ (denial of service) มีจุดมุ่งหมายที่จะกันคุณจากการใช้คอมพิวเตอร์ของคุณ

การโจมตีแบบนี้ เช่น การสร้างโปรแกรม (program) ใหญ่ ๆ ขึ้นมา แล้วส่งมาอย่างรุนแรงผ่านการให้บริการของเครือข่าย ทำให้ข้อความอื่น ๆ ไม่สามารถผ่านเข้ามาได้ ทำให้ต้องปิดระบบลง จึงทำให้การทำงานต่าง ๆ หยุดชะงัก

2.2.1.3 การขโมยข้อมูล (information theft)

เป็นลักษณะของการโจมตีที่ผู้โจมตีจะนำข้อมูลออกมาโดยไม่ต้องเข้าไปที่คอมพิวเตอร์ของคุณโดยตรง โดยทั่วไปจะทำการขูดรีดมาจากการบริการของอินเตอร์เน็ต (Internet) ที่มีจุดประสงค์จะให้ข่าวสารข้อมูล แต่จะทำให้ให้ข่าวสารข้อมูลออกมามากกว่าที่กำหนดไว้ หรือ ออกมาให้ผิดคน

2.2.2 ชนิดของผู้โจมตี

ผู้โจมตีมีหลายลักษณะ พวกเขาไม่ต้องการที่จะถูกจับดังนั้นพวกเขาจึงพยายามที่จะซ่อนตัว ถ้าพวกเขาเข้ามาในระบบของคุณ เขามักจะเก็บข้อมูลไว้และนำข้อมูลที่ได้นี้มาแลกกับคนอื่นที่มีข้อมูลที่เขาสนใจ

2.2.2.1 จอยไรเดอร์ (joyriders)

เป็นคนที่มองหาเครื่องหย่อนใจ พวกเขาเข้ามาเพราะเขาคิดว่าคุณมีข้อมูลที่น่าสนใจ หรือเพราะความสนุกที่ได้ใช้คอมพิวเตอร์ของคุณ หรือไม่มีอะไรทำ พวกเขามักจะเรียนรู้ว่าคุณมีคอมพิวเตอร์ชนิดใดหรือมีข้อมูลอะไร เขาเป็นคนที่อยากรู้อยากเห็น แต่เขาไม่มีเจตนามุ่งร้าย อย่างไรก็ตาม เขาอาจจะทำลายระบบได้เพราะความไม่รู้

2.2.2.2 แวนดัล (vandals)

แวนดัล (vandals) เป็นบุคคลที่เขามาทำลาย เพราะเขาไม่ชอบคุณ แวนดัลเป็นปัญหาใหญ่ ถ้าคุณเป็นคนที่เล่นอินเตอร์เน็ตอยู่ทั่วไปจะต้องมีพวกที่ชอบคอยทำความรำคาญกับคนที่มีเวลาเล่นคอมพิวเตอร์ (เช่น คุณเป็นมหาวิทยาลัยที่มีเด็กมีปัญหา , หรือบริษัทที่มีลูกค้าเจ้าปัญหา) ถ้าคุณนำกำแพงสีขาวมากั้น ก็จะต้องมีคนมาเขียนหรือทำให้เลอะเทอะโดยไม่บอกเหตุผล

โชคดีที่แวนดัลพบเห็นได้น้อย คนที่ไม่มีอะไรจะป้องกันแวนดัลจะไม่ชอบพวกนี้ แวนดัลมักจะท้าทายให้มีคนค้นพบเขาและหยุดพวกเขา ซึ่งแตกต่างกับอินทรูเดอ (intruders) เพราะแวนดัลจะทำการสั้น ๆ แต่เฉียบคม โดยทั่วไปจะทำการลบข้อมูลของคุณ หรือทำความเสียหายกับอุปกรณ์คอมพิวเตอร์ของคุณ

เป็นโชคไม่ดีที่เป็นการยากที่จะหยุดแวนดัลคนที่ไม่ชอบคุณก็จะพยายามเข้ามาทำลายคุณ การโจมตีนี้จะดึงดูดพวกแวนดัลให้มาทำ แต่ไม่ดึงดูดผู้โจมตีแบบอื่น ๆ เช่น การขัดขวางการให้บริการ (denial of dervice) จะไม่ดึงดูดใจกับพวกจอยไรเดอร์ (joyriders)

2.2.2.3 สกอร์คีปเปอร์ (score keepers)

เป็นพวกที่ชอบคุยโวว่าสามารถเข้าไปในระบบผู้อื่นได้มาก ๆ คล้ายกับจอยไรเดอร์ (joyriders) และ แวนดัล (vandals) สกอร์คีปเปอร์ (score keepers) จะชอบที่จะบุกรุกเข้าไปในระบบที่มีความปลอดภัยสูง ๆ หรือมีการจัดการกับระบบอย่างดี อย่างไรก็ตามพวกเขาจะโจมตีทุกสิ่งที่สามารถเข้าไปได้ โดยเขาจะสนใจปริมาณมากกว่าคุณภาพ พวกเขาไม่มีเป้าหมายจะทำลายตามทางที่ผ่าน แต่พวกเขาอาจจะเก็บข้อมูลข่าวสารมาเก็บไว้เพื่อใช้ต่อไป (อาจจะทำการขายให้ผู้โจมตีอื่น ๆ) พวกเขาจะพยายามเลือกทางใหม่ ๆ อยู่เสมอและเป็นไปได้ว่าเขาจะมายึดระบบของคุณเป็นฐานไปโจมตีผู้อื่น

คนส่วนใหญ่จะพบหลังจากที่ถูกบุกรุกเขามาในระบบแล้ว เพราะว่ามีบางสิ่งเปลี่ยนแปลงขึ้นในระบบของคุณหรือมีคนจากที่อื่นมาบอกคุณว่าระบบของคุณโจมตีเขา

2.2.2.4 สไปซ์ (spies)

พวกนี้จะลักสิ่งของเล็ก ๆ มาเก็บไว้เพื่ออนาคต (เช่น บัตรเครดิต) ถ้าพวกเขาสามารถค้นพบความลับได้ พวกเขาก็จะนำมันไปขาย

2.3 อะไรคืออินเตอร์เน็ตไฟร์วอลล์ (Internet firewall)

ไฟร์วอลล์ (firewall) เป็นระบบความปลอดภัยบนเครือข่าย (network security) ที่มีประสิทธิภาพ โดยเปรียบเสมือนคูน้ำที่ล้อมรอบเมืองที่ป้องกันไฟลุกลามเข้าปราสาท ไฟร์วอลล์ (firewall) มีจุดประสงค์หลายอย่างดังนี้

- ไฟร์วอลล์จำกัดคนที่จะเข้ามาในส่วนควบคุม
- ไฟร์วอลล์ป้องกันการบุกรุกจากที่อื่น
- ไฟร์วอลล์จำกัดคนที่สามารถเข้ามาในส่วนควบคุม

อินเตอร์เน็ตไฟร์วอลล์ (Internet firewall) ส่วนใหญ่จะติดตั้งในจุดที่เชื่อมระหว่างเครือข่ายภายใน (Internal Network) และอินเตอร์เน็ต (Internet) โดยการสัญจรจากอินเตอร์เน็ต (Internet) หรือจากเครือข่ายภายใน (internal network) จะต้องผ่านไฟร์วอลล์ ส่วนใหญ่ไฟร์วอลล์จะต้องประกอบด้วยส่วนของฮาร์ดแวร์ (hardware) คือ เราเตอร์ (router) , โฮสต์คอมพิวเตอร์ (host computer) , การเชื่อมกันของเราเตอร์ (router) , คอมพิวเตอร์ (computer) , เครือข่าย (network) โดยเหมาะกับซอฟท์แวร์ (software) โดยโครงสร้างของการป้องกันจะขึ้นกับนโยบาย , งบประมาณ และองค์ประกอบทั้งหมด

2.4 ข้อดีของไฟร์วอลล์ (firewall)

2.4.1 ไฟร์วอลล์ (firewall) เป็นจุดรวมสำหรับการตัดสินใจอย่างปลอดภัย

ไฟร์วอลล์เปรียบเสมือนจุดตรวจสอบ (check point) ในการส่งผ่านข้อมูลเข้าออกจะต้องผ่านจุดตรวจสอบ (check point) ที่เป็นช่องแคบเดี่ยว ซึ่งจุดนี้เป็นจุดที่เครือข่ายติดต่อกับอินเตอร์เน็ต (Internet)

2.4.2 ไฟร์วอลล์ (firewall) สามารถบังคับได้ตามต้องการ

ไฟร์วอลล์เปรียบเสมือนตำรวจสำหรับการให้บริการ บังคับให้ทำตามกฎเกณฑ์นโยบาย โดยอนุญาตให้บริการที่อนุมัติแล้วสามารถผ่านโดยขึ้นอยู่กับกฎที่ตั้งขึ้นมา

2.4.3 ไฟร์วอลล์ (firewall) สามารถเก็บรายละเอียดของสิ่งที่เกิดขึ้นได้อย่างมีประสิทธิภาพ

เพราะการสัญจรทุกอย่างจะต้องผ่านไฟร์วอลล์ ดังนั้นไฟร์วอลล์จึงเป็นที่ที่ดีที่สุดที่จะเก็บข้อมูลเกี่ยวกับสิ่งที่ระบบและเครือข่ายใช้หรือไม่ใช้ เช่น ไฟร์วอลล์สามารถบันทึกได้ว่ามีอะไรเกิดขึ้นบ้างระหว่างเครือข่ายภายใน (internal network) กับเครือข่ายภายนอก (external network)

2.4.4 ไฟร์วอลล์ (firewall) มีการเปิดเผยคุณอย่างจำกัด

ไฟร์วอลล์สามารถแบ่งส่วนของเครือข่ายออกมาได้ ซึ่งส่วนนี้อาจจะถูกทำให้น่าเชื่อถือที่สุด

2.5 ข้อเสียของไฟร์วอลล์ (firewall)

2.5.1 ไฟร์วอลล์ (firewall) ไม่สามารถป้องกันจากผู้ประสงค์ร้ายภายใน

ถ้าผู้บุกรุกอยู่ภายในไฟร์วอลล์แล้ว ไฟร์วอลล์ก็ไม่สามารถทำอะไรได้ ผู้ใช้ภายในสามารถลักขโมยข้อมูลทำลายฮาร์ดแวร์และซอฟท์แวร์และแก้ไขโปรแกรมได้

2.5.2 ไฟร์วอลล์ (firewall) ไม่สามารถป้องกันจากการติดต่อที่ไม่ได้ผ่านมัน

ไฟร์วอลล์สามารถควบคุมการสัญจรที่ผ่านมัน แต่ถ้าไม่มีอะไรไฟร์วอลล์ก็ไม่สามารถทำอะไรได้ ตัวอย่างเช่น ถ้าไซต์ (site) อนุญาตให้เข้าถึงโดยการโทรเข้าแต่อยู่หลังไฟร์วอลล์ ไฟร์วอลล์ก็จะไม่สามรถป้องกันได้

2.5.3 ไฟร์วอลล์ (firewall) ไม่สามารถป้องกันไวรัส (Viruses)

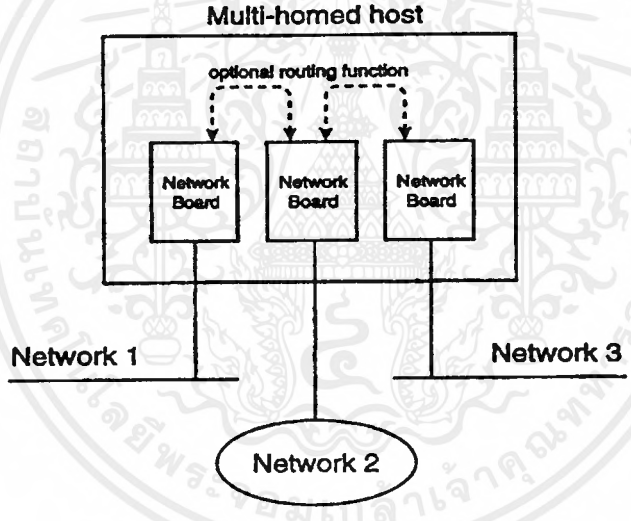
ไฟร์วอลล์ไม่สามารถป้องกันพีซี (PC) และแมคอินทอชไวรัส (macintosh virus) ที่เกิดบนเครือข่ายได้แม้ว่าไฟร์วอลล์จะสแกน (scan) ทุก ๆ การสัญจรที่เข้ามาเครือข่ายภายใน แต่มันจะสแกนเฉพาะจุดเริ่มต้น , จุดปลายทาง และหมายเลขพอร์ต (port) แต่ไม่ได้ดูรายละเอียดของข้อมูล

บทที่ 3

สถาปัตยกรรมของไฟร์วอลล์ (firewall architecture)

3.1 สถาปัตยกรรมดูอัลโฮมโฮสต์ (Dual-homed host architecture)

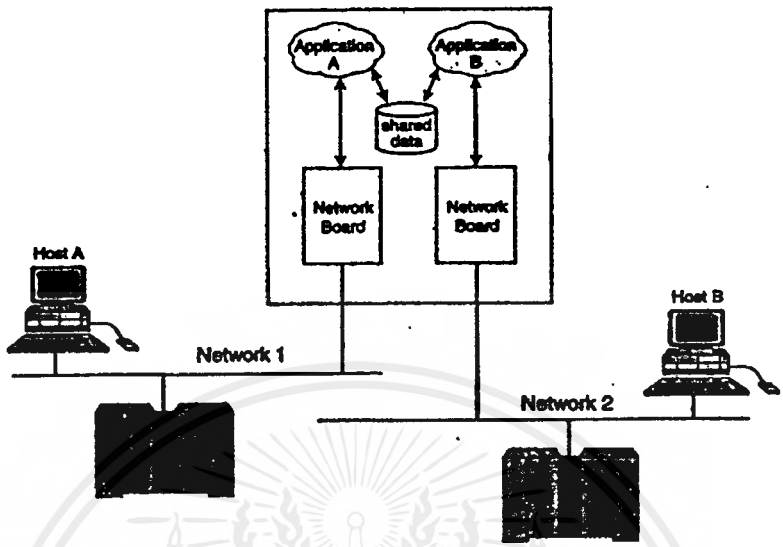
ในเครือข่ายแบบทีซีพีไอพี (TCP/IP) เทอมของมัลติโฮมโฮสต์ (Multi-homed host) คือ โฮสต์ที่มีหลายแผงการเชื่อมต่อเครือข่าย (network interface board) ซึ่งโดยทั่วไปแผงการเชื่อมต่อเครือข่ายจะถูกเชื่อมกับเครือข่าย ดังรูป 3.1 ในอดีตมัลติโฮมโฮสต์นี้สามารถกำหนดเส้นทางระหว่างเครือข่าย เทอมเกตเวย์ (gateway) ถูกใช้ระบุหน้าที่ในการหาเส้นทาง (routing function) แต่วันนี้เทอมของเราเตอร์ (router) ถูกใช้ระบุหน้าที่ในการหาเส้นทางเพราะว่า เทอมของเกตเวย์ถูกจองไว้สำหรับหน้าที่ที่เหมาะกับชั้นสูง ๆ ของโอเอสไอโมเดล (OSI model)



รูป 3.1 แสดงคลาสสิคมัลติโฮมโฮสต์ (Classic multi-home host)

ถ้าหน้าที่ในการหาเส้นทาง (routing function) ในมัลติโฮมโฮสต์ไม่สามารถทำงานได้โฮสต์จะทำการแบ่งแยกเส้นทางของเครือข่ายระหว่างเครือข่ายที่มันติดต่ออยู่ และเครือข่ายอื่น ๆ ที่สามารถจัดการกับแอพพลิเคชั่น (applications) บนมัลติโฮมโฮสต์โดยเฉพาะอย่างยิ่งถ้าแอพพลิเคชั่นอนุญาตให้เครือข่ายสามารถที่จะแบ่งปันข้อมูล (share data) ได้ ดูอัลโฮมโฮสต์ (Dual-homed host) เป็นตัวอย่างพิเศษของมัลติโฮมโฮสต์ซึ่งมีสองเครือข่ายมาอินเตอร์เฟส (interface) กัน โดยที่ไม่มีการระบุหน้าที่ในการหาเส้นทาง ดังรูป 3.2 แสดงตัวอย่างของดูอัลโฮมโฮสต์ (Dual-homed host) ซึ่งไม่มีการใช้การระบุหน้าที่ในการหาเส้นทาง โฮสต์เอ (host A) อยู่บนเครือข่ายที่หนึ่งซึ่งสามารถเข้าถึงแอพพลิเคชั่นเอบนดูอัลโฮมโฮสต์ที่คล้ายคลึงกัน , โฮสต์บี (host B) สามารถเข้าถึงแอพพลิเคชั่นบี (application B) บนดูอัลโฮมโฮสต์ แอพพลิเคชั่นทั้งสองบนดูอัลโฮมโฮสต์สามารถแบ่งปันข้อมูลได้ จึงเป็นไปได้ว่าโฮสต์เอและ โฮสต์บีสามารถ

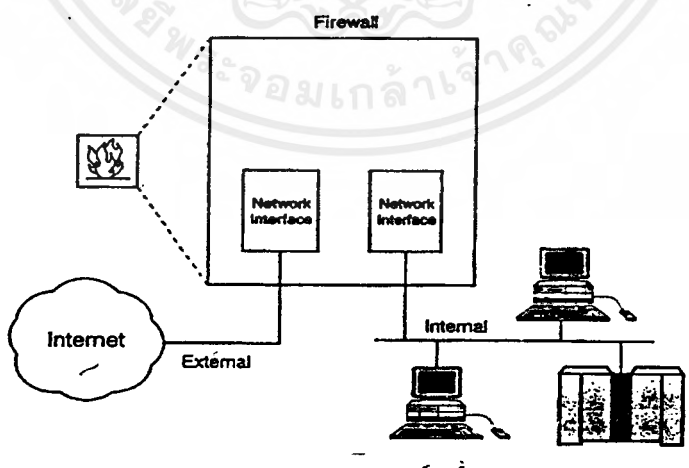
แลกเปลี่ยนข้อมูลผ่านข้อมูลที่ถูกแบ่งปันบนดูอัลโฮมโฮสต์และยังไม่แลกเปลี่ยนเส้นทางของเครือข่าย ระหว่างสองส่วนของเครือข่ายที่เชื่อมต่อกับดูอัลโฮมโฮสต์



รูป 3.2 แสดงดูอัลโฮมโฮสต์ (Dual-homed host)

3.1.1 ดูอัลโฮมโฮสต์ (Dual-homed host) ที่เป็นไฟร์วอลล์

ดูอัลโฮมโฮสต์สามารถแยกเครือข่ายภายใน (ซึ่งมีเครือข่ายอย่างน้อย 2 เครือข่าย) และเครือข่ายภายนอก ดังรูป 3.3 เพราะดูอัลโฮมโฮสต์ไม่ขึ้นกับการส่งแบบทีซีพีไอพี (TCP/IP) มันสามารถขัดขวางทุก ๆ ไอพี (IP) ระหว่างเครือข่ายภายในและเครือข่ายภายนอกที่ไม่น่าไว้วางใจ

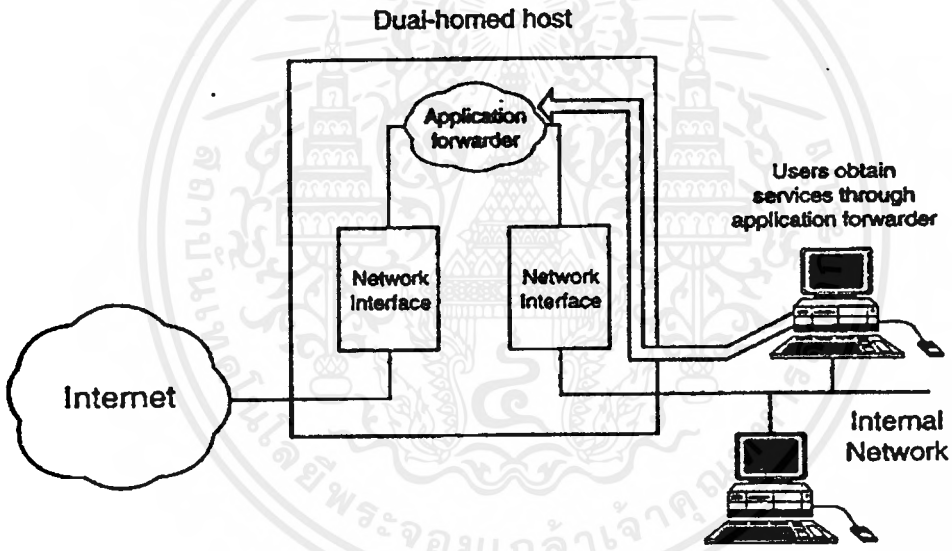


รูป 3.3 แสดงดูอัลโฮมโฮสต์ (Dual-homed host) ที่เป็นไฟร์วอลล์

การบริการของอินเตอร์เน็ต เช่น จดหมาย (mail) และข่าวสาร (news) เป็นสิ่งที่จำเป็นพื้นฐานในการบริการเก็บและส่ง ถ้าการบริการเหล่านี้ทำงานบนดูอัลโฮมโฮสต์จะทำให้สามารถกำหนดโครงสร้างในการส่งแอพพลิเคชั่นจากเครือข่ายหนึ่งถึงที่อื่น ๆ ถ้าข้อมูลแอพพลิเคชั่นจำเป็นที่จะต้องข้าม

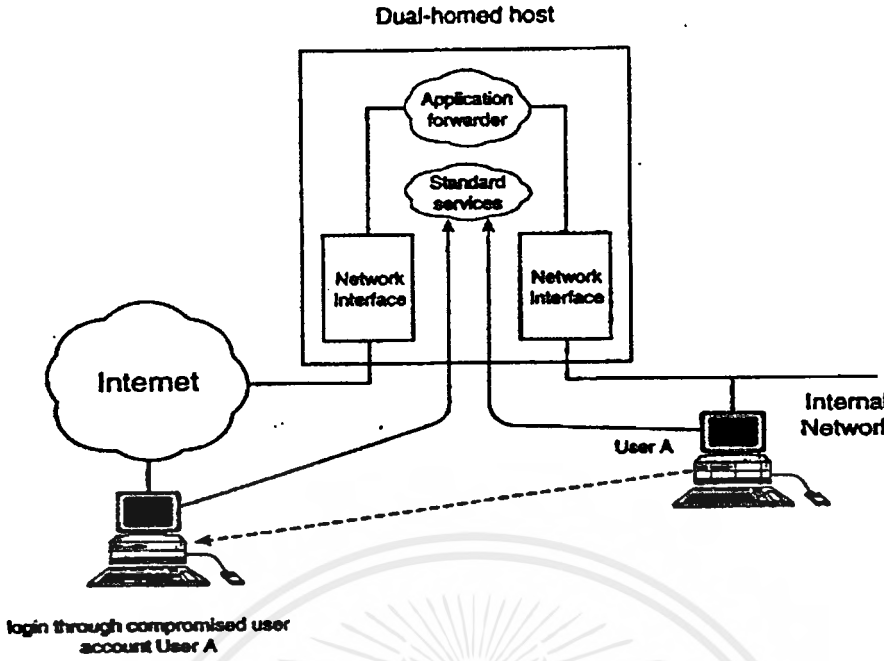
ไฟร์วอลล์ ผู้ส่งต่อแอพพลิเคชั่น (application forwarder) สามารถติดตั้งให้ทำงานบนดูอัลโฮมโฮสต์ได้ ดังรูปที่ 3.4 ผู้ส่งต่อแอพพลิเคชั่น (application forwarder) คือ ซอฟท์แวร์ (software) พิเศษ ที่ถูกใช้ส่งระหว่างสองเครือข่ายที่ติดต่อกัน การเข้าถึงอย่างอื่น คือ อนุญาตให้ผู้ใช้เข้ามาล็อกอิน (login) ในดูอัลโฮมโฮสต์และเข้าถึงบริการภายนอกจากเครือข่ายภายนอกที่เชื่อมติดต่อกับดูอัลโฮมโฮสต์ ดังรูปที่ 3.5

ถ้าผู้ส่งต่อแอพพลิเคชั่น (application forwarders) ที่ถูกใช้การสัญจรของแอพพลิเคชั่นจะไม่สามารถผ่านดูอัลโฮมโฮสต์ไฟร์วอลล์นอกเสียจากว่าผู้ส่งต่อแอพพลิเคชั่นถูกกำหนดไว้บนระบบไฟร์วอลล์ ถ้าผู้ใช้ได้รับการอนุญาตจากไฟร์วอลล์โดยตรง ดังรูปที่ 3.5 ระบบรักษาความปลอดภัยของไฟร์วอลล์สามารถประนีประนอมได้ เพราะว่าดูอัลโฮมโฮสต์ไฟร์วอลล์เป็นจุดศูนย์รวมการติดต่อระหว่างเครือข่ายภายนอกและภายใน ดังนั้นดูอัลโฮมโฮสต์ไฟร์วอลล์ก็จะอยู่ในบริเวณที่มีความเสี่ยง (zone of risk) ถ้าผู้ใช้เลือกรหัสผ่าน (password) ที่อ่อนแอ ในบริเวณที่มีความเสี่ยงก็สามารถขยายมาที่ระบบเครือข่ายภายในทำให้ผิดกับจุดประสงค์ของดูอัลโฮมโฮสต์ไฟร์วอลล์ได้



รูป 3.4 แสดงดูอัลโฮมโฮสต์ซึ่งมีผู้ส่งต่อแอพพลิเคชั่น (application forwarder)

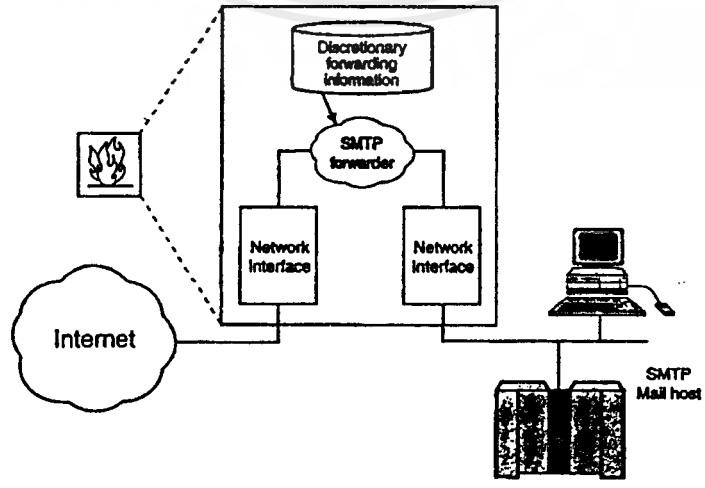
ผู้จัดการระบบรักษาความปลอดภัย จะไม่ทำการสร้างบัญชีของผู้ใช้ (user account) ให้เข้ามาที่ ไฟร์วอลล์ ดังนั้นไฟร์วอลล์ควรจะถูกใช้ได้เฉพาะผู้ใช้ที่มีสิทธิโดยแท้จริงเท่านั้นที่จะผ่านเข้ามาได้



รูป 3.5 แสดงความไม่ปลอดภัยเมื่อผู้ใช้ล็อกอิน (login) เข้ามาในดูอัลโฮมโฮสต์

ถ้าหากมีการเก็บรายละเอียดของผู้ใช้การล็อกอินของผู้ใช้ (users login) ไว้อย่างดี มันจะถูกบุกรุกเมื่อมีการผิดเพี้ยนของระบบรักษาความปลอดภัย ดังนั้นควรที่จะมีการบันทึกไว้ว่ามีใครเข้ามาบ้าง

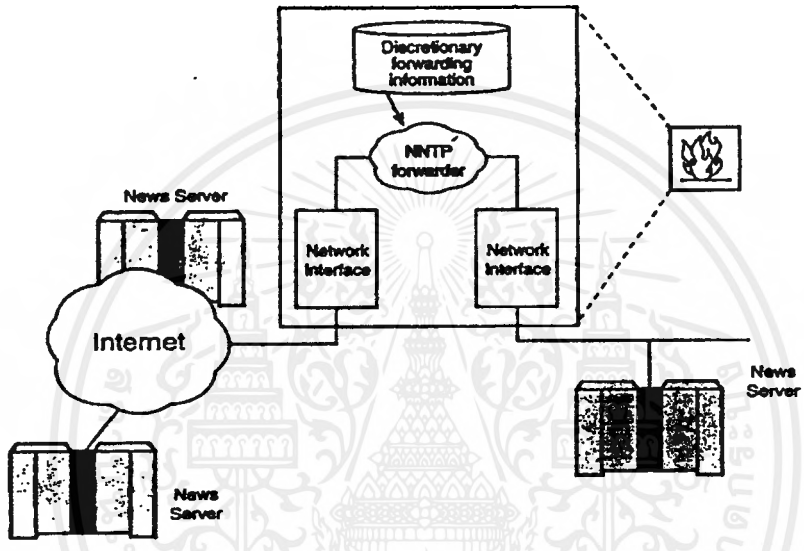
ตัวอย่างของการบริการเก็บและส่ง คือ เอสเอ็มทีพี (SMTP-mail) และเอ็นเอ็นทีพี (NNTP-news) ดังรูปที่ 3.6 แสดงการติดตั้งตัวระมัดระวังในการส่งข้อมูล (discretionary forwarding) ของข้อความเมล์ (mail) ระหว่างเครือข่ายภายนอกที่ไม่น่าไว้วางใจกับเครือข่ายภายใน ดังรูปที่ 3.7 แสดงการติดตั้งตัวระมัดระวังในการส่งข้อมูลสำหรับข้อความข่าวสาร (news) ระหว่างเซิร์ฟเวอร์ข่าวสาร (news Servers) บนเครือข่ายภายนอกและเครือข่ายภายใน



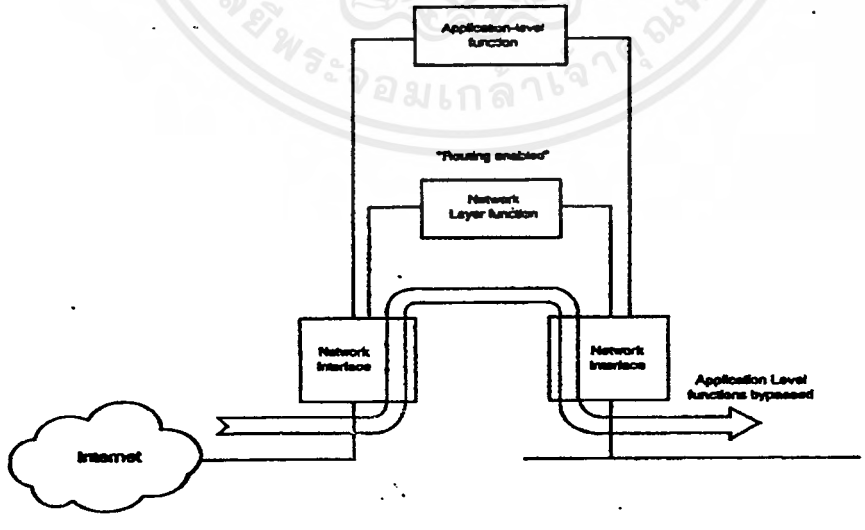
รูป 3.6 แสดงดูอัลโฮมโฮสต์ที่มีผู้ส่งต่อเมล์ (mail forwarder)

ดูอัลโฮมโฮสต์เป็นโครงสร้างพื้นฐานที่ใช้ในไฟร์วอลล์ ความสามารถที่ใช้เมื่อฉุกเฉินของดูอัลโฮมโฮสต์ คือ เมื่อการหาเส้นทาง (routing) ใช้ไม่ได้ และมีเพียงเส้นทางระหว่างส่วนหนึ่งของเครือข่าย (network segment) ที่ผ่านทางชั้นแอพพลิเคชั่น (application layer) เมื่อตัวหาเส้นทางมีปัญหา ไอพี (IP) ก็สามารถส่งไปได้ มันคือไม่ผ่านชั้นแอพพลิเคชั่นของดูอัลโฮมโฮสต์ไฟร์วอลล์ ดังรูปที่ 3.8

ไฟร์วอลล์ส่วนมากจะสร้างขึ้นรอบ ๆ ระบบยูนิกซ์ (UNIX) ซึ่งบนการทำงานของระบบยูนิกซ์นั้นหน้าที่การจัดหาเส้นทาง (rounting function) จะถูกจัดตั้งให้ทำงานได้ ดังนั้นมันจึงเป็นสิ่งสำคัญในการตรวจสอบว่า หน้าที่การจัดหาเส้นทางในดูอัลโฮมโฮสต์ไฟร์วอลล์ทำงานได้หรือไม่ และถ้ามันไม่ทำงาน คุณควรจะรู้ว่ามันไม่ทำงานได้อย่างไร



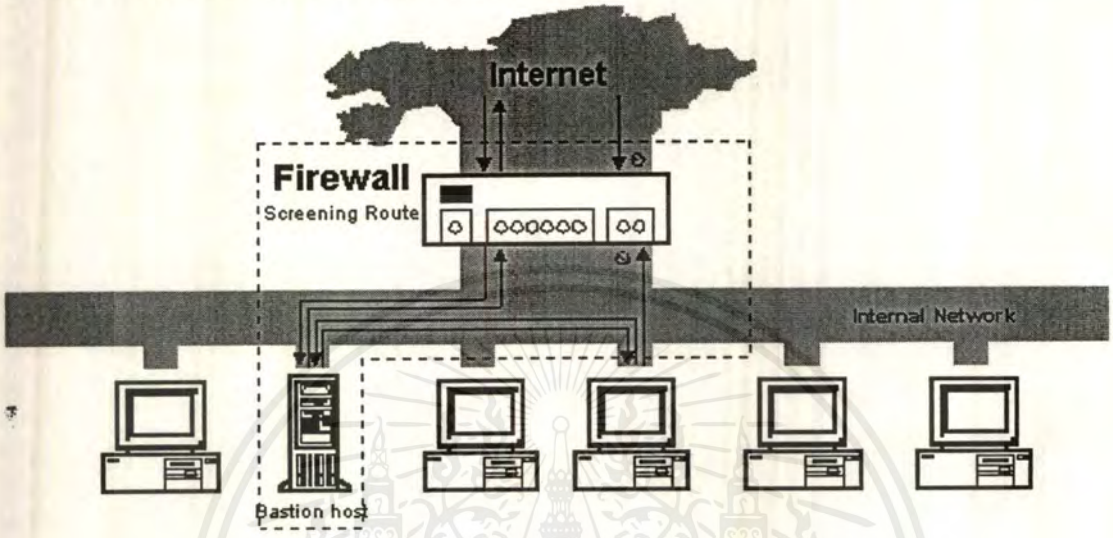
รูป 3.7 ดูอัลโฮมโฮสต์ที่มีตัวส่งต่อข่าวสาร (news forwarder)



รูป 3.8 แสดงการคอนฟิก (config) ที่ผิดพลาดบนดูอัลโฮมโฮสต์ไฟร์วอลล์

3.2 สถาปัตยกรรมสกรีนโฮสต์ (Screened host architectures)

ขณะที่สถาปัตยกรรมแบบดูอัลโฮมโฮสต์จัดการกับการบริการที่เกี่ยวกับหลาย ๆ เครือข่าย (แต่ไม่มีการหากเส้นทาง) ส่วนสถาปัตยกรรมแบบสกรีนโฮสต์ (Screened host architecture) จะจัดการกับบริการ (service) จากโฮสต์ที่จะเข้ามาติดต่อกับเครือข่ายภายใน โดยการแบ่งการหาเส้นทางซึ่งการป้องกันขั้นแรกจะใช้ แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering)



รูป 3.9 แสดงสถาปัตยกรรมแบบสกรีนโฮสต์ (Screened host architecture)

แบสชั่นโฮสต์ (bastion host) จะต้องอยู่บนเครือข่ายภายใน แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) บนสกรีนนิ่งเราเตอร์ (screening router) จะกำหนดเส้นทางให้ผ่านแบสชั่นโฮสต์ก่อนทุกครั้งจึงจะติดต่อได้

โครงสร้างของแพ็กเก็ตฟิลเตอร์ริ่งในสกรีนนิ่งเราเตอร์อาจจะกระทำอย่างใดอย่างหนึ่งดังต่อไปนี้

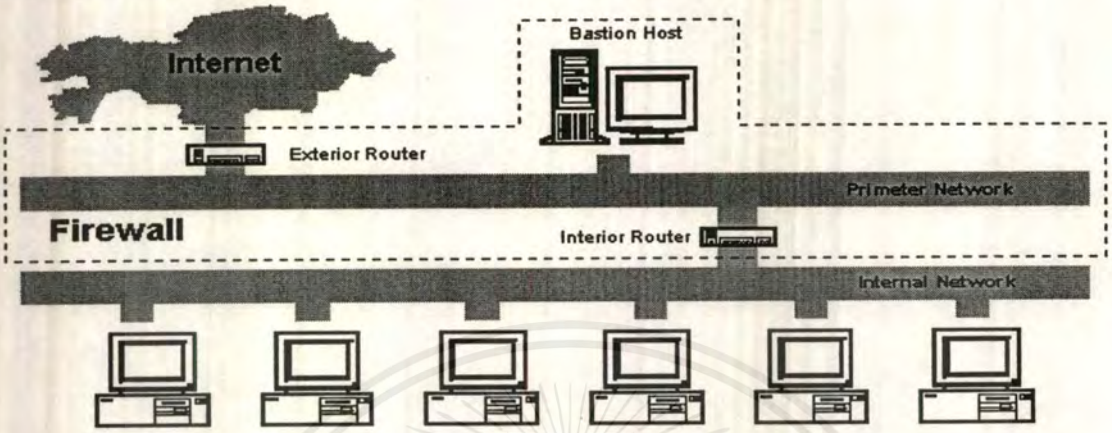
- สำหรับโฮสต์ที่แน่นอน จะอนุญาตให้โฮสต์ภายใน (internal host) ติดต่อกับโฮสต์บนอินเตอร์เน็ตได้
- ไม่อนุญาตทุกการติดต่อจากโฮสต์ภายใน

คุณสามารถผสมและรวม (match) การเข้าถึงเหล่านี้ในแต่ละบริการ บางบริการอาจจะอนุญาตให้ผ่านแพ็กเก็ตฟิลเตอร์ริ่งโดยตรง ขณะที่อื่น ๆ อาจจะไม่อนุญาตให้ผ่านได้โดยตรง โดยขึ้นอยู่กับข้อกำหนดของแต่ละไซต์ (site) ที่ตั้งขึ้น

ในสถาปัตยกรรมนี้แพ็กเก็ตผ่านจากอินเตอร์เน็ตเข้าสู่เครือข่ายภายในโดยตรง จึงดูเหมือนว่าจะมีความเสี่ยงมากกว่าดูอัลโฮมโฮสต์ที่ไม่มีแพ็กเก็ตภายนอกเข้าถึงเครือข่ายภายในได้ ในทางปฏิบัติดูอัลโฮมโฮสต์อาจจะเสียหายได้ง่าย ซึ่งทำให้แพ็กเก็ตผ่านเข้ามาได้ นอกจากนี้มันง่ายกว่าในการป้องกันด้วยเราเตอร์ (router) ซึ่งสามารถจำกัดกลุ่มของการบริการมากกว่าป้องกันด้วยโฮสต์ด้วยจุดนี้สกรีนโฮสต์จึงรักษาความปลอดภัยได้ดีกว่าและนิยมใช้มากกว่าดูอัลโฮมโฮสต์

3.3 สถาปัตยกรรมแบบสกรีนสับเน็ต (Screened subnet architectur)

สถาปัตยกรรมแบบสกรีนสับเน็ต (Screened subnet architecture) ได้เพิ่มชั้นพิเศษของความปลอดภัยบนสถาปัตยกรรมแบบสกรีนโฮสต์โดยเพิ่มชั้นของเครือข่ายเพอริมิเตอร์ (perimeter network) เพื่อแบ่ง เครือข่ายภายในจากอินเตอร์เน็ต



รูป 3.10 แสดงสถาปัตยกรรมแบบสกรีนสับเน็ต (Screened subnet architecture) ที่ใช้เราเตอร์สองตัว

โดยทั่วไปแบสชั่นโฮสต์ (bastion host) เป็นเครื่องที่อ่อนแอบนเครือข่าย ถึงแม้ว่าจะพยายามป้องกันแต่ก็มักจะถูกบุกรุก สำหรับในสกรีนโฮสต์ถ้าแบสชั่นโฮสต์ถูกบุกรุก จะไม่มีอะไรที่จะป้องกันระหว่างมันกับเครื่องกลภายใน ดังนั้นถ้ามีคนบุกรุกเข้าไปในแบสชั่นโฮสต์ของสกรีนโฮสต์ได้ถือว่าเขาได้รางวัลใหญ่ (jackpot)

โดยการแบ่งแบสชั่นโฮสต์ไว้บนเครือข่ายเพอริมิเตอร์ (perimeter network) คุณก็จะสามารถลดผลกระทบที่เกิดจากการที่มีคนบุกรุกแบสชั่นโฮสต์ มันจะช่วยป้องกันผู้บุกรุกในการเข้าถึงแต่ไม่ถึงกับทั้งหมด

โดยทั่วไปสถาปัตยกรรมแบบสกรีนสับเน็ตจะมีสกรีนนิ่งเราเตอร์ (screening router) สองตัวซึ่งแต่ละตัวจะเชื่อมต่อกับเครือข่ายเพอริมิเตอร์ ดังรูปที่ 3.10 ในการบุกรุกเข้าในเครือข่ายภายในจะต้องผ่านเราเตอร์ถึงสองตัว อย่างไรก็ตามถ้าระบบฟิลเตอร์ริ่ง (filtering) ระหว่างแต่ละชั้นอนุญาตในสิ่งเดียวกันชั้นที่เพิ่มขึ้นก็ไม่ได้เพิ่มความปลอดภัยเลย

3.3.1 เครือข่ายเพอริมิเตอร์ (perimeter network)

เครือข่ายเพอริมิเตอร์ (perimeter network) เป็นชั้นพิเศษของความปลอดภัยโดยเพิ่มแทรกขึ้นระหว่างเครือข่ายภายนอก (external network) และเครือข่ายที่จะทำการป้องกัน (Protected Network) ถ้าผู้บุกรุกเข้ามาที่ชั้นนอกของไฟร์วอลล์ เครือข่ายเพอริมิเตอร์ก็จะทำการป้องกันอีกชั้นหนึ่ง

บนเครือข่ายเพอริมิเตอร์ถ้ามีคนบุกรุกเข้ามาในแบสชั่นโฮสต์บนเครือข่ายเพอริมิเตอร์ เขาก็สามารถเห็นได้เพียงการสัญจร (traffic) บนเครือข่าย ทุก ๆ การสัญจรบนเครือข่ายเพอริมิเตอร์ควรจะเป็นการเข้าถึงและออกจากแบสชั่นโฮสต์หรือไปถึงหรือออกจากอินเตอร์เน็ต

เพราะว่าไม่มีการกำหนดเส้นทางของเครือข่ายภายในให้ผ่านเครือข่ายเพอริมิเตอร์ ดังนั้นเครือข่ายเพอริมิเตอร์ก็จะปลอดภัยจากการบุกรุกถ้าแบสชั่นโฮสต์รัดกุม

3.3.2 แบสชั่นโฮสต์ (bastion host)

เป็นโฮสต์ที่เป็นจุดรวมของการติดต่อกับโลกภายนอก เช่น

- สำหรับรับและส่งอีเมล์ (SMTP)
- รับการติดต่อเอฟทีพี (FTP) ไปที่ไซต์ของแอโนนีมัสเซิร์ฟเวอร์ (anonymous server)
- รับโดเมนเนมเซอวิส (Domain Name Service : DNS) คำถามเกี่ยวกับไซต์และ อื่นๆ

การบริการออกสู่ภายนอกจะกระทำได้ดังนี้ จากไคลเอนต์ภายใน (internal clients) ถึงเซิร์ฟเวอร์บนอินเตอร์เน็ต (Internet)

- ตั้งแพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) บนทั้งเราเตอร์ภายนอก (exterior router) และเราเตอร์ภายใน (interior router) อนุญาตให้ไคลเอนต์ภายใน (internal client) เข้าถึงเซิร์ฟเวอร์ภายนอก (external server) ได้โดยตรง
- ตั้งพร็อกซีเซิร์ฟเวอร์ (proxy server) ให้ทำงานบนแบสชั่นโฮสต์ ในการอนุญาตไคลเอนต์ภายในให้เข้าถึงเซิร์ฟเวอร์ภายนอก (external server) ทำไม่ได้โดยตรง แต่จะทำโดยจะตั้งแพ็กเก็ตฟิลเตอร์ริ่งให้ไคลเอนต์ภายในคุยกับพร็อกซีเซิร์ฟเวอร์บนแบสชั่นโฮสต์แต่ห้ามการติดต่อกันโดยตรงระหว่างไคลเอนต์ภายในกับโลกภายนอก

ในแต่ละกรณีแพ็กเก็ตฟิลเตอร์ริ่งอนุญาตให้แบสชั่นโฮสต์ติดต่อไปและตอบรับการติดต่อกับโฮสต์บนอินเตอร์เน็ต โดยจะขึ้นอยู่กับข้อกำหนดของแต่ละไซต์

3.3.3 เราเตอร์ภายใน (interior router)

เราเตอร์ภายใน บางครั้งจะเรียกว่าโช้คเราเตอร์ (choke router) ในไฟร์วอลล์ ทำหน้าที่ป้องกันเครือข่ายภายในจากอินเตอร์เน็ตและจากเครือข่ายเพอริมิเตอร์ โดยเราเตอร์ภายในจะคอยเลือกการบริการที่ออกจากเครือข่ายภายในไปที่อินเตอร์เน็ตโดยจะมีการตรวจสอบด้วยแพ็กเก็ตฟิลเตอร์ริ่ง โดยการบริการที่มักจะให้มีการติดต่อออกได้แก่ เทลเน็ต (TELNET) , เอฟทีพี (FTP) , WAIS , Archie , Gopher และแอพพลิเคชั่นอื่นๆ ที่เหมาะกับความต้องการ

การบริการที่เราเตอร์ภายในจะให้อนุญาตระหว่างแบสชั่นโฮสต์กับเครือข่ายภายในโดยไม่จำเป็นต้องเป็นการบริการเดียวกับที่ให้อนุญาตระหว่างอินเตอร์เน็ตกับเครือข่ายภายใน เหตุผลที่จำกัดการบริการระหว่างแบสชั่นโฮสต์และเครือข่ายภายใน คือ ลดจำนวนโอกาสบุกรุกจากแบสชั่นโฮสต์

3.3.4 เราเตอร์ภายนอก (exterior router)

เราเตอร์ภายนอก บางครั้งเรียกว่าแอคเซสเราเตอร์ (access router) ทำหน้าที่ป้องกันทั้งเครือข่ายเพอริมิเตอร์และเครือข่ายภายในจากอินเตอร์เน็ต ในทางปฏิบัติเราเตอร์ภายนอกตั้งใจจะให้อนุญาตเกือบทุก ๆ สิ่งที่ออกจากเครือข่ายเพอริมิเตอร์ และทำแพ็กเก็ตฟิลเตอร์ริ่งน้อยมาก

แพ็กเก็ตฟิลเตอร์ริ่งที่ใช้บนเราเตอร์ภายนอกเป็นพิเศษที่ป้องกันเครื่องบนเครือข่ายเพอริมิเตอร์ (ซึ่งมีแบสชั่นโฮสต์และเราเตอร์ภายใน) โดยทั่วไปจะไม่มีการป้องกันมาก เพราะโฮสต์บนเครือข่ายเพอริมิเตอร์จะทำการป้องกันเริ่มแรกผ่านโฮสต์ความปลอดภัย (host security)

3.4 การปรับเปลี่ยนสถาปัตยกรรมของไฟร์วอลล์ (firewall architecture)

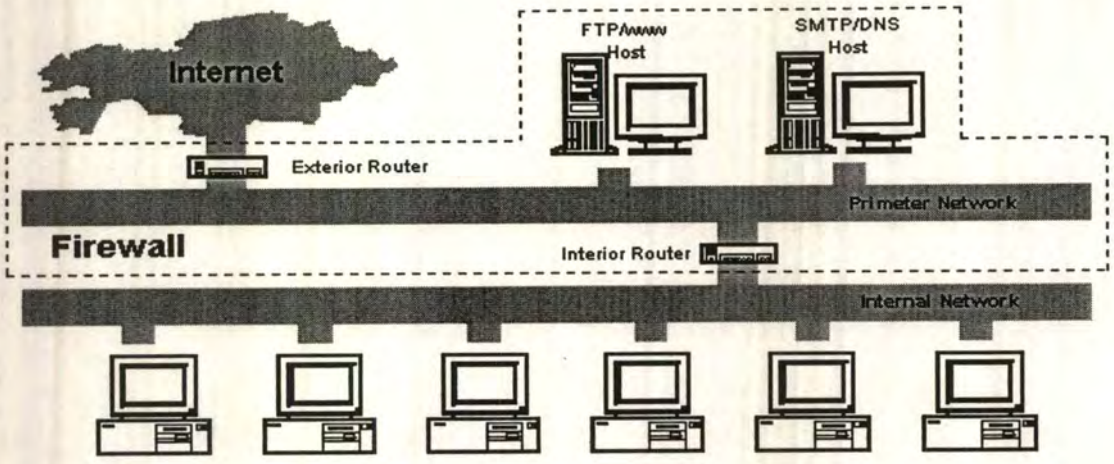
3.4.1 สามารถใช้แบสชั่นโฮสต์ได้หลายตัว (multiple bastion host)

เราสามารถใช้แบสชั่นโฮสต์หลายๆตัวได้ เพื่อเพิ่มประสิทธิภาพและต้องการการแบ่งข้อมูลหรือเซิร์ฟเวอร์ โดยเราอาจให้แบสชั่นโฮสต์ตัวหนึ่งเป็นตัวหลักของผู้ใช้ของคุณ เช่น เอสเอ็มทีพีเซิร์ฟเวอร์ (SMTP server) , พร็อกซีเซิร์ฟเวอร์ (proxy server) และอื่นๆ ขณะที่โฮสต์อื่น ๆ จัดการเกี่ยวกับอินเตอร์เน็ตที่ไม่เกี่ยวข้องกับผู้ใช้ของคุณ เช่น แอโนนีมัสเอฟทีพีเซิร์ฟเวอร์ (anonymous FTP server) ดังนั้นการทำงานของผู้ใช้ของคุณจะไม่ถูกดึงให้ช้าเพราะผู้ใช้ภายนอก

ในด้านการแสดงผล ในการใช้แบสชั่นโฮสต์หลายตัวบางการบริการมีความหนาแน่นมาก เช่น ยูสเน็ตนิวส์ (Usenet news) มันเป็นไปได้ที่เราจะทำการบริการเดียวกันลงในแบสชั่นโฮสต์หลายตัว แต่มันจะยากในการทำการโหลดแบบสมมาตร (loading balance) แต่บริการส่วนมากต้องการเซิร์ฟเวอร์โดยเฉพาะ การที่เราจะใช้แบสชั่นโฮสต์หลายตัวให้ดีที่สุดควรจะมีการคาดคะเนการใช้ว่าบริการใดมีควรจะมีเซิร์ฟเวอร์โดยเฉพาะ

ในด้านการเกินความต้องการ (redundancy) ถ้าไฟร์วอลล์คอนฟิกเป็นแบสชั่นโฮสต์หลายตัว คุณควรจะมีการสำรอง คือ ถ้าตัวหนึ่งบกพร่อง (fail) , บริการสามารถถูกจัดการได้โดยอีกตัวหนึ่ง แต่ต้องระวังว่ามีเพียงการบริการบางอย่างเท่านั้นที่เข้าถึงได้

อาจจะใช้แบสชั่นโฮสต์หลายตัวในการเก็บกลุ่มข้อมูลของการบริการ จากการแทรกแซงกันและกัน นอกจากนี้ยังสามารถแบ่งความปลอดภัย เช่น ออกแบบให้ด้านหนึ่งเป็นเอชทีทีพีเซิร์ฟเวอร์ (HTTP server) เพื่อใช้สำหรับผู้ใช้บนอินเตอร์เน็ตและตัวใช้เป็นเจเนอรัลพับบลิก (general public) ซึ่งช่วยในการลดโหลด หรือเพิ่มประสิทธิภาพ

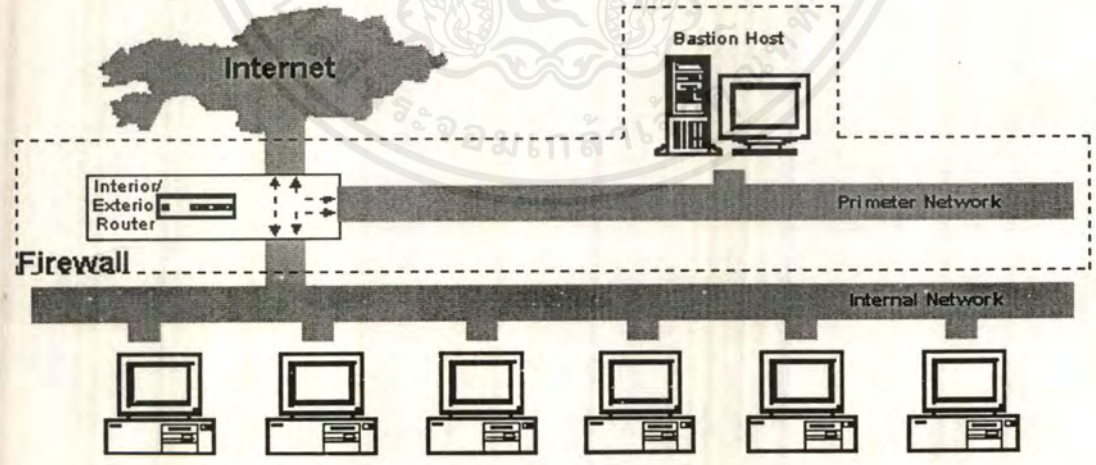


รูป 3.11 แสดงสถาปัตยกรรมโดยใช้แบสชั่นโฮสต์ 2 ตัว

3.4.2 สามารถรวมเราเตอร์ภายใน (interior router) และ เราเตอร์ภายนอก (exterior router)

สามารถใช้เราเตอร์ภายในและเราเตอร์ภายนอกเป็นเราเตอร์ตัวเดียว (single router) แต่จะต้องมีเราเตอร์ที่มีความสามารถเพียงพอและยืดหยุ่นได้ โดยทั่วไปคุณต้องการเราเตอร์ที่ทำการกรองทั้งเข้าและออกจากจุดเชื่อมต่อ

ถ้าเรารวมเราเตอร์ภายในและเราท์เตอร์ภายนอก ดังรูปที่ 3.12 จะมีเครือข่ายเพอริมิเตอร์และเครือข่ายภายในติดต่อกับเราเตอร์ บางการสัญจรสามารถไหลได้โดยตรงระหว่างเครือข่ายภายในและอินเตอร์เน็ต (การสัญจรอนุญาตโดยตั้งกฎของแพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering rule) สำหรับเราเตอร์) และการสัญจรอื่นก็สามารถไหลผ่านระหว่างเครือข่ายเพอริมิเตอร์และอินเตอร์เน็ต หรือเครือข่ายเพอริมิเตอร์กับเครือข่ายภายใน (การสัญจรนี้ คือ พร็อกซี)



รูป 3.12 แสดงสถาปัตยกรรมโดยรวมเราท์ตอร์ภายในและเราเตอร์ภายนอก

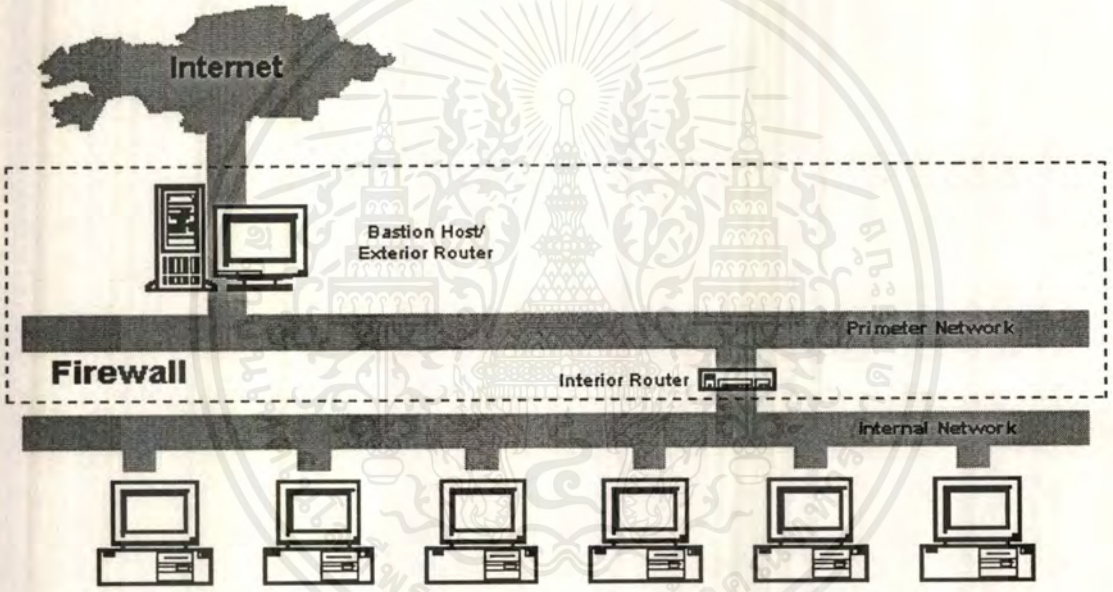
สถาปัตยกรรมนี้คล้ายกับสถาปัตยกรรมของสกรีนโฮสต์ ซึ่งทำให้ไซต์ที่ไม่มั่นคงรวมเอาเราเตอร์เข้าด้วยกัน โดยทั่วไปเราเตอร์จะทำการปกป้องได้ง่ายกว่าโฮสต์ แต่มันจะถูกทะลุผ่านได้

3.4.3 สามารถรวมแบสชั่นโฮสต์ (bastion host) และเราทีเตอร์ภายนอก (exterior router)

เราสามารถใช้กลไกซิงเกิลโฮม (single homed machine) ที่เป็นทั้งแบสชั่นโฮสต์และเราเตอร์ภายนอก เช่น ในกรณีที่มีเพียงแค่ไดอัลอัพซลิป (dial-up SLIP) หรือ การเชื่อมต่อพีพีพี (PPP connection) ในอินเตอร์เน็ต กรณีนี้สามารถใช้พีพีพีแพ็กเกจแบบมอร์นิ่งสตาร์ (morning star PPP package) ในแบสชั่นโฮสต์และให้มันทำงานเป็นทั้งแบสชั่นโฮสต์และเราเตอร์ภายนอก ซึ่งมีหน้าที่ในการทำงานเหมือนกับสามเครื่อง (แบสชั่นโฮสต์ , เราเตอร์ภายใน และ เราเตอร์ภายนอก)

ในการรวมแพ็กเกจ เช่น แพ็กเกจแบบมอร์นิ่งสตาร์ ที่กล่าวมาจะมีคุณสมบัติที่ดีในการทำแพ็กเก็ตฟิลเตอร์ริ่งอย่างไรก็ตามเราเตอร์ภายนอกไม่จำเป็นต้องทำแพ็กเก็ตฟิลเตอร์ริ่ง เพียงแค่ใช้เชื่อมต่อแพ็กเกจ (interface package) ซึ่งไม่ได้ทำแพ็กเก็ตฟิลเตอร์ริ่งดีนัก แต่มันก็ไม่ใช่ปัญหา

ในสถาปัตยกรรมนี้ แบสชั่นโฮสต์จะเปิดสู่อินเตอร์เน็ต ดังนั้นคุณจำเป็นจะต้องมีการดูแลเป็นพิเศษ

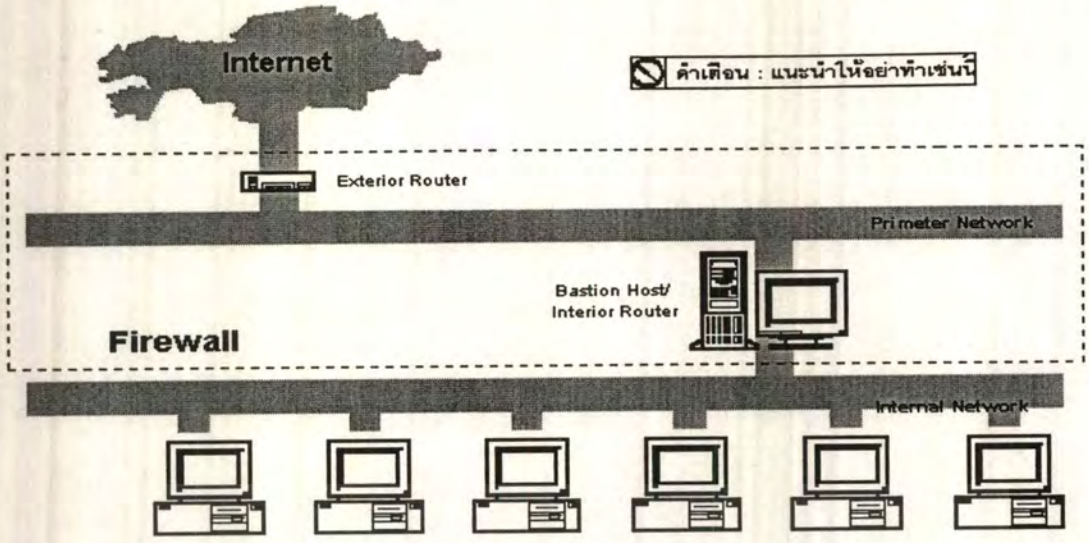


รูป 3.13 แสดงสถาปัตยกรรมที่ทำการรวมแบสชั่นโฮสต์และเราเตอร์ภายนอก

3.4.4 เป็นอันตรายในการรวมแบสชั่นโฮสต์และเราเตอร์ภายใน

เดิมในไฟร์วอลล์แบบสกรีนสับเน็ตมีเราเตอร์ภายในบนเครือข่ายเพอริมิเตอร์ สำหรับแบสชั่นโฮสต์ไม่มีกำหนดเส้นทางตายตัวในการเข้าสู่ภายใน เมื่อมีการบุกทะลวงแบสชั่นโฮสต์การสัญจรก็ยังถูกป้องกันจากการแอบดู เพราะต้องผ่านเราเตอร์ภายใน แต่หากรวมแบสชั่นโฮสต์กับเราเตอร์ภายในคุณจะได้เป็นไฟร์วอลล์แบบสกรีนโฮสต์และถ้าแบสชั่นโฮสต์ถูกบุกรุก จะไม่มีอะไรรักษาความปลอดภัยระหว่างแบสชั่นโฮสต์และเครือข่ายภายใน

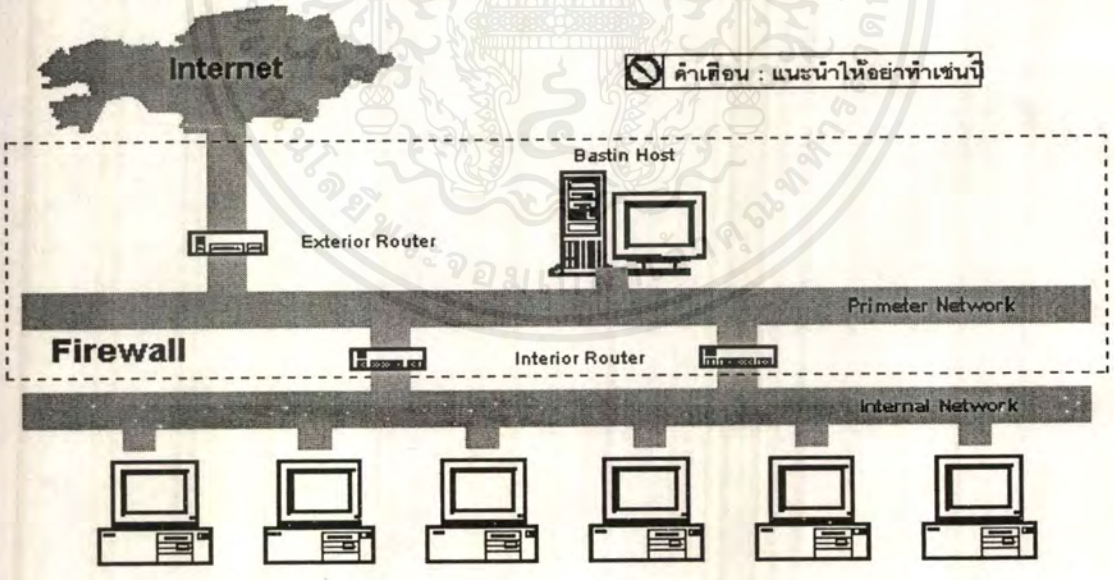
จุดประสงค์อย่างหนึ่งของเครือข่ายเพอริมิเตอร์ คือ ป้องกันแบสชั่นโฮสต์จากการแอบดูการสัญจรภายใน การรวมแบสชั่นโฮสต์และเราเตอร์ภายในจะทำให้การสัญจรภายในถูกมองเห็น



รูป 3.14 แสดงสถาปัตยกรรมที่ทำการรวมแบสชั่นโฮสต์และเราเตอร์ภายใน

3.4.5 เป็นอันตรายในการใช้เราเตอร์ภายในหลายตัว (mutiple interior routers)

ปัญหาพื้นฐานของซอฟท์แวร์เราติ้ง (routing software) บนระบบภายใน (tnternal system) คือ จะหาทางที่เร็วที่สุดที่จะหาระบบภายในอื่นผ่านเครือข่ายเพอริมิเตอร์ ถ้าโชคดีการเข้าถึงจะไม่ทำงาน เพราะมันถูกขัดขวางโดยแพ็กเก็ตฟิลเตอร์ริ่งบนเราเตอร์ตัวหนึ่ง แต่ถ้าโชคไม่ดีมันทำงาน การสัญจรจะผ่านเครือข่ายเพอริมิเตอร์ ซึ่งมันอาจจะถูกแอบดูได้ถ้าแบสชั่นโฮสต์ถูกบุกรุก

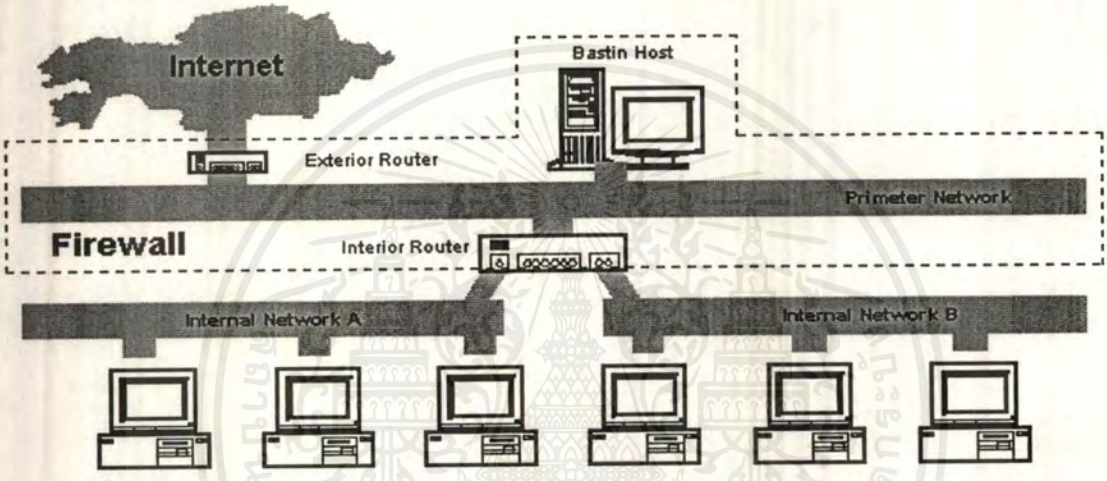


รูป 3.15 แสดงการใช้เราเตอร์ภายในหลายตัว (multiple interior router)

เป็นการยากที่จะกำหนดคอนฟิกมัลติเพิลอินทีเรีย (multiple interior configuration) ได้ถูกต้อง เพราะเราเตอร์ภายในหนึ่งตัวนั้นสำคัญมาก และมีกลุ่มของแพ็กเก็ตฟิลเตอร์ริ่งที่ซับซ้อนมาก ยิ่งมีเราเตอร์ภายในสองตัวก็จะยิ่งมีโอกาสที่จะกำหนดเกณฑ์ผิดได้ง่ายขึ้น

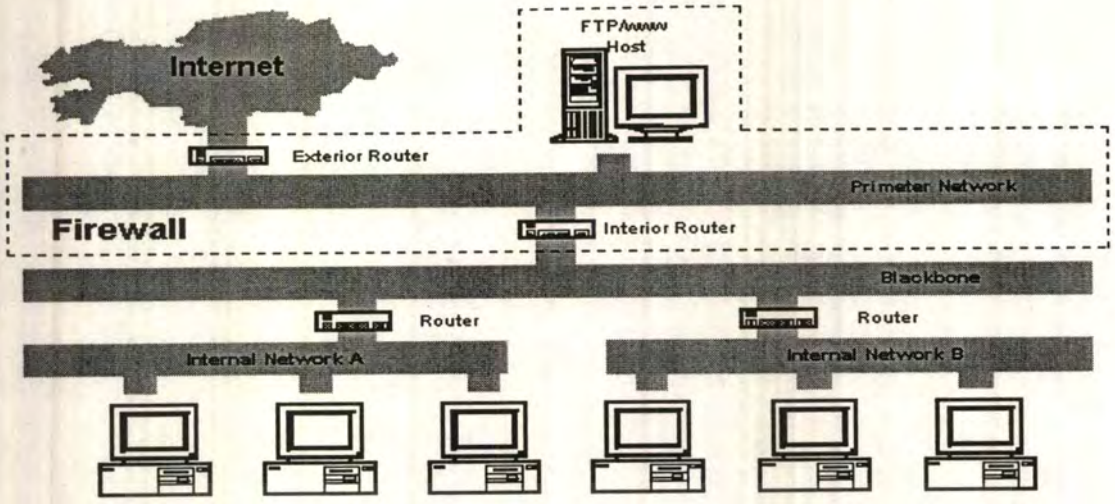
กรณีแรก ถ้าคุณกำหนดคอนฟิกผิดพลาดบางสิ่ง จากเครือข่ายเพอริมิเตอร์จะมีบางครั้งที่สามารถผ่านไปยังโลกภายนอกโดยที่ไม่ได้กำหนด ในกรณีที่สอง คุณควรจะที่จะพิจารณาการยกระดับเราเตอร์ภายในให้สอดคล้องกับเราเตอร์ภายนอกแทนที่จะเพิ่มไปอีกหนึ่งตัวอย่างรอบคอบ

เหตุผลอื่นที่จะใช้เราเตอร์ภายในหลายตัว คือ คุณมีหลาย ๆ เครือข่าย ซึ่งมีเทคนิค , องค์กร หรือ เหตุผลทางการที่จะทำให้ไม่สามารถแบ่งปันเราเตอร์ได้ ทางที่ง่าย ๆ คือ ทำตามรูปที่ 3.16 มันจะทำให้สับสนในการกำหนดคอนฟิกแต่ไม่ได้เพิ่มโอกาสความเสี่ยงที่เกิดจากการใช้เราเตอร์ภายในหลายตัว ถ้ามีหลายเครือข่ายเกินกว่าที่จะแบ่งปันเราเตอร์ตัวเดียว หรือถ้าการแบ่งปันเราเตอร์ไม่เป็นที่ยอมรับกับเหตุผลอื่น ๆ ที่พิจารณาการสร้างแบ็คโบนภายใน (internal backbone) และติดต่อกับมันถึงเครือข่ายเพอริมิเตอร์ด้วยเราเตอร์ตัวเดียว ดังรูปที่ 3.17



รูป 3.16 แสดงเครือข่ายภายในหลายเครือข่าย (โดยการแบ่งการติดต่อไปที่เราเตอร์ตัวเดียว)

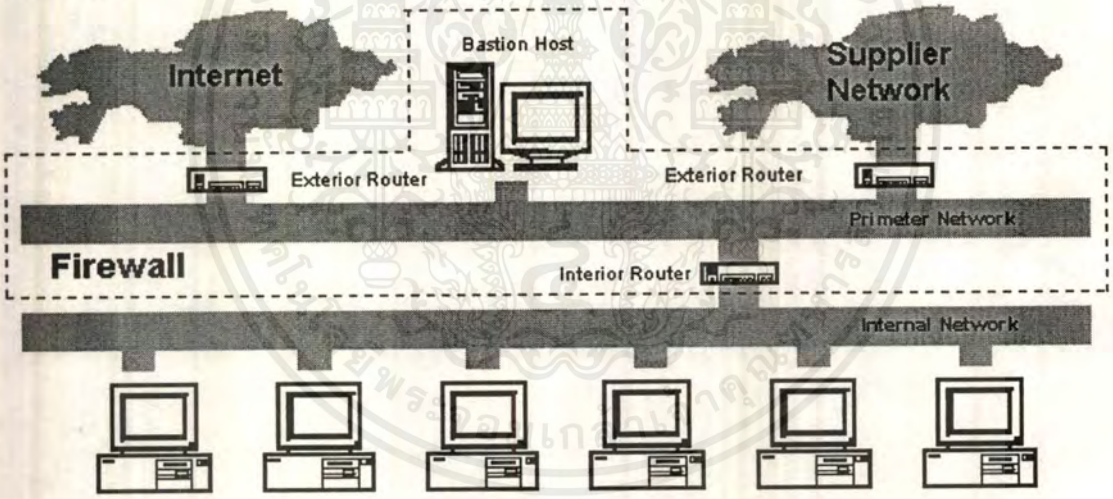
คุณอาจจะหาเส้นทางที่จะปรับความปลอดภัยที่ต่างกันท่ามกลางแต่ละเครือข่ายภายใน (internal network) ที่จะติดต่อถึงเครือข่ายเพอริมิเตอร์ (perimeter network) ผ่านเราเตอร์ (router) ที่ถูกแบ่ง กรณีนี้เครือข่ายเพอริมิเตอร์ควรจะมีการเชื่อมต่อภายใน (interconnection) ระหว่างเครือข่ายภายใน ซึ่งไม่ควรจะไว้ใจการสัญจร (traffic) ที่ผ่านระหว่างมันและแต่ละเครือข่ายภายในควรจะระวังความไม่น่าเชื่อจากภายนอกซึ่งมันจะไม่อำนวยความสะดวกกับผู้ใช้บางคนบนแต่ละเครือข่ายแต่ควรจะมีการยอมรับความปลอดภัยของแต่ละไซต์ที่แตกต่างกัน



รูป 3.17 แสดงเครือข่ายภายในหลายเครือข่าย (multiple internal networks : blackbone architecture)

3.4.6 สามารถใช้ เราเตอร์ภายนอกได้หลายตัว (multiple exterior routers)

บางกรณีสามารถติดต่อเราเตอร์ภายนอกหลายตัวถึงเครือข่ายเพอริมิเตอร์เดียวกัน ดังรูปที่ 3.18 ตัวอย่างเช่น



รูป 3.18 แสดงสถาปัตยกรรมที่ใช้เราเตอร์ภายนอกหลายตัว (multiple exterior routers)

- คุณมีหลายการติดต่อถึงอินเตอร์เน็ต เช่น ผ่านหลายๆผู้จัดหาบริการ (service providers) สำหรับการมีเกินความจำเป็น (redundancy)
- คุณมีการติดต่อกับอินเตอร์เน็ตและติดต่ออย่างอื่นกับไซต์อื่นๆ

ในกรณีนี้คุณอาจจะแทนที่เราเตอร์ภายนอกหนึ่งตัว ด้วยเราเตอร์ภายนอกหลายตัว

ในการติดต่อเราเตอร์ภายนอกหลายตัวซึ่งไปที่เครือข่ายภายนอกเดียวกัน เช่น ผู้จัดหาบริการอินเตอร์เน็ต 2 ราย (2 Internet provider) ไม่ใช่ปัญหาใหญ่ เขาอาจจะมีฟิลเตอร์เซ็ต (filter set) ที่ต่างกัน

แต่ไม่วิกฤตกับเราเตอร์ภายนอกซึ่งมีโอกาสที่ด้านหนึ่งยอมรับ แต่การยอมรับของเราเตอร์ภายนอกไม่น่ากลัวเท่าใดนัก

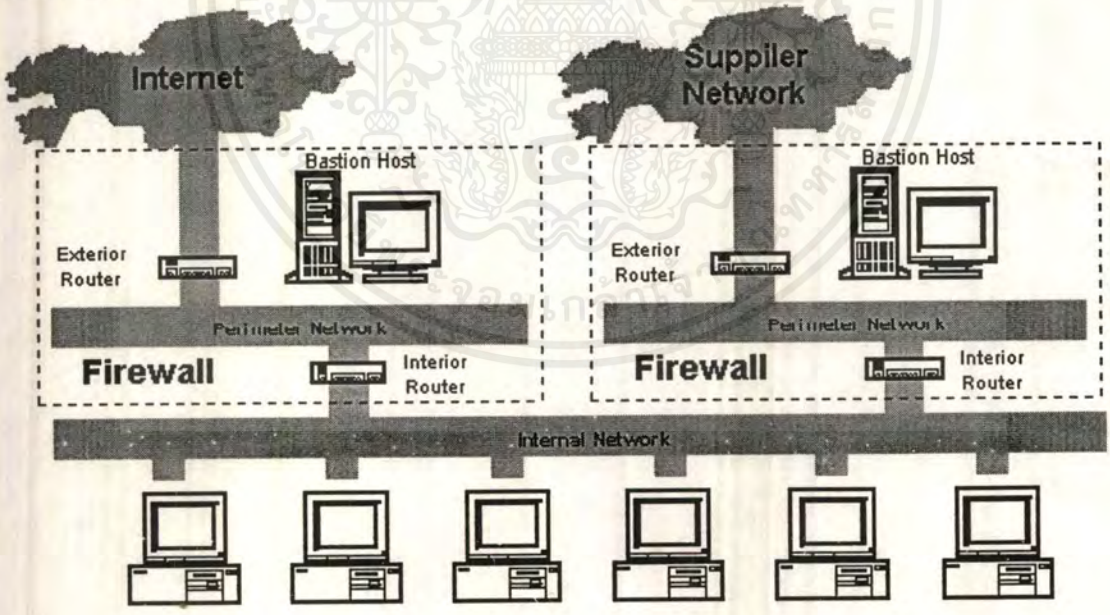
สิ่งที่ซับซ้อนถ้าติดต่อกับสถานที่ที่ต่างกัน เช่น หนึ่งติดต่อกับอินเตอร์เน็ตอีกหนึ่งติดต่อกับไซต์ที่คุณร่วมมือหรือเป็นไส้ศึกหรือต้องการช่วงการติดต่อที่กว้าง ควรคำนึงถึงการสัญจรใดที่คนเห็นเมื่อเขาสามารถบุกรุกบนเครือข่ายเพอริมิเตอร์ ตัวอย่างเช่น ถ้าผู้บุกรุกเข้ามา เขาจะสามารถแอบดูการสัญจรระหว่างไซต์ของคุณและส่วนเสริมหรือส่วนที่เป็นสมาชิกได้ไหม ถ้าเป็นเช่นนั้นแล้วคุณอาจจะต้องการที่จะติดตั้งเครือข่ายเพอริมิเตอร์หลาย ๆ เครือข่ายแทนที่จะใช้หลาย ๆ เราเตอร์บนเครือข่ายเพอริมิเตอร์เดียว

3.4.7 สามารถใช้หลายเครือข่ายเพอริมิเตอร์ (multiple perimeter networks)

รูปที่ 3.19 ต่อไปนี้แสดงคอนฟิกชนิดนี้

คุณอาจจะวางหลาย ๆ เครือข่ายเพอริมิเตอร์สำหรับความเป็นส่วนตัว ซึ่งคุณสามารถให้ข้อมูลที่คุณเชื่อมั่นผ่านเส้นทางหนึ่ง และการติดต่อกับอินเตอร์เน็ตอีกเส้นทางหนึ่ง ในกรณีนี้ คุณอาจจะติดต่อเครือข่ายเพอริมิเตอร์ทั้งสองด้านด้วยเราเตอร์ภายในเดียวกัน

การใช้หลาย ๆ เครือข่ายเพอริมิเตอร์มีความเสี่ยงน้อยกว่าการใช้เราเตอร์ภายในหลายตัวที่แบ่งปันเครือข่ายภายในเดียวกันแต่มันจะทำให้ปวดหัว คุณควรจะมีเราเตอร์ภายในหลายตัวแสดงหลายๆจุด (multiple point) ที่เป็นไปได้และยอมรับ ซึ่งเราเตอร์จะต้องเฝ้าดูอย่างระวังทำให้เหมาะกับระดับความปลอดภัยที่ตั้งไว้ ถ้าทั้งคู่ติดต่อกับอินเตอร์เน็ตก็ต้องการทำให้เป็นนโยบายเดียวกัน



รูป 3.19 แสดงสถาปัตยกรรมที่ใช้หลายเครือข่ายเพอริมิเตอร์ (multiple firewalls)

3.4.8 สามารถใช้ดูอัลโฮมโฮสต์ (Dual-homed hosts) และสกรีนสับเน็ต (Screened subnets)

เราสามารถเพิ่มระดับความปลอดภัยโดย รวมดูอัลโฮมโฮสต์กับสกรีนสับเน็ต ซึ่งในการทำจะต้องแยกเครือข่ายเพอริมิเตอร์และแทรกดูอัลโฮมโฮสต์ เราเตอร์จะทำการป้องกันการปลอมแปลงและป้องกันการผิดพลาดที่ดูอัลโฮมโฮสต์เริ่มจัดหาเส้นทางให้กับการสัญจร ดูอัลโฮมโฮสต์สามารถจัดการควบคุมได้ละเอียดดีกว่าแพ็กเก็ตฟิลเตอร์ริ่ง ซึ่งก็คือ จัดการป้องกันหลายชั้นขึ้น (belt-and-suspenders)

3.5 ไฟร์วอลล์ภายใน (internal firewalls)

เราอาจจะป้องกันเครือข่ายภายในจากส่วนอื่นๆซึ่งอาจจะมีเหตุผลที่ต้องกระทำดังนี้

- คุณมีการทดลองหรือทดสอบเครือข่ายกับสิ่งแปลกปลอมที่จะเข้ามา
- คุณมีเครือข่ายที่ปลอดภัยน้อยกว่าไซต์ที่คุณอยู่
- คุณมีเครือข่ายที่ปลอดภัยมากกว่าไซต์ที่คุณอยู่ เช่น โครงงานลับที่กำลังพัฒนา หรือข้อมูลที่เกี่ยวกับการเงินหรือเกรดที่สอบผ่าน

จากเหตุผลข้างบนจึงสร้างไฟร์วอลล์ภายใน (internal firewalls) ซึ่งไฟร์วอลล์นี้จะตั้งระหว่างสองส่วนขององค์กรเดียวกันหรือระหว่างสององค์กรที่คุณแบ่งปันเครือข่ายมากกว่าระหว่างหนึ่งองค์กรกับอินเตอร์เน็ต

มันมักจะเป็นการเก็บส่วนหนึ่งขององค์กรของคุณจากส่วนอื่นๆ แต่ไม่ใช่ทุกคนในองค์กรต้องการการบริการเดียวกัน และความปลอดภัยที่ใช้บ่อยๆ และสำคัญในบางส่วนขององค์กรมากกว่าอื่นๆ

เครื่องมือและเทคนิคที่คุณใช้สร้างอินเตอร์เน็ตไฟร์วอลล์ (Internet firewall) มีประโยชน์ในการสร้าง ไฟร์วอลล์ภายในอย่างไรก็ตามมีจุดพิจารณาพิเศษที่คุณต้องเก็บไว้ในใจ ถ้าคุณจะสร้างไฟร์วอลล์ภายใน

บทที่ 4
แบสชั่นโฮสต์ (bastion host)

แบสชั่นโฮสต์ (bastion host) แสดงตัวเปิดเผยบนอินเตอร์เน็ตซึ่งเปรียบเสมือนกับล็อบบี้ (lobby) โรงแรมที่ให้คนภายนอกผ่านเข้ามาก่อนจึงจะเข้าภายในได้ ซึ่งแบสชั่นโฮสต์ก็เป็นระบบไฟร์วอลล์ (firewall)ที่คนภายนอกจะต้องผ่านเข้ามาก่อนจึงสามารถใช้บริการในไฟร์วอลล์ได้

ในการออกแบบเนื่องจากแบสชั่นโฮสต์เป็นโฮสต์ที่เปิดเผยดังนั้นไฟร์วอลล์ชนิดนี้จึงจะต้องมีการรักษาความปลอดภัยอย่างรัดกุม

4.1 หลักการทั่วไป

มีสองหลักการ สำหรับการออกแบบและสร้างแบสชั่นโฮสต์ดังต่อไปนี้

- ทำให้ง่ายที่สุด (Keep is simple)
  ควรจะให้บริการที่มีสิทธิ์เข้ามาได้น้อยที่สุด เพราะซอฟท์แวร์อาจจะมีข้อผิดพลาด (bug) ซึ่งอาจจะทำให้การรักษาความปลอดภัยบกพร่องได้

- เตรียมประนีประนอม (Prepared for bastion host to be compromised)
  ควรจะต้องมีการคาดการไว้ล่วงหน้าว่าจะมีอะไรเกิดขึ้น และหากมีการผิดพลาดหรือมีการบุกรุกเข้ามาจะทำการป้องกันอย่างไร เช่น การสร้างการป้องกันที่เครือข่ายภายใน (รหัสผ่าน (password) , อุปกรณ์ป้องกันบางอย่าง) หรือใส่แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) ระหว่างแบสชั่นโฮสต์และโฮสต์ภายใน (internal host)

4.2 แบสชั่นโฮสต์ (bastion host) ชนิดพิเศษ

4.2.1 นอนเราติ้งดูอัลโฮมโฮสต์ (nonrouting dual-homed hosts)

มีการติดต่อหลาย ๆ เครือข่ายแต่จะไม่มีการสื่อสารที่ผ่านตัวมัน ซึ่งโฮสต์จะเป็นไฟร์วอลล์เองหรืออาจเป็นส่วนที่ซับซ้อนของไฟร์วอลล์ โดยส่วนมากจะถูกคอนฟิกเหมือนแบสชั่นโฮสต์ทั่วไป แต่ต้องการการระมัดระวังและไม่มีการทำเราติ้ง (routing)

4.2.2 วิกทิมแมชชีน (victim machines)

ถ้าต้องการการทำงานบริการที่จะไม่รู้จะรักษาความปลอดภัยอย่างไรมักจะใช้วิธีนี้ คือ วิกทิมแมชชีน (victim machines) โดยจะเป็นเครื่องที่เราไม่สนใจอะไรข้างใน แต่เราจะทำการคอนฟิกเหมือนกับแบสชั่นโฮสต์แต่ไม่ให้ผู้ใช้ล็อกอิน (login) เข้าไปเพื่อที่จะให้ผู้บุกรุกเข้าไปแทน

4.2.3 แบสชั่นโฮสต์ภายใน (internal bastion host)

จะมีแบสชั่นโฮสต์อยู่ภายในอีกทีหนึ่ง ซึ่งจะมีปฏิกิริยากับแบสชั่นโฮสต์ภายนอก คือ เมื่อข้อมูลเปลี่ยนไปมันก็จะเปลี่ยนตาม แบสชั่นโฮสต์ภายในมักจะมีการคอนฟิกและป้องกันที่เข้มแข็งมากกว่าโฮสต์อื่นๆ ทั่วไป

4.3 การวางตำแหน่งแบสชั่นโฮสต์บนเครือข่าย

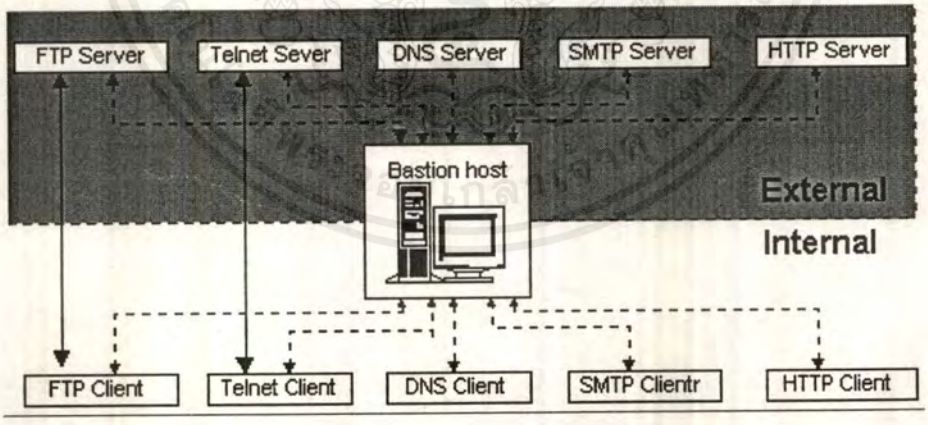
ในเครือข่ายเราควรจะจับทุกๆแพ็กเก็ตบนเครือข่ายซึ่งการจับเราจะมีโปรแกรมที่จะจับ แต่โปรแกรมเหล่านี้ผู้บุกรุกก็สามารถใช้ได้ แล้วลักลอบเข้ามาในเครือข่ายซึ่งอาจจะเป็นเทลเน็ต (TELNET) ,เอฟทีพี (FTP) , หรืออื่น ๆ แล้วเขาก็จะเข้ามาอยู่บนแบสชั่นโฮสต์

ทางที่จะแก้ไขปัญหานี้ คือ การนำเอาแบสชั่นโฮสต์ไปอยู่ที่ที่ไม่ใช่เครือข่ายภายในซึ่งมีเครือข่ายเพอริมิเตอร์เป็นชั้นแบ่งระหว่างเครือข่ายภายในกับอินเตอร์เน็ต อันจะถูกแบ่งโดยเราเตอร์ (router)หรือบริดจ์ (bridge) ซึ่งจะทำให้ไม่เห็นการขนส่งของกันและกันได้

การใช้แพ็กเก็ตฟิลเตอร์ริ่งระหว่างเครือข่ายเพอริมิเตอร์และเครือข่ายภายในช่วยในการจำกัดส่วนที่จะเปิดเผยและลดจำนวนโฮสต์และบริการที่แบสชั่นโฮสต์จะเข้าไปถึง

4.4 การเลือกบริการ (service) ที่จะจัดการโดยแบสชั่นโฮสต์ (bastion host)

แบสชั่นโฮสต์จัดการทุกๆบริการบนไซต์ที่ต้องการเข้าถึงผ่านบริการอินเตอร์เน็ต (nternet service) ส่วนบริการที่ไม่ต้องการความปลอดภัยมากนักก็จะวิ่งผ่านแพ็กเก็ตฟิลเตอร์ริ่ง



รูป 4.1 แสดงแบสชั่นโฮสต์ที่ทำงาน บนหลายๆบริการอินเตอร์เน็ตที่ต่างกัน

สามารถแบ่งบริการ (service) ออกเป็น 4 กลุ่มดังนี้

1. **บริการที่ความปลอดภัยอยู่แล้ว** บริการชนิดนี้ถูกกระทำผ่านแพ็กเก็ตฟิลเตอร์ริ่ง
2. **บริการที่ไม่ปลอดภัย** แต่สามารถที่จะทำให้ปลอดภัยได้ บริการชนิดนี้จะกระทำกับแบสชั่นโฮสต์

3. **บริการที่ไม่ปลอดภัย และไม่สามารถที่จะทำให้ปลอดภัยได้** บริการชนิดนี้จะต้องทำให้ใช้ไม่ได้หรือกระทำบนโฮสต์ที่เป็นเหยื่อ (victim host)
4. **บริการที่ห้ามใช้หรือไม่สามารถใช้เชื่อมกับอินเตอร์เน็ต** บริการจะต้องทำให้ใช้ไม่ได้

4.5 ไม่อนุญาตให้มีบัญชีผู้ใช้ (user account) บนแบสชั่นโฮสต์ (bastion host)

การจะทำให้ระบบมีความปลอดภัย (security) ดีที่สุดไม่ควรจะให้มีบัญชีผู้ใช้ (user account) เนื่องจากเหตุผลดังนี้

- ความไม่มั่นคงของบัญชีผู้ใช้
- ความไม่มั่นคงของการบริการที่ต้องรองรับกับบัญชีผู้ใช้
- เสถียรภาพและความไว้วางใจของเครื่อง
- การทำลายความปลอดภัยโดยบังเอิญจากผู้ใช้
- เพิ่มความยากในการที่จะค้นหาในการบุกรุก

4.6 การสร้างแบสชั่นโฮสต์

ในการสร้างแบสชั่นโฮสต์มีขั้นตอนดังต่อไปนี้

4.6.1 ความปลอดภัยของกลไก (secure the machines)

ในการเริ่มต้นสร้างให้เริ่มจากขบวนการของระบบ (algorithm system) ที่สะอาดซึ่งมีขั้นตอนย่อยๆ ดังต่อไปนี้

1. ติดตั้งระบบปฏิบัติการ (operating system) ที่สะอาดให้เล็กที่สุด

   ควรเลือกระบบปฏิบัติการ (operating system) ที่ง่ายต่อการทำงานในอนาคต และพยายามติดตั้ง เฉพาะส่วนที่ใช้เท่านั้น

2. ซ่อมข้อบกพร่อง (bug) ที่เราทราบ

   โดยการหาหนทาง (path) หรือคำปรึกษาเกี่ยวกับระบบปฏิบัติการที่ใช้

3. ใช้เช็คลิสท์ (checklist)

   ในการตรวจสอบทุกๆ อย่างในระบบความปลอดภัย

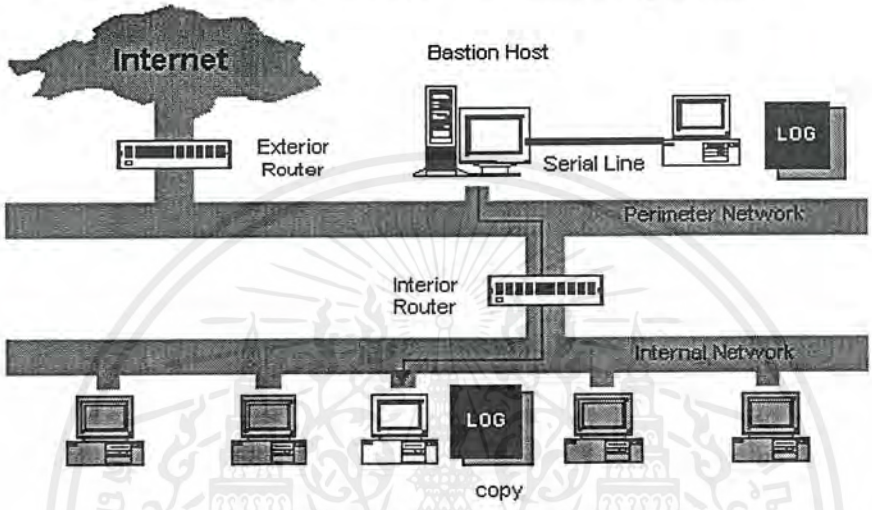
4. ป้องกันด้วยล็อกระบบ (system log)

แบสชั่นโฮสต์ต้องมีการเก็บการล็อกกิ้ง (logging) ซึ่งมีเหตุผลสำคัญสองอย่าง

- ช่วยบอกว่าแบสชั่นโฮสต์ได้ทำอะไรไปบ้าง ตัดสินใจอะไรไปบ้าง
- ช่วยดูว่ามีใครบ้างที่บุกรุกเข้ามา และเกิดอะไรขึ้นบ้าง

ที่ที่จะเกิดล็อกระบบ (system log) ควรจะมีสำเนาสองชุด ชุดหนึ่งอยู่ในที่ที่ระบุเพื่อให้ง่ายในการดู ส่วนอีกชุดหนึ่งเพื่อป้องกันการเสียหาย

- **ล็อกระบบ (system log) สำหรับความสะดวก** สามารถเก็บไว้ในแบสชั่นโฮสต์หรือในเครือข่ายภายในได้ ข้อดีของการเก็บในแบสชั่นโฮสต์ คือ ไม่ต้องจัดตั้งเกี่ยวกับระบบที่ต่างกัน และไม่ต้องคอนฟิกในแพ็กเก็ตฟิลเตอร์ริ่ง ส่วนข้อดีของการเก็บในโฮสต์ภายใน คือ สะดวกในการเข้าถึงเพราะไม่ต้องเข้าไปในแบสชั่นโฮสต์
- **ล็อกระบบ (system log) เพื่อป้องกันการเสียหาย** ทำได้โดยการต่อสายไปที่ฟิลเตอร์ แต่มีปัญหา คือ กระดาษอาจจะหมดหรือติด และยากในการค้นหาสิ่งที่ต้องการค้น ส่วนอีกวิธีหนึ่งก็คือต่อพอร์ตอนุกรมไปที่เครื่องพีซี ซึ่งจะทำให้ไม่มีปัญหาข้างต้น



รูป 4.2 แสดงการติดต่อพีซีโดยใช้สายอนุกรม

4.6.2 ไม่อนุญาตุสำหรับบริการที่ไม่ต้องการ (disabling nonrequired service)

ทำการตัดบริการที่ไม่ต้องการเนื่องจากบริการที่ทำงานบนแบสชั่นโฮสต์อาจจะมีข้อบกพร่องหรือมีปัญหาได้จึงต้องมีการป้องกันโดยการตัดบริการที่ไม่ใช่ หรือไม่สำคัญออกไป

4.6.3 การติดตั้งและปรับปรุงเปลี่ยนแปลงบริการ (modifies service)

วางบริการที่ไม่ได้จัดการโดยระบบปฏิบัติการ (operating system) ที่ใช้ จำเป็นต้องติดตั้งเข้ามา แต่บริการเหล่านี้ไม่ได้ถูกรักษาความปลอดภัยโดยระบบปฏิบัติการ ดังนั้นจึงควรที่จะมีตัวที่เข้ามาป้องกัน โดยที่ส่วนใหญ่นิยมใช้คือ

1. ทีซีพีแวรพเปอร์แพ็กเก็ต (TCP wrapper packet)
2. เน็ตาคัล (NETACL) เป็นส่วนประกอบของไฟร์วอลล์ทูลคิท (firewall toolkit)

4.6.4 ทำคอนฟิกใหม่สำหรับผลิตภัณฑ์ (reconfig for production)

เพื่อที่จะให้ได้คอนฟิกที่ดีที่สุดจะต้องทำดังนี้

- ทำคอนฟิกและสร้างสีใหม่
- โปรแกรมที่ไม่จำเป็น
- ทำให้แฟ้มของระบบเป็นแบบอ่านได้อย่างเดียว (readonly)

4.6.5 ดำเนินการตรวจสอบความปลอดภัย (running a security audit)

ทำการตรวจสอบระบบความปลอดภัยเพื่อทำให้แน่ใจว่าไม่ได้มองข้ามสิ่งใดไปขณะติดตั้งและตั้งเป็นหลักเกณฑ์พื้นฐานในการเปรียบเทียบเพื่อใช้ในการตรวจสอบต่อไป

4.6.6 ทำการติดต่อกับเครื่องจักร (machines)

ตอนนี้จะได้เครื่องจักรที่มีระบบความปลอดภัยเต็มตัวก็จะทำการติดต่อกับเครือข่ายปลายทาง

<u>หมายเหตุ</u> เมื่อทำการคอนฟิกแบสชั่นโฮสต์พร้อมแล้วในการทำการป้องกันควรจะทำสำรอง (back up) ระบบไว้ด้วย

บทที่ 5
แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering)

แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) เป็นเครือข่ายความปลอดภัย (network security) ที่ทำการควบคุมการไหลของข้อมูล เพราะในการส่งข้อมูลผ่านเครือข่ายข้อมูลนี้จะถูกแตกเป็นชิ้นเล็ก ๆ แต่ละชิ้นก็จะถูกแบ่งกันส่งไป ชิ้นเล็กเหล่านี้เรียกว่าแพ็กเก็ต (packet) ซึ่งจะถูกส่งในชั้นไอพี (IP layer)

อุปกรณ์ที่ติดต่อระดับไอพี เรียกว่า เราเตอร์ (router) โดยเราเตอร์นี้จะทำการตัดสินใจหาเส้นทางที่จะส่งแพ็กเก็ตไป เพราะโดยทั่วไปแพ็กเก็ตจะบอกเพียงจุดหมายที่จะไปแต่ไม่ได้บอกเส้นทางที่จะไป การหาเส้นทางนี้จะใช้เราติ้งโปรโตคอล (routing protocols) เช่น อาร์ไอพี (Routing Information Protocol : RIP) และโอเอสพีเอฟ (Open Shortest Path First : OSPF) ในการสร้างตารางเราติ้ง (routing table) เพื่อบอกว่า แพ็กเก็ตจะไปจุดหมายอย่างไร

5.1 ทำไมต้องใช้แพ็กเก็ตฟิลเตอร์ริ่ง (packer filtering)

แพ็กเก็ตฟิลเตอร์ริ่งทำการควบคุม (อนุญาตหรือไม่อนุญาต) การส่งข้อมูลบนฐานต่อไปนี้

- ที่อยู่ที่ข้อมูลมา
- ที่อยู่ที่ข้อมูลจะไป
- เซสชั่น (session) และแอพพลิเคชั่นโปรโตคอล (application protocol) ที่ถูกใช้ส่งข้อมูล

โดยแพ็กเก็ตฟิลเตอร์ริ่งสามารถกำหนดเงื่อนไขต่าง ๆ ได้ เช่น ไม่อนุญาตให้คนที่ใช้เทลเน็ต (TELNET) เป็นแอพพลิเคชั่นโปรโตคอล (application protocol) เข้ามาจากภายนอก หรือ ให้ทุกคนส่งอีเมล์ (E-Mail) ผ่านเอสเอ็มทีพี (SMTP) เป็นแอพพลิเคชั่นโปรโตคอล (Application Protocol) อีกตัวหนึ่ง แต่มันไม่สามารถกำหนดเงื่อนไขจากหลักฐาน (identify) เช่น ผู้ใช้คนนี้สามารถเทลเน็ต (TELNET) จากภายนอก แต่ผู้ใช้คนอื่นไม่สามารถทำได้ หรือ คุณสามารถส่งไฟล์นี้ได้แต่ไม่สามารถส่งไฟล์อื่นได้

ข้อได้เปรียบของแพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) คือ มันอนุญาตที่จะจัดการในจุด ๆ เดียวหรือในส่วนที่จะต้องการป้องกันโดยเฉพาะ พิจารณาบริการเทลเน็ต (TELNET service) ถ้าต้องการจะไม่อนุญาตให้เทลเน็ต โดยการปิดเซิร์ฟเวอร์ (server) ที่ให้บริการเทลเน็ตหากมีคนในองค์กรของคุณต้องการเพิ่มเครื่องใหม่ซึ่งต้องใช้เทลเน็ตเซิร์ฟเวอร์ (TELNET server) ในทางตรงกันข้าม หากทำการไม่อนุญาตโดยแพ็กเก็ตฟิลเตอร์ริ่งเราเตอร์ (packet filtering router) เมื่อจะทำการใส่เครื่องใหม่ก็ไม่ต้องห่วงเรื่องการปิดเปิดเครื่องเซิร์ฟเวอร์

เราเตอร์ควรจะวางบนตำแหน่งที่เรียกว่า จุดศูนย์กลางหรือศูนย์รวมการติดต่อ (choke point) สำหรับทุก ๆ การสัญจรที่เข้าหรือออกจากเครือข่าย

5.2 ข้อได้เปรียบของแพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering)

5.2.1 สกรีนนิ่งเราเตอร์ (screening router) สามารถป้องกันเครือข่ายทั้งหมด

ข้อได้เปรียบของแพ็กเก็ตฟิลเตอร์ริ่งข้อหนึ่ง คือ หากวางดี ๆ แล้วสามารถป้องกันได้ทั้งเครือข่าย แม้จะวางเพียงแค่ตัวเดียว

5.2.2 แพ็กเก็ตฟิลเตอร์ริ่งไม่ต้องการการกำหนดจากผู้ใช้

แพ็กเก็ตฟิลเตอร์ริ่งไม่เหมือนกับพร็อกซี (proxy) , แพ็กเก็ตฟิลเตอร์ริ่งไม่ต้องการซอฟท์แวร์ทั่วไป (custom software) หรือ การกำหนดของเครื่องไคลเอนต์ (client) หรือไม่ต้องการการปฏิบัติการสำหรับผู้ใช้เมื่อแพ็กเก็ตฟิลเตอร์ริ่งถูกออกแบบให้แพ็กเก็ตผ่านเราเตอร์ก็จะไม่แตกต่างจากเราเตอร์ทั่วไป

แพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering) ทำงานได้เสมือนโปร่งใส เนื่องจากไม่ต้องการความเกี่ยวข้องและรู้ตัวของผู้ใช้

5.2.3 แพ็กเก็ตฟิลเตอร์ริ่งเปิดกว้างที่จะมีเราเตอร์หลายๆ ตัว

แพ็กเก็ตฟิลเตอร์ริ่งสามารถมีอยู่ได้ในหลาย ๆ ฮาร์ดแวร์เราติ้งและซอฟท์แวร์เราติ้งซึ่งมีทั้งขายและให้ฟรี บนอินเตอร์เน็ตและไซต์ส่วนมากจะมีแพ็กเก็ตฟิลเตอร์ริ่งในเราเตอร์ที่เขาใช้

5.3 ข้อเสียเปรียบของแพ็กเก็ตฟิลเตอร์ริ่ง

5.3.1 เครื่องมือทำฟิลเตอร์ริ่ง (filtering tools) ไม่สมบูรณ์แบบ

แพ็กเก็ตฟิลเตอร์ริ่งสามารถใช้ได้บนหลายฮาร์ดแวร์และซอฟท์แวร์ แต่แพ็กเก็ตฟิลเตอร์ริ่งก็ยังไม่ใช้เครื่องมือที่สมบูรณ์แบบ นั้นคือ

- กฎของแพ็กเก็ตฟิลเตอร์ริ่งตั้งใจให้ยากในการคอนฟิก
- การคอนฟิกกฎของแพ็กเก็ตฟิลเตอร์ริ่งตั้งใจที่จะทำให้ยากในการทดสอบ
- การที่จะทำให้อยู่ในหลายผลิตภัณฑ์ทำได้ยาก
- แพ็กเก็ตฟิลเตอร์ริ่งอาจจะมีข้อบกพร่อง ซึ่งเมื่อแพ็กเก็ตฟิลเตอร์ริ่งปัญหาอาจจะทำให้แพ็กเก็ตผ่านเข้าไปได้ทั้งๆ ที่ควรจะเข้าไม่ได้

5.3.2 บางโปรโตคอลไม่เหมาะกับแพ็กเก็ตฟิลเตอร์ริ่ง

แม้ว่าจะทำให้แพ็กเก็ตฟิลเตอร์ริ่งสมบูรณ์แบบ ก็อาจจะพบว่าบางโปรโตคอลไม่เหมาะสมที่จะสร้างความปลอดภัยผ่านแพ็กเก็ตฟิลเตอร์ริ่ง

5.3.3 บางข้อกำหนดไม่สามารถทำได้โดยแพ็กเก็ตฟิลเตอร์ริ่งเราเตอร์

บางข้อกำหนดคุณไม่สามารถกำหนดได้ เช่น แพ็กเก็ต สามารถกล่าวได้ว่าแพ็กเก็ตมาจากโฮสต์ใด แต่ไม่รู้ว่าเป็นผู้ใช้คนใด นั่นคือ คุณไม่สามารถกำหนดเฉพาะเจาะจงผู้ใช้ คล้ายคลึงกับแพ็กเก็ตกล่าวได้ว่ามาจากพอร์ตใด แต่บอกไม่ได้ว่า มาจากแอพพลิเคชั่นใด

5.4 คอนฟิกแพ็กเก็ตฟิลเตอร์ริ่งเราเตอร์

การคอนฟิกแพ็กเก็ตฟิลเตอร์ริ่งเราเตอร์ ขั้นแรกต้องทราบว่าบริการใดที่จะอนุญาตบ้าง แล้วค่อยเปลี่ยนสิ่งที่คิดไว้เป็นกฎเกี่ยวกับแพ็กเก็ตสิ่งต่อไป นี้เป็นความคิด (concept) โดยทั่วไปเกี่ยวกับแพ็กเก็ต

5.4.1 โปรโตคอล ควรจะเป็นแบบสองทิศทาง (bidirectional)

โปรโตคอลควรจะเป็นแบบสองทิศทาง มันเกี่ยวเนื่องทั้งด้านที่ส่งคำสั่งและด้านที่ตอบรับ เมื่อทำการตั้งกฎให้แพ็กเก็ตฟิลเตอร์ริ่งต้องรู้ว่าแพ็กเก็ตไปทั้งสองทาง เช่น จะไม่ส่งคีย์ที่กดไปข้างนอก (เมื่อเทลเน็ต) และจะไม่รับแพ็กเก็ตกลับมาแต่รับเป็นหน้าจอกลับมาแทน

5.4.2 ระวังความแตกต่างระหว่างการเข้าและออก

จะต้องระวังเกี่ยวกับข้อแตกต่างระหว่างแพ็กเก็ตที่เข้าและออก และบริการ (service) ที่เข้าและออก เช่น เทลเน็ต (TELNET) ที่เป็นบริการออก แต่จะมีทั้งแพ็กเก็ตออก (ปุ่มกด) และแพ็กเก็ตเข้า (หน้าจอ)

5.4.3 ตั้งให้อนุญาตเป็นหลักหรือไม่อนุญาตเป็นหลัก

ในการตั้งกฎของแพ็กเก็ตฟิลเตอร์ริ่งอาจจะมีกฎบางข้อที่เราไม่กำหนด ซึ่งในกฎที่ไม่ได้ตั้งนี้สามารถตั้งให้มันอนุญาตหรือไม่อนุญาตอยู่ก่อนก็ได้ แต่ถ้าจะให้ปลอดภัยและมีประสิทธิภาพ ควรจะตั้งเป็นไม่อนุญาตมากกว่าที่จะตั้งเป็นอนุญาต

5.5 อะไรที่เราเตอร์ทำกับแพ็กเก็ต

- ให้แพ็กเก็ตผ่านไป เมื่อแพ็กเก็ตผ่านข้อกำหนดของแพ็กเก็ตฟิลเตอร์ริ่ง เราเตอร์ก็จะส่งแพ็กเก็ตต่อไปเหมือนกับเราเตอร์ธรรมดา
- หยุดแพ็กเก็ตเมื่อไม่ผ่านข้อกำหนด

5.5.1 เก็บการกระทำ

ถึงแม้ว่าแพ็กเก็ตจะถูกส่งต่อไปหรือหยุด (permitted or denied) เราเตอร์ก็จะต้องเก็บการกระทำที่เกิดขึ้น โดยเฉพาะเมื่อเกิดการหยุด ในบางกรณีคุณต้องการทราบว่าเหตุใดจึงไม่อนุญาต

ในการเก็บไม่จำเป็นต้องเก็บทุกๆแพ็กเก็ตแต่ควรจะเก็บเพียงบางส่วนของแพ็กเก็ต เช่น คุณต้องการที่จะเก็บจุดเริ่มของการติดต่อ (start-of-connection) ทีซีพีแพ็กเก็ต (TCP packet) ซึ่งบอกรายละเอียดเกี่ยวกับการเข้าออก

5.6 การส่งไอซีเอ็มพี (ICMP) ซึ่งเป็นโค้ดที่ผิดพลาด (error code) กลับมา

ถ้าแพ็กเก็ตถูกหยุด เราเตอร์ควรจะส่งโค้ดที่ผิดพลาด (ICMP error code) กลับมา เพื่อที่จะเตือนให้ไม่ต้องส่งแพ็กเก็ตกลับมาอีก ซึ่งจะทำให้ช่วยประหยัดการสัญจรและ เวลา

มีสองกลุ่มของโค้ดไอซีเอ็มพี (ICMP code) ที่จะกลับมา :

- โค้ดที่ไม่สามารถถึงจุดหมายปลายทาง (destination unreachable) ได้แก่ โค้ดที่ไม่สามารถถึงโฮสต์ (host unreachable) และ โค้ดที่ไม่สามารถถึงเครือข่าย (network uinreachable) ซึ่งบอกว่าจุดปลายทางมีปัญหา หรือ มีบางสิ่งบนเส้นทางที่จะไปหาโฮสต์เสียหาย โดยโค้ดนี้เกิดขึ้นก่อนถึงไฟร์วอลล์และแพ็กเก็ตฟิลเตอร์ริ่ง
- โค้ดที่ถูกบริหารเมื่อไม่สามารถถึงจุดหมายปลายทาง (destination administratively unreachable) ได้แก่ โค้ดที่ถูกบริหารเมื่อไม่สามารถถึงโฮสต์ (host administrative unreachable) และ โค้ดที่ถูกบริหารเมื่อไม่สามารถถึงเครือข่าย (network administratively unreachable) ซึ่งเป็นการบอกว่ามีการแพ็กเก็ตฟิลเตอร์ริ่งเมื่อมันมีการหยุดแพ็กเก็ต (drop packet)

5.7 แบบแผนสำหรับกฎของแพ็กเก็ตฟิลเตอร์ริ่ง (packet filtering rules)

ไวยากรณ์ที่ใช้ในการฟิลเตอร์เพื่อเปรียบเทียบ โดยมีบิตหลังสแลช (/) เป็นตัวบอกว่าจะเปรียบเทียบบิตใด เช่น 10.0.0.0/8 ตรงกับตำแหน่ง (address) ที่เริ่มต้นด้วยสิบ

ตัวอย่างของการตั้งให้อนุญาตให้การสัญจรไอพี (IP traffic) ระหว่างโฮสต์ภายนอกที่เชื่อถือได้ (trust external host : โฮสต์ 172.16.51.50) และ โฮสต์บนเครือข่ายภายใน (class C net 192.168.10.0) เราสามารถแสดงได้ดังนี้

Rule	Direction	Source Address	Destination Address	ACK set	Action
A	Inbound	Trusted external host	Internal	Any	Permit
B	Outbound	Internal	Trusted external host	Any	Permit
C	Either	Any	Any	Any	Deny

ตาราง 5.1 ตัวอย่างของการตั้งให้อนุญาตให้การสัญจรไอพี (IP traffic) ระหว่างโฮสต์ภายนอกที่เชื่อถือได้ และ โฮสต์บนเครือข่ายภายใน

กับการสกรีน (screened) คุณจะมีลักษณะเฉพาะ ดังนี้

between host 172.16.51.50 and net 192.168.10 accept ;

between host any and host any reject ;

สำหรับเทเลบิตเน็ตบลาสเซอร์ (telebit netblazer) สามารถตั้งกฎได้ดังนี้ (การติดต่อกับภายนอกเชื่อมต่อด้วย " syn0 ")

permit 172.16.51.50 / 32 192.168.10 / 24 syn0 in

deny 0.0.0.0 / 0 0.0.0.0 / 0 syn0 in

permit 192.168.10 / 24 172.16.51.50 / 32 syn0 out

deny 0.0.0.0 / 0 0.0.0.0 / 0 syn0 out

บนลีฟวิ่งสโตนพอร์ตมาสเตอร์ (livingston portMaster) หรือ ไออาร์เอกส์ (IRX) สามารถตั้งกฎได้ดังนี้ (การติดต่อกับภายนอกเชื่อมต่อด้วย "s1")

add filter s1.in

set filter s1.in 1 permit 172.16.51.50 / 32 192.168.10.0 / 24

set filter s1.in 2 deny 0.0.0.0 / 0 0.0.0.0 / 0

set s1 ifilter s1.in

add filter s1.out

set filter s1.out 1 permit 192.168.10.0 / 24 172.16.51.50 / 32

set filter s1.out 2 deny 0.0.0.0 / 0 0.0.0.0 / 0

set s1 ofilter s1.out

บนซิสโคเราเตอร์ (cisco router) สามารถตั้งกฎได้ดังนี้ (การติดต่อกับภายนอกเชื่อมต่อด้วย "serial1")

access-list 101 permit ip 172.16.51.50 0.0.0.0 192.168.10.0 0.0.0.255

access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

interface serial 0

access-group 101 in

access-list 102 permit ip 192.168.10.0 0.0.0.255 172.16.51.50 0.0.0.0

access-list 102 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

interface serial 0

access-group 102 in

สำหรับรายละเอียดของแต่ละไวยากรณ์ของแต่ละผลิตภัณฑ์ ถ้าหากเข้าใจไวยากรณ์ที่จะใช้ในตารางก็ไม่ใช่เรื่องยากที่จะแปลงจากตารางมาใช้กับระบบ

5.8 ฟิลเตอร์ริ่งโดยใช้ตำแหน่ง (filtering by address)

โดยพื้นฐานที่สุดคือการแพ็กเก็ตฟิลเตอร์ริ่งโดยใช้ตำแหน่ง (address) การฟิลเตอร์ริ่งนี้ทำโดยอนุญาตเฉพาะบางจุดตั้งต้นและจุดปลายทางของแพ็กเก็ตที่แน่นอน และป้องกันผู้บุกรุกที่ทำการปลอมแพ็กเก็ตเข้ามาใน เครือข่าย

ตัวอย่าง การป้องกันการเข้ามาของแพ็กเก็ตที่ปลอมตำแหน่งตั้งต้น (source address) ซึ่งมีลักษณะดังนี้

Rule	Direction	Source Address	Destination Address	Action
A	Inbound	Internal	Any	Deny

ตาราง 5.2 การป้องกันการเข้ามาของแพ็กเก็ตที่ปลอมตำแหน่งตั้งต้น (source address)

5.9 ความเสี่ยงของการทำฟิลเตอร์ริ่งโดยใช้ตำแหน่งตั้งต้น (source address)

มันไม่ปลอดภัยแน่นอนที่จะเชื่อตำแหน่งตั้งต้น (aource address) เพราะตำแหน่งตั้งต้นสามารถที่จะปลอมแปลงได้ สิ่งที่ผู้บุกรุกใช้ปลอมแปลงเข้ามามีสองประเภท คือ ตำแหน่งตั้งต้น (aource address) และ คนกลาง (man in the middle)

การบุกรุกแบบตำแหน่งตั้งต้น (source address) ผู้บุกรุกจะส่งแพ็กเก็ตเขามาในชื่อของโฮสต์ที่เราไว้วางใจ และหวังว่าเราจะมีปฏิกิริยาตอบกลับมา แต่ไม่ได้ต้องการที่จะได้รับแพ็กเก็ตกลับมา ในกรณีที่ผู้บุกรุกสนใจในการที่จะได้รับแพ็กเก็ตกลับมา เขาก็จะไม่อยู่บนเส้นทางระหว่างเราและเครื่องที่เขาแอบอ้าง (real machine)

ผลตอบของเราจะไปที่เครื่องที่เขาแอบอ้าง (real machine) แต่ไม่ได้ไปถึงยังผู้บุกรุก ถ้าผู้บุกรุกสามารถทำนายผลตอบสนองของเราเขาก็ไม่จำเป็นต้องรู้ว่าเราโต้ตอบอะไรกลับมา ในโปรโตคอลส่วนใหญ่ผู้บุกรุกสามารถทำนายได้ว่าผลตอบสนองจะเป็นเช่นไร เช่น ผู้บุกรุกเข้าไปในระบบของคุณและให้ส่งอีเมล์ (E-Mail) ที่เป็นไฟล์รหัสผ่าน (password file) มาให้เขา ถ้าระบบจะทำการส่งไฟล์ในรูปของเมล์มา ผู้บุกรุกก็ไม่รอดูการตอบสนองนี้

ในหลาย ๆ สถานการณ์ โดยเฉพาะการเชื่อมต่อแบบทีซีพี (TCP) เครื่องที่ผู้บุกรุกแอบอ้างชื่อจะมีปฏิกิริยาไปที่แพ็กเก็ตของคุณโดยพยายามตัดการติดต่อที่มีการปลอมแปลง ผู้บุกรุกไม่ต้องการให้สิ่งนี้เกิดขึ้น ดังนั้นผู้บุกรุกจึงต้องทำให้เสร็จก่อนที่เครื่องที่เขาแอบอ้าง (real machine) จะได้รับแพ็กเก็ต

ตอบกลับมา หรือก่อนที่จะมีการตัดการติดต่อ (reset) จากเครื่องที่เขาแอบอ้าง มีหลายวิธีทำการบุกรุกได้สำเร็จ เช่น

- ทำการบุกรุกขณะที่เครื่องที่เขาแอบอ้างกำลังล่ม
- ทำให้เครื่องที่เขาแอบอ้างเสียหาย แล้วทำการบุกรุก
- ทำให้เกิดการสับสนในการหาเส้นทางระหว่างเครื่องที่เขาแอบอ้างและ เป้าหมาย
- ใช้การบุกรุกในแพ็กเก็ตที่มีการโต้ตอบหนแรกเพื่อว่าจะไม่มีการตัดการติดต่อ (reset)

เดิมปัญหาเป็นปัญหาที่เล็ก ๆ แต่ปัจจุบันกลายเป็นปัญหาใหญ่แล้ว

การบุกรุกปลอมแปลงแบบคนกลาง (man in the middle)

- ผู้บุกรุกจะเข้าไปอยู่ในเส้นทางระหว่างคุณและเครื่องที่เขาแอบอ้าง (real machine) โดยเข้าไปที่ใกล้จุดปลายของเส้นทาง เพราะเส้นทางจะถูกเปลี่ยนแปลงทุกวินาที
- การเปลี่ยนแปลงเส้นทางระหว่างเครื่อง (machine) จะง่ายหรือยากขึ้นอยู่กับรูปแบบการวางเครือข่าย (topology) ระบบการเราติ้ง (routing)

จากการปลอมแปลงข้างต้นทำให้ไม่สามารถเชื่อถือได้ ดังนั้นหากจะให้ทำการเชื่อถือได้ควรจะมีการทำการเข้ารหัส (encryption) เพื่อที่จะทำให้มีความปลอดภัยบนเส้นทางที่ไม่มีความปลอดภัย

5.10 ฟิลเตอร์ริ่งโดยการบริการ (filtering by service)

ที่ผ่านมาเป็นการป้องกันโดยตำแหน่ง (address) เพียงอย่างเดียว แต่ในการใช้แพ็กเก็ตฟิลเตอร์ริ่งยังมีการฟิลเตอร์ริ่งโดยการบริการ (service) ซึ่งมีความยุ่งยาก

จากแพ็กเก็ตฟิลเตอร์ริ่งที่ดูมาแพ็กเก็ตฟิลเตอร์ริ่งยังสามารถเกี่ยวข้องกับบริการ (service) โดยเฉพาะ ดังตัวอย่างที่นำมาบรรยายต่อไปโดยดูที่เทลเน็ต (TELNET) โดยเทลเน็ตอนุญาตสำหรับผู้ใช้ในการล็อกอิน (login) เข้าไปในระบบอื่น ๆ ราวกับว่าผู้ใช้มีการติดต่อโดยตรงกับระบบ

5.11 บริการเทลเน็ตออกข้างนอก (outbound TELNET service)

เริ่มแรกดูที่การเทลเน็ตออกข้างนอก ซึ่งมีไคลเอนต์ท้องถิ่น (local client : user) ติดต่อไปที่เซิร์ฟเวอร์ทางไกล (remote server) จะต้องมีการจัดการทั้งแพ็กเก็ตเข้าและออก

แพ็กเก็ตที่ออกไปของการเทลเน็ตออก (outbound) จะบรรจุคีย์ที่กดของผู้ใช้และลักษณะเฉพาะต่อไปนี้

- ไอพีตำแหน่งตั้งต้น (IP source address) ของแพ็กเก็ตที่ออกไปจากโฮสต์ท้องถิ่น (local host)
- ไอพีตำแหน่งปลายทาง (IP destination address) ที่เป็นโฮสต์ทางไกล (remote host) เทลเน็ตเป็นบริการพื้นฐานของทีซีพี (TCP) ดังนั้นไอพีแพ็กเก็ต (IP packet) จะเป็นชนิดทีซีพี
- ทีซีพีพอร์ต (TCP port) ปลายทางเป็น 23 ซึ่งเป็นที่รู้จักอยู่แล้วว่าเป็นเลขพอร์ทที่เทลเน็ตใช้
- ทีซีพีพอร์ทต้นทางซึ่งเป็นเลขที่สุ่มขึ้นมา มีค่ามากกว่า 1023

- แพ็กเก็ตแรกที่ออกไปจะไม่มีบิตแอค (ACK)

แพ็กเก็ตที่เข้ามาจะบรรจุข้อมูลที่ใช้แสดงบนหน้าจอของผู้ใช้ (เช่น " login : " prompt) และมีลักษณะเฉพาะต่อไปนี้

- ไอพีที่อยู่ตั้งต้นของแพ็กเก็ตที่เข้ามาจากโฮสต์ทางไกล
- ไอพีที่อยู่ปลายทางที่เป็นโฮสต์ท้องถิ่น
- ไอพีแพ็กเก็ตชนิดที่เป็นทีซีพี (TCP)
- ทีซีพีพอร์ทตั้งต้น (TCP source port) หมายเลข 23 ซึ่งเป็นพอร์ทที่เซิร์ฟเวอร์ใช้
- ทีซีพีพอร์ทปลายทาง (TCP destination port)
- ทุกๆแพ็กเก็ตที่จะเข้ามาซึ่งมีบิตแอค (ACK)

5.12 บริการเทลเน็ตเข้าข้างใน (inbound TELNET service)

ต่อมาดูที่บริการเทลเน็ตเข้าข้างใน (inbound TELNET service) ซึ่งเป็นส่วนของไคลเอนต์ทางไกล (remote client : remote user) ซึ่งต้องมีการจัดการทั้งแพ็กเก็ตที่เข้าและออก

แพ็กเก็ตที่เข้ามาสำหรับบริการเทลเน็ตเข้าข้างใน (inbound TELNET service) จะบรรจุคีย์ที่ผู้ใช้กด และลักษณะเฉพาะต่อไปนี้

- ไอพีที่อยู่ตั้งต้นของแพ็กเก็ตที่เข้ามาจากโฮสต์ทางไกล
- ไอพีที่อยู่ปลายทางที่เป็นโฮสต์ท้องถิ่น
- ไอพีแพ็กเก็ตชนิดที่เป็นทีซีพี (TCP)
- ทีซีพีพอร์ทตั้งต้นซึ่งเป็นเลขที่สุ่มขึ้นมามากกว่า 1023
- ทีซีพีพอร์ทปลายทาง
- ทีซีพีแอคบิต (TCP ACK bit) ที่ไม่ตั้งค่าทุกๆแพ็กเก็ตแรก

แพ็กเก็ตที่ออกมาบรรจุข้อมูลที่จะแสดงให้ผู้ใช้ (server responses) และลักษณะเฉพาะต่อไปนี้

- ไอพีที่อยู่ตั้งต้นของแพ็กเก็ตที่เข้ามาจากโฮสต์ท้องถิ่น
- ไอพีที่อยู่ปลายทางที่เป็นโฮสต์ทางไกล
- ไอพีแพ็กเก็ตชนิดที่เป็นทีซีพี (TCP)
- ทีซีพีพอร์ทตั้งต้นหมายเลข 23
- ทีซีพีพอร์ทปลายทาง
- ทุกๆแพ็กเก็ตที่จะเข้ามาซึ่งมีบิตแอค (ACK)

5.13 สรุปเทลเน็ต (TELNET summary)

จากข้อมูลข้างต้นเราสามารถทำเป็นตารางดังนี้

Service Direction	Packet Direction	Source Address	Dest. Address	Packet Type	Source port	Dest. Port	ACK set
Outbound	Outgoing	Internal	External	TCP	Y	23	a
Outbound	Incoming	External	Internal	TCP	23	y	Yes
Inbound	Incoming	External	Internal	TCP	Z	23	a
Inbound	Outgoing	Internal	External	TCP	23	Z	Yes

ตาราง 5.3 แสดงเทลเน็ต

* a เป็นทีซีพีแอคบิต (TCP ACK bit) ซึ่งจะเซ็ตตลอดยกเว้นแพ็กเก็ตแรก ส่วน Y และ Z เป็นค่าสุ่มขึ้นที่มีค่ามากกว่า 1023

ถ้าคุณต้องการอนุญาตการเทลเน็ตออก แต่อย่างอื่นไม่ คุณสามารถจัดตั้งแพ็กเก็ตฟิลเตอร์ริ่งของคุณดังต่อไปนี้

Rule	Direc-tion	Source Address	Dest Address	Pro-tocol	Source Port	Dest. Port	ACK set	Action
A	Out	Internal	Any	TCP	> 1023	23	Either	Permit
B	In	Any	Internal	TCP	23	> 1023	Yes	Permit
C	Either	Any	Any	Any	Any	Any	Either	Deny

ตาราง 5.4 แสดงเฉพาะเทลเน็ตออกเท่านั้น

- กฎ A อนุญาตให้แพ็กเก็ตออกไปที่เทลเน็ตเซิร์ฟเวอร์ทางไกล (remote TELNET server)
- กฎ B อนุญาตให้รับแพ็กเก็ตกลับมา เพราะมันจะเปรียบเทียบที่บิตแอค (ACK bit) ว่าเป็นเซ็ต
- กฎ C เป็นกฎปกติ ถ้าไม่ตรงกับที่ตั้งไว้จะทำการขัดขวาง

บทที่ 6
ระบบพร็อกซี (proxy system)

พร็อกซีเซิร์ฟเวอร์ (proxy server) มักจะใช้สำหรับโปรโตคอลพิเศษ หรือกลุ่มของโปรโตคอลที่ทำงานบนดูอัลโฮมโฮสต์ (Dual-homed host) หรือ แบสชั่นโฮสต์ (bastion host) โดยโปรแกรมที่ผู้ใช้ใช้จะคุยกับพร็อกซีเซิร์ฟเวอร์ (proxy server) แทนที่จะติดต่อโดยตรงกับเซิร์ฟเวอร์จริง (real server) ถ้ามีการร้องขอ (request) จากไคลเอนต์ (client) มาถึงพร็อกซีเซิร์ฟเวอร์ พร็อกซีเซิร์ฟเวอร์จะคุยกับเซิร์ฟเวอร์จริงโดยประพฤติตัวเหมือนไคลเอนต์ (คือ proxy) และส่งการร้องขอจากไคลเอนต์ถึงเซิร์ฟเวอร์จริงและส่งคำตอบจากเซิร์ฟเวอร์จริง กลับมาที่ไคลเอนต์ ดังนั้นสำหรับผู้ใช้ที่คุยกับพร็อกซีเซิร์ฟเวอร์ก็จะเหมือนกับคุยกับเซิร์ฟเวอร์จริงส่วน เซิร์ฟเวอร์จริง ก็จะไม่รู้ว่าผู้ใช้อยู่ที่ใด

ระบบพร็อกซีไม่ต้องการฮาร์ดแวร์พิเศษ แต่ต้องการซอฟท์แวร์พิเศษสำหรับการบริการทั้งหมด

ระบบพร็อกซีมีประสิทธิภาพเมื่อกำหนดการเชื่อมต่อพร็อกซีในระดับไอพี (IP) ระหว่างไคลเอนต์ และ เซิร์ฟเวอร์จริง เช่น สกรีนนิ่งเราเตอร์ (screening router) หรือดูอัลโฮมโฮสต์ (Dual-homed host) ซึ่งไม่สามารถกำหนดเส้นทางให้แพ็กเก็ต ถ้ามีการเชื่อมต่อระดับไอพี (IP) ระหว่างไคลเอนต์ และเซิร์ฟเวอร์จริง ไคลเอนต์สามารถอ้อมผ่านระบบพร็อกซีได้

6.1 ทำไมต้องใช้พร็อกซี (proxy)

ในระบบไฟร์วอลล์ที่ผ่านมาผู้ใช้ที่ต้องการจะเข้าถึงบริการของเครือข่ายไม่สามารถทำได้โดยตรง คือต้องทำการล็อกอิน (login) เข้าไปในระบบดูอัลโฮมโฮสต์ (Dual-homed host) โดยทำงานทุกงานจากที่นี่ แล้วส่งผลลัพธ์กลับไปที่เวิร์คสเตชั่น (workstation) ของคุณ

ปัญหาเกิดขึ้นเมื่อมีระบบปฏิบัติการ (operating system) หลายระบบ เช่น ถ้าที่ที่คุณทำงานใช้ระบบแมคอินทอช (Macintosh) และดูอัลโฮมโฮสต์ (Dual-homed host) เป็นระบบยูนิกซ์ (UNIX) คุณจะถูกจำกัด เครื่องมือ (tools) ที่สามารถใช้ได้บนดูอัลโฮมโฮสต์ (Dual-homed host) เพราะเครื่องมือของคุณไม่เหมือนกับ เครื่องมือบนดูอัลโฮมโฮสต์

ปัญหาที่อาจจะเกิดขึ้นอีกอย่างหนึ่งก็คือ ไม่สามารถควบคุมเครื่องมือที่ใช้ได้ คือ ผู้ใช้ของคุณสามารถส่งเครื่องมือจากภายใน ตัวอย่างเช่น บนดูอัลโฮมโฮสต์ (Dual-homed host) ไม่สามารถแน่ใจได้ว่าทุกไฟล์ที่ส่งออกไปจะถูกตรวจสอบ เพราะเมื่อส่งออกไปแล้วไฟล์นี้อาจถูกนำไปใช้ได้โดยไม่ต้องล็อกอิน (login)

พร็อกซี (proxy) จะทำงานได้โดยอัตโนมัติสำหรับดูอัลโฮมโฮสต์ (Dual-homed host) พร็อกซีจะทำให้ทุกการทำงานอยู่หลังฉาก ผู้ใช้จะถูกทำให้เห็นเหมือนกับติดต่อโดยตรงกับเซิร์ฟเวอร์บนอินเตอร์เน็ตซึ่งจะมีการทำงาน ดังรูป 6.1

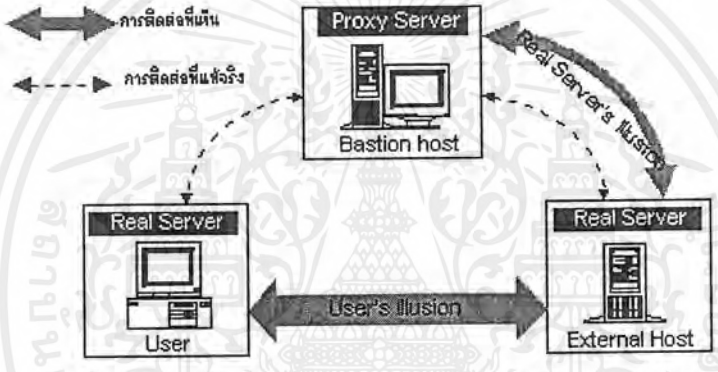
พร็อกซีซอฟท์แวร์ (proxy software) ทำงานได้โดยไม่ต้องอาศัยล็อกอินของผู้ใช้ (users login) เข้าไปในโฮสต์ ดังนั้นเป็นไปไม่ได้ที่ไฟล์ที่ควบคุมไม่ได้จะไปถึงภายนอก เพราะพร็อกซีจะทำการตรวจสอบเสมอ

6.2 ข้อดีของพร็อกซี (proxy)

6.2.1 พร็อกซีอนุญาตให้ผู้ใช้เข้าถึงอินเตอร์เน็ตได้โดยตรง

ในดูอัลโฮมโฮสต์ (Dual-homed Host) ผู้ใช้ต้องล็อกอิน (login) เข้าไปในโฮสต์ก่อนใช้บริการของอินเตอร์เน็ตซึ่งทำให้เป็นการไม่สะดวกสบาย ซึ่งพร็อกซีบริการให้ผู้ใช้ติดต่อได้โดยตรง

ขณะที่พร็อกซีอนุญาตให้ผู้ใช้เข้าถึงบริการของอินเตอร์เน็ตจากระบบของคุณ จะไม่อนุญาตให้แพ็กเก็ต ผ่านระบบของผู้ใช้และอินเตอร์เน็ต เพราะในเส้นทางนี้จะต้องมีการผ่านดูอัลโฮมโฮสต์ หรือแบสชั่นโฮสต์ที่มี สกรีนนิ่งเราเตอร์



รูป 6.1 แสดงสิ่งที่เห็นกับการทำงานของพร็อกซี (proxy)

6.2.2 พร็อกซีมีการเก็บของมูลที่ดี

เพราะว่าพร็อกซีเซิร์ฟเวอร์ (proxy server) เข้าใจโปรโตคอลที่แอบแฝง ดังนั้นการเก็บข้อมูลเข้าออกก็สามารถเก็บได้ตามที่ต้องการ เช่น แทนที่จะเก็บทุกๆข้อมูลที่ส่งผ่าน เอฟทีพีเซิร์ฟเวอร์ (FTP server) ก็จะเก็บเฉพาะคำสั่งและผลที่ทำ ดังนั้นจะทำให้เก็บข้อมูลได้เล็กกว่าและตรงกับความต้องการ

6.3 ข้อเสียของพร็อกซี (proxy)

6.3.1 การบริการของพร็อกซี (proxy services) มีมาตามหลังการไม่บริการของพร็อกซี (nonproxied services)

พร็อกซีซอฟท์แวร์ (proxy software) ที่ออกมามักจะตอบสนองกับบริการเก่า ๆ หรือบริการทั่วไป เช่น เอฟทีพี (FTP) และเทลเน็ต (TELNET) เพราะเป็นการยากที่จะสร้างซอฟท์แวร์มาตอบเสนอและอาจทำให้ ความปลอดภัยลดลงได้

6.3.2 การบริการของพร็อกซีอาจจะต้องการเซิร์ฟเวอร์ที่แตกต่างกันสำหรับแต่ละบริการ

บางครั้งต้องการพร็อกซีเซิร์ฟเวอร์ที่แตกต่างกันสำหรับแต่ละโปรโตคอล เพราะพร็อกซีเซิร์ฟเวอร์ต้องเข้าใจโปรโตคอลเพื่อจะหาว่าอนุญาตได้ไหม และเพื่อที่จะแสดงกับไคลเอนต์เป็นเซิร์ฟเวอร์จริง

6.3.3 พร็อกซีมักจะต้องการการเปลี่ยนแปลงกับไคลเอนต์ (client) , ขบวนการ (procedure) หรือทั้งคู่

บางบริการที่ไม่ได้ถูกออกแบบมาสำหรับพร็อกซี พร็อกซีเซิร์ฟเวอร์ต้องการที่จะเปลี่ยนแปลงกับไคลเอนต์หรือขบวนการ (procedure) ถ้าการเปลี่ยนแปลงบางอย่างปัญหาก็อาจจะไม่สามารถใช้เครื่องมือนั้นได้เหมือนกับไม่มีพร็อกซี (non-proxy)

6.3.4 การบริการของพร็อกซีใช้ไม่ได้กับบนบางบริการ

พร็อกซีแทรกอยู่ระหว่างการติดต่อโดยตรงกันระหว่างไคลเอนต์และเซิร์ฟเวอร์จริงแต่การบริการเช่น ทอล์ค (talk) นั้นจะมีความยุ่งเหยิงและไม่เป็นระเบียบ ซึ่งอาจจะใช้ไม่ได้กับพร็อกซี

6.3.5 การบริการของพร็อกซีไม่ได้ปกป้องจากโปรโตคอลที่ไม่มั่นคง

บางโปรโตคอลถูกสร้างมาเพื่อส่งข้อมูลผ่านเข้ามาปฏิบัติ (execute) โดยไม่ผ่านพร็อกซี ดังนั้นพร็อกซี จึงไม่สามารถตรวจสอบได้

6.4 พร็อกซีทำงานอย่างไร

พร็อกซีต้องการพร็อกซีเซิร์ฟเวอร์ซอฟท์แวร์ที่เหมาะสมกับเซิร์ฟเวอร์ และบนด้านไคลเอนต์จำเป็นต้องมีสิ่งหนึ่งต่อไปนี้

6.4.1 ไคลเอนต์ซอฟท์แวร์ (custom client software)

ซอฟท์แวร์จะต้องรู้ว่าจะติดต่อพร็อกซีเซิร์ฟเวอร์อย่างไร แทนที่เซิร์ฟเวอร์จริง (real server) เมื่อผู้ใช้ทำการร้องขอ (request) เช่น เอฟทีพี (FTP) หรือเทลเน็ต (TELNET) และจะบอกกับพร็อกซีเซิร์ฟเวอร์อย่างไรเมื่อเซิร์ฟเวอร์จริงจะติดต่อ

6.4.2 ขบวนการของผู้ใช้ (custom user procedure)

ผู้ใช้ใช้มาตรฐานของไคลเอนต์ซอฟท์แวร์ในการที่จะคุยกับพร็อกซีเซิร์ฟเวอร์ และค่อยติดต่อกับเซิร์ฟเวอร์จริงแทนที่การติดต่อกับเซิร์ฟเวอร์จริงโดยตรง

6.5 คำศัพท์เฉพาะของพร็อกซีเซิร์ฟเวอร์ (proxy server)

6.5.1 ระดับของแอพพลิเคชั่น (application-level) เทียบกับระดับของวงจรไฟฟ้า (circuit-level)

ระดับของแอพพลิเคชั่น (application-level) พร็อกซีรู้เกี่ยวกับแอพพลิเคชั่นพิเศษที่จัดการกับการบริการของพร็อกซีมันจะเข้าใจและแปลคำสั่งในแอพพลิเคชั่นโปรโตคอล ส่วนระดับของวงจรไฟฟ้า (circuit-level) เป็นการสร้างวงจรระหว่างไคลเอนต์และเซิร์ฟเวอร์โดยไม่มีการแปลคำสั่งของแอพพลิเคชั่นโปรโตคอล

โดยทั่วไประดับของแอพพลิเคชั่นของพร็อกซีใช้ในการปรับเปลี่ยนขบวนการ (procedure) และระดับของวงจรไฟฟ้าของพร็อกซีใช้ในการปรับเปลี่ยนไคลเอนต์

6.5.2 เจเนอริคพร็อกซี (generic proxies) เทียบกับ ดีดิเคตพร็อกซี (dedicated proxies)

ดีดิเคตพร็อกซีเซิร์ฟเวอร์ (dedicated proxy server) เป็นการบริการเพียงโปรโตคอลเดียว ส่วนเจเนอริคพร็อกซีเซิร์ฟเวอร์ (generic proxy server) เป็นการบริการได้หลายๆ โปรโตคอล ในทางปฏิบัติดีดิเคตพร็อกซีเซิร์ฟเวอร์ คือ ระดับของแอพพลิเคชั่น และเจเนอริคพร็อกซีเซิร์ฟเวอร์ คือ ระดับของวงจรไฟฟ้าแต่ก็เป็นไปได้ที่จะสร้าง เจเนอริคแอพพลิเคชั่นเลฟเวลพร็อกซีเซิร์ฟเวอร์ (generic application-level proxy server คือ เข้าใจหลาย ๆ โปรโตคอล) หรือดีดิเคตเซอกิตเลฟเวลพร็อกซีเซิร์ฟเวอร์ (dedicated circuit-level proxy server คือ จัดการพียง บริการเดียว)

6.5.3 พร็อกซีเซิร์ฟเวอร์แบบฉลาด (intelligent proxy servers)

พร็อกซีเซิร์ฟเวอร์สามารถทำงานได้มากกว่าการส่งผ่านการร้องขอที่เข้ามา นั้นคือถ้าหากเราเพิ่มความสามารถเข้าไป (ข้อจำกัดต่าง ๆ) ความสามารถจะเพิ่มขึ้นได้ หรือทำการแบ่งเป็นเซิร์ฟเวอร์ที่เป็นพร็อกซีของแต่ละ บริการ

6.5.4 การใช้พร็อกซีกับบริการของอินเตอร์เน็ต

เพราะว่าพร็อกซีแทรกการติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ มันจึงต้องมีการปรับเปลี่ยนและแบ่งแต่ละบริการบางอย่างก็จัดการได้ง่ายแต่บางอันก็ยาก

6.5.5 ทีซีพี (TCP) เทียบกับโปรโตคอลอื่นๆ

เพราะว่าทีซีพี (TCP) เป็นโปรโตคอลที่มีการติดต่อรวมกับโปรโตคอลอื่นๆ (connection-oriented protocol) ดังนั้นต้องทำการแบ่งส่วนการพร็อกซีแล้วจึงนำมารวมกัน

6.5.6 การเชื่อมต่อทิศทางเดียว (unidirectional connections) เทียบกับการเชื่อมต่อหลายทิศทาง (multidirectional connections)

เป็นการง่ายที่พร็อกซีเซิร์ฟเวอร์ขัดขวางการติดต่อจากไคลเอนต์ถึงเซิร์ฟเวอร์ แต่มันยากกว่าเมื่อจะขัดขวางจากเซิร์ฟเวอร์ถึงไคลเอนต์ ดังนั้นผลลัพธ์ของการติดต่อจะต้องเป็นที่ทราบของพร็อกซีเซิร์ฟเวอร์ หรือเซิร์ฟเวอร์ จะต้องแปลและปรับเปลี่ยนโปรโตคอลให้ผลลัพธ์ที่ออกมาถูกต้องแน่นอน

6.5.7 ไคลเอนต์ภายใน (internal clients) เทียบกับไคลเอนต์ภายนอก (external clients)

พร็อกซีเซิร์ฟเวอร์ส่วนมากจะออกแบบให้เหมาะกับไคลเอนต์ภายในและเซิร์ฟเวอร์ภายนอกของไฟร์วอลล์ เพราะว่าต้องใช้งานร่วมกันทั้งภายในและภายนอก แต่ภายในจะสามารถปรับเปลี่ยนไคลเอนต์สำหรับภายนอกทำงานตามความเหมาะสม

6.5.8 พร็อกซีโดยไม่มีพร็อกซีเซิร์ฟเวอร์

บางบริการที่เรียกว่า "เก็บและส่ง" (store-and-forward) เช่น เอสเอ็มทีพี (SMTP) , เอ็นเอ็นทีพี (NNTP), เอ็นทีพี (NTP) ทำงานเหมือนกับพร็อกซี บริการเหล่านี้เกี่ยวกับข่าวสารข้อความ (Email => SMTP , Usenet news => NNTP , clock setting => NTP) ซึ่งรับจากเซิร์ฟเวอร์และส่งต่อไปให้เซิร์ฟเวอร์อื่นๆ ต่อไป ซึ่งเซิร์ฟเวอร์ตัวกลางของการส่งนี้แสดงเหมือนกับพร็อกซี

6.6 การใช้ซ็อคส์ (Socks) สำหรับพร็อกซี

ซ็อคส์แพ็กเกจ (Socks package) ถูกเขียนขึ้นโดยเดวิด โกแบลส (David Koblas) และมิชเชลล์ โกแบลส (Michelle Koblas) และตรวจสอบโดย Ying-Da Lee ซึ่งเป็นระบบพร็อกซีที่ต้องการไคลเอนต์ทั่วไป (custom clients)

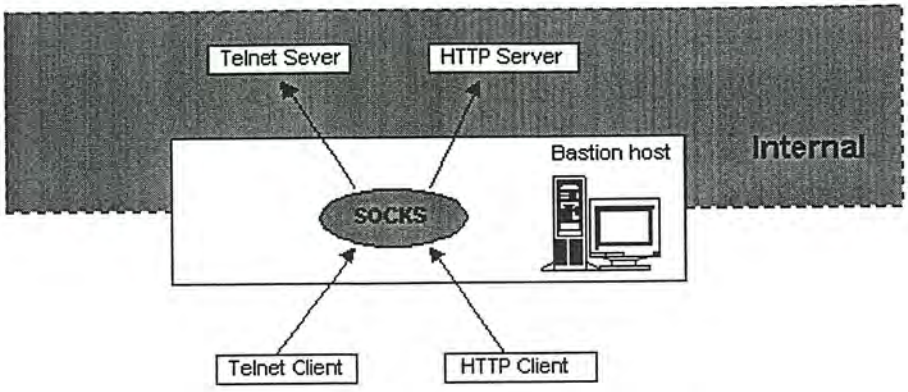
เพื่อที่จะทำให้ ง่ายสำหรับไคลเอนต์ใหม่ๆซ็อคส์ (Socks) จึงเป็นเจเนอริค (generic) ซึ่งทำให้ถูกใช้มาก แต่มันก็มีข้อเสียก็คือ ซ็อคส์ไม่ฉลาดในการเก็บข้อมูลและควบคุมการเข้าถึงมันกระทำการเก็บข้อมูลแต่ส่วนมากมันจะเก็บจากไคลเอนต์ ซ็อคส์ทำการเก็บข้อมูลการร้องขอ (request) บนเซิร์ฟเวอร์จัดการควบคุมการเข้าถึงโดยที่ตั้ง (source) , โฮสต์ปลายทาง และโปรโตคอล และสามารถคอนฟิกให้เข้าถึงไม่ได้

อุปสรรคอย่างหนึ่งของซ็อคส์ คือทำงานเฉพาะบนฐานของทีซีพีไคลเอนต์ (TCP clients) มันไม่สามารถทำงานบนฐานของยูดีพีไคลเอนต์ (UDP clients) ถ้าต้องการใช้ฐานของยูดีพีไคลเอนต์จะต้องเอาแพ็กเกจ (package) อื่นมาช่วย

ข้อได้เปรียบของซ็อคส์ (Socks) คือ มีการใช้อย่างกว้างขวาง , เซิร์ฟเวอร์ปฏิบัติการได้

ส่วนประกอบของซ็อคส์ (Socks) มีดังนี้

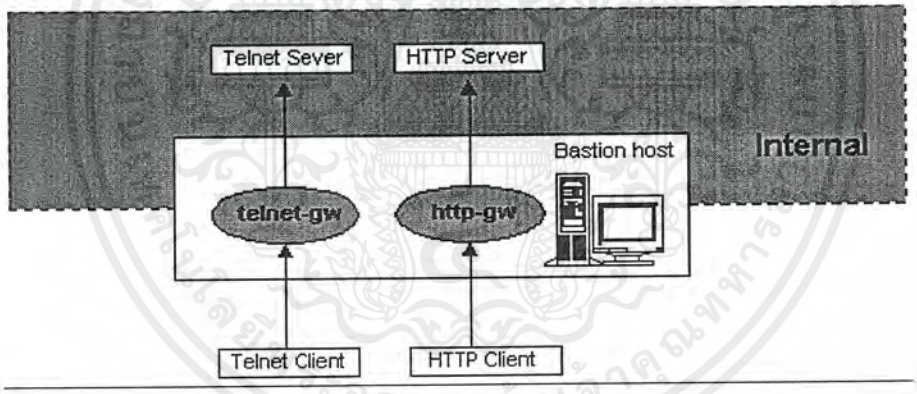
- ซ็อคส์เซิร์ฟเวอร์ (Socks server) ซึ่งเซิร์ฟเวอร์นี้จะต้องทำงานบนระบบยูนิกซ์ (UNIX)
- ซ็อคส์ ไคลเอนต์ไลบารี่ (Socks client library) สำหรับเครื่องยูนิกซ์
- ตัวแปลซ็อคส์ (Socks-ified) รูปแบบของหลายๆ มาตรฐานยูนิกซ์ไคลเอนต์



รูป 6.2 แสดงการใช้ซ็อคส์ (Socks) สำหรับพร็อกซี

6.7 การใช้ทีไอเอสอินเตอร์เน็ตไฟร์วอลล์ทูลคิท (TIS Internet Firewall Toolkit) สำหรับพร็อกซี

ด้วยเหตุที่ซ็อคส์ (Socks) พยายามจัดการพร็อกซีทั้งหมดด้วยตัวมันตัวเดียวแต่ทีไอเอสอินเตอร์เน็ตไฟร์วอลล์ทูลคิท (TIS FWTK) จะจัดการพร็อกซีเฉพาะบริการในอินเตอร์เน็ตเท่านั้น ความคิดนี้ทำได้โดยแบ่งโปรแกรมเล็กๆจากการคอนฟิกของไฟล์มันสามารถจัดการเป็นอินเทลลิเจนท์ (intelligent proxy) ที่ปลอดภัยแน่นอน



รูป 6.3 แสดงการใช้ทีไอเอสอินเตอร์เน็ตไฟร์วอลล์ทูลคิท (TIS FWTK) สำหรับพร็อกซี

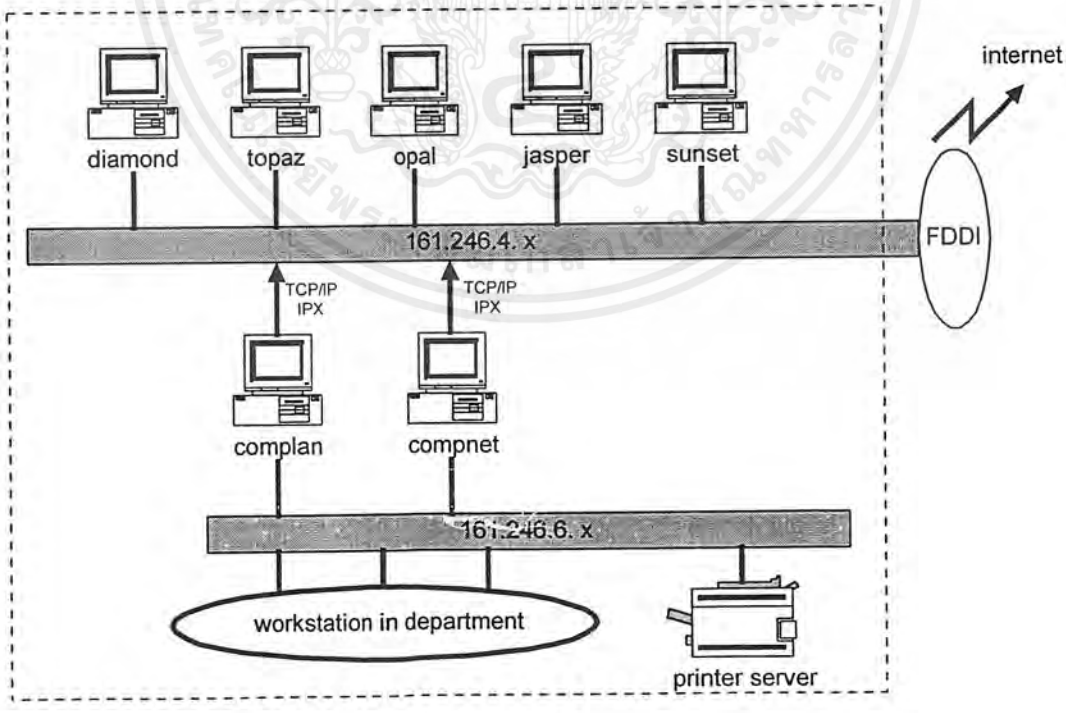
บทที่ 7
การดำเนินการติดตั้งระบบไฟร์วอลล์

7.1 ขั้นตอนการดำเนินการติดตั้งระบบไฟร์วอลล์ทั้งหมด

การศึกษาติดตั้งระบบไฟร์วอลล์ให้เข้าได้กับเครือข่ายใด ๆ นั้น จำเป็นต้องทราบความเป็นไปของระบบทั้งหมด กล่าวคือ ผู้ติดตั้งระบบจำเป็นจะต้องเป็นผู้อยู่ระบบนั้น ๆ จนเล็งเห็นปัญหาและจุดบกพร่องที่เกิดขึ้นกับระบบเครือข่ายของตน เราอาจกล่าวได้ว่าระบบเครือข่ายทุกระบบมีความจำเป็นที่จะต้องติดตั้งระบบไฟร์วอลล์เพื่อเพิ่มมาตรฐานความปลอดภัยให้กับระบบเครือข่ายของตน แต่ก็อาจมีเครือข่ายอีกมากมายที่เราไม่จำเป็นจะต้องติดตั้งระบบไฟร์วอลล์เลย เพราะระบบปฏิบัติการที่มีอยู่ในระบบเครือข่ายได้จัดการในส่วนนี้ให้เราเรียบร้อยแล้ว หรืออาจกล่าวได้ว่า ระบบไฟร์วอลล์ไม่เหมาะสมสำหรับเครือข่ายที่มีขนาดเล็ก เช่น มีโฮสต์ในระบบน้อยกว่าสองเครื่อง เป็นต้น

โครงงานนี้ใช้เครือข่ายของภาควิชาวิศวกรรมคอมพิวเตอร์เป็นเครือข่ายในการดำเนินงานระบบไฟร์วอลล์ เพื่อให้เข้าได้กับระบบเครือข่ายที่มีอยู่ จะขอแบ่งการอธิบายออกเป็น 2 หัวข้อใหญ่ ๆ ดังนี้

- กำหนดบริการที่มีอยู่เดิมทั้งหมดของระบบ (จะเป็นบริการที่เกิดจากโปรโตคอลทีซีพีไอพีเท่านั้น (TCP/IP) โปรโตคอลอื่น ๆ เช่นไอพีเอ็กซ์ (IPX) เราจะไม่กล่าวถึง เนื่องจากไม่เกี่ยวข้องกับระบบไฟร์วอลล์บนโปรโตคอลทีซีพีไอพี)
- ดำเนินการระบบไฟร์วอลล์ โดยใช้โปรแกรมที่ได้จากไฟร์วอลล์ทูลคิท (FWTK) โดยแยกการดำเนินการออกเป็นบริการย่อย ๆ



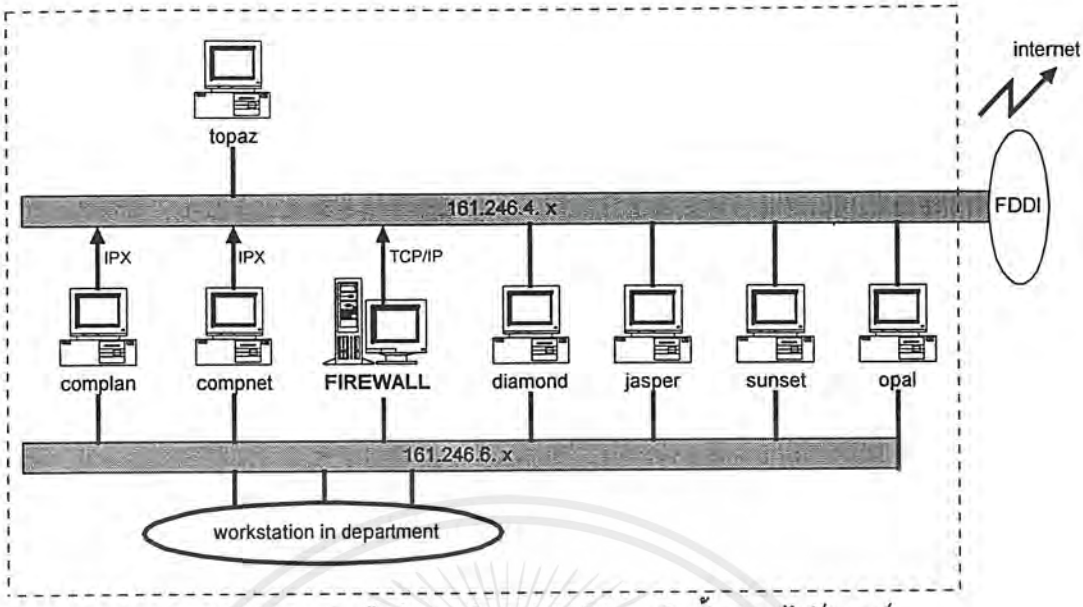
รูป 7.1 ระบบเครือข่ายภายในก่อนดำเนินการติดตั้งระบบไฟร์วอลล์

7.1.1 บริการที่มีอยู่เดิม

จะแยกตามบริการที่มีอยู่ตามรายชื่อของโฮสต์ที่มีอยู่ดังนี้

- ไดมอนด์ (diamond)
  - โปรแกรมรับส่งเมล์ (sendmail server)
  - โปรแกรมดีเอ็นเอส ซึ่งเป็นตัวหลัก (primary DNS)
  - โปรแกรมบริการรีโมตล็อกอิน (TELNET server)
  - โปรแกรมบริการโอนย้ายแฟ้มข้อมูล (FTP server)
- โทพาส (topaz)
  - โปรแกรมบริการเว็บเซิร์ฟเวอร์ของภาควิชา (WWW)
  - โปรแกรมบริการรีโมตล็อกอิน (TELNET server)
- แจสเพอร์ (jasper)
  - ให้บริการโมเด็ม (modem) แก่ผู้ใช้บริการภายในภาควิชา ฯ โดยมีบริการทั้งหมด 2 หมายเลขดังนี้คือ 3269969 และ 7390678
  - โปรแกรมดีเอ็นเอสตัวรอง (secondary DNS)
- โอปอล (opal) และซันเซ็ต (sunset)
  - ให้บริการเป็นเครื่องเอ็นจิเนียริงเวิร์คสเตชั่นที่วิ่งโปรแกรมประเภทเอ็กซ์วินโดว์
- คอมพ์แลน (complan) และคอมพ์เน็ต (compnet)
  - เป็นไฟล์เซิร์ฟเวอร์ให้กับภาควิชา ฯ ทั้งหมด
  - บริการเป็นเครื่องดิสก์เลส (diskless) ให้กับเครื่องพีซีเวิร์คสเตชั่น
  - เป็นตัวบริการดีเอชซีพี (DHCP) แก่เครื่องพีซีเวิร์คสเตชั่นทั้งหมดในภาควิชา ฯ
  - เป็นไอพีเอ็กซ์เกตเวย์ (IPX gateway) และทีซีพีไอพีเกตเวย์ (TCP/IP gateway)

7.1.2 ทำการติดตั้งใช้งานจริง



รูป 7.2 ระบบเครือข่ายภายในหลังดำเนินการติดตั้งระบบไฟร์วอลล์

หลังจากที่มีการติดตั้งระบบไฟร์วอลล์แล้ว บริการต่างๆที่เคยมีบนโฮสต์บางตัวจำเป็นต้องมีการย้ายไปโฮสต์ตัวอื่นๆที่มีความเหมาะสมกับระบบไฟร์วอลล์ที่เราทำการติดตั้งไว้ด้วย ดังนี้

จะแยกตามบริการที่มีอยู่ตามรายชื่อของโฮสต์ที่มีอยู่ดังนี้

- ไดมอนด์ (diamond)
  - โปรแกรมรับส่งเมล์ (sendmail)
  - โปรแกรมดีเอ็นเอส ซึ่งเป็นตัวรอง (Secondary DNS)
  - โปรแกรมบริการรีโมตล็อกอิน (TELNET Server)
- โทพาส (topaz)
  - โปรแกรมบริการเว็บเซิร์ฟเวอร์ของภาควิชา (WWW)
  - โปรแกรมบริการโอนย้ายแฟ้มข้อมูล (FTP Server)
- แจสเพอร์ (jasper)
  - ให้บริการโมเด็ม (modem) แก่ผู้ใช้บริการภายในภาควิชา ฯ โดยมีบริการทั้งหมด 2 หมายเลขดังนี้คือ 3269969 และ 7390678
- โอปอล (opal)
  - ให้บริการเป็นเครื่องเอ็นจิเนียริงเวิร์คสเตชั่นที่วิ่งโปรแกรมประเภทเอ็กซ์วินโดว์
- ซันเซ็ต (sunset)
  - เป็นออเธนทิเคชั่นเซิร์ฟเวอร์ (authentication server)

- ไฟร์วอลล์ (firewall)
  - เป็นเทลเน็ตพร็อกซี (TELNET proxy)
  - เป็นเอฟทีพีพร็อกซี (FTP proxy)
  - เป็นอาร์ล็อกอินพร็อกซี (rlogin proxy)
  - โปรแกรมรับส่งเมล์ (sendmail server)
  - โปรแกรมดีเอ็นเอสตัวหลัก (primary DNS)
  - เป็นทีซีพีไอพีเกตเวย์ (TCP/IP gateway)
- คอมพ์แลน (complan) และคอมพ์เน็ต (compnet)
  - เป็นไฟล์เซิร์ฟเวอร์ให้กับภาควิชา ฯ ทั้งหมด
  - บริการเป็นเครื่องดิสก์เลส (diskless) ให้กับเครื่องพีซีเวิร์คสเตชั่น
  - เป็นตัวบริการดีเอชซีพี (DHCP) แก่เครื่องพีซีเวิร์คสเตชั่นทั้งหมดในภาควิชา ฯ
  - เป็นไอพีเอ็กซ์เกตเวย์ (IPX gateway)

7.2 โปรแกรมที่มากับไฟร์วอลล์ทูลคิท (FWTK)

7.2.1 smap: บริการเอสเอ็มทีพี (SMTP Service)

เอสเอ็มทีพี (SMTP) ประกอบด้วยการทำงานของสองโพรเซส (process) หลัก ๆ คือ smap และ smapd โดยปกติแล้ว การทำงานของเอสเอ็มทีพีมีความเสี่ยงต่อการที่จะเป็นอันตรายต่อระบบ เนื่องจากตัวเมลเลอร์ (mailer) ทำงานตามการยินยอมของระบบ (system-level permission) ในการที่จะส่งเมล์ไปยังเมล์บ๊อกซ์ (mailbox) ของผู้ใช้ ซึ่ง smap และ smapd แก้ไขจุดอ่อนนี้โดยการจำกัดให้เมลเลอร์ทำงานอยู่ภายใต้ไดเร็กทอรีที่กำหนดไว้ ด้วยการทำคำสั่ง chroot แต่ smap และ smapd ไม่สามารถแก้ไขการส่งเมล์ด้วยชื่อปลอม (mail spoofing) และการบุกรุกแบบขัดขวางการให้บริการ (denial of service) โดยการใช้เมล์ จุดประสงค์หลักของ smap คือการป้องกันและแก้ไขจุดบอด (bug) ของโปรแกรมที่ทำให้ระบบมีจุดอ่อนให้เกิดการบุกรุกได้ การจัดการเกี่ยวกับเมล์โดยส่วนใหญ่ถูกทำโดยโปรแกรม sendmail ดังนั้นจึงไม่ต้องทำการดัดแปลงหรือแก้ไขโปรแกรม sendmail หรือไฟล์ที่เกี่ยวกับค่าต่าง ๆ ของมัน เมื่อระบบจากระยะไกลติดต่อมายังพอร์ตเอสเอ็มทีพี ระบบปฏิบัติการจะเรียก smap ขึ้นมาทำงาน โดย smap จะทำการ chroot ไปยังไดเร็กทอรีที่กำหนดไว้ทันที และเปลี่ยนรหัสของผู้ใช้ (user-id) เนื่องจาก smap ไม่ต้องใช้งานไฟล์ใด ๆ ที่เกี่ยวกับระบบ ไดเร็กทอรีนั้นจึงสามารถเป็นไดเร็กทอรีว่างได้ จะมีเฉพาะไฟล์ที่สร้างขึ้นจาก smap และ smap จะไม่ยุ่งเกี่ยวกับไฟล์ของระบบ เนื่องจากได้ทำการ chroot ไปแล้ว รวมทั้งจะไม่มีการทำงานแบบปฏิสัมพันธ์ (interactive) ผ่านทาง smap เนื่องจากไม่มีไฟล์ที่ทำงานได้ (executable file) อยู่ภายในสพูลไดเร็กทอรี (คือไดเร็กทอรีที่ทำการ chroot ไป)

smap จะทำการติดต่อกับโพรเซสต้นทางผ่านทางพอร์ตเอสเอ็มทีพี แล้วรับเมล์เข้ามา จากนั้นจะทำการเขียนเมล์ลงไปยังดิสก์ เขียนล็อกไฟล์และออกจากการทำงาน ส่วนที่สองของการทำงานเกี่ยวกับเมล์คือ smapd จะไม่ทำการ chroot หน้าที่ของ smapd คือการตรวจสอบสพูลไดเร็กทอรีอยู่เป็นระยะ ๆ และส่งเมล์ที่มีอยู่ไปยัง sendmail เพื่อให้ sendmail ทำการส่งต่อไป เมื่อ smapd ส่งเมล์ไปยัง sendmail แล้วก็จะลบไฟล์ออกจากสพูลไดเร็กทอรี

ด้วยหลักการนี้ หน้าที่ของ sendmail ก็ยังคงดำเนินไปอย่างปกติ ในขณะเดียวกันก็สามารถป้องกันไม่ให้ผู้ใช้จากเครือข่ายอื่นสามารถติดต่อกับระบบเครือข่ายของเราได้โดยตรง การตรวจสอบโปรแกรมของ sendmail ที่มีโค้ดประมาณ 20,000 บรรทัดเพื่อหาจุดอ่อนเป็นงานใหญ่ เมื่อเปรียบเทียบกับการตรวจสอบโปรแกรมของ smap ที่มีโค้ดเพียงประมาณ 700 บรรทัด

7.2.2 netacl: บริการเทลเน็ต และการควบคุมการเข้าถึงระบบ (TELNET Service and Network Access Control Lists)

inetd ไม่มีการควบคุมการเข้าถึงระบบ (access control) เนื่องจาก inetd จะอนุญาตให้ระบบเครือข่ายใด ๆ ติดต่อขอใช้บริการ (service) ได้ทุกบริการที่มีอยู่ในไฟล์ inetd.conf แต่ในบางทีเราอาจต้องการควบคุมการเข้าถึงระบบเครือข่าย และมีทูลต่าง ๆ ที่ช่วยในการทำงานนี้ netacl ซึ่งเป็นหนึ่งในทูลคิท ก็มีหน้าที่ในการควบคุมการเข้าถึงระบบเช่นเดียวกัน โดย netacl จะทำการควบคุมการขอใช้บริการจากเครือข่ายต่าง ๆ ด้วยการตรวจสอบแอดเดรสของโฮสท์นั้น ๆ

netacl ไม่มีการควบคุมการเข้าถึงระบบของบริการที่ใช้งานโปรโตคอลยูดีพี (UDP) แต่ใช้โปรแกรม tcp wrapper แทนในการควบคุมการเข้าถึงระบบของบริการที่ใช้งานโปรโตคอลยูดีพี ด้วยเทคโนโลยีที่มีอยู่ในปัจจุบัน เรายังไม่สามารถรับรอง (authenticating) ต้นทางของยูดีพีแพ็กเก็ตได้ ดังนั้น netacl จึงรับรองความปลอดภัยด้วยการปิดบริการที่ใช้งานโปรโตคอลยูดีพี (UDP-based service) ทั้งหมด

มีบริการหลายตัวที่ไม่ได้ถูกใช้งาน (disable) ตั้งแต่ระบบเริ่มต้นทำงาน (default) เช่น finger เราสามารถเปิดบริการ (enable) นั้น ๆ และอนุญาตให้โฮสต์ภายในเครือข่ายเท่านั้นใช้งานได้ และ netacl ยังสามารถควบคุมบริการอื่น ๆ ได้ด้วยการปรับตั้งค่า (configure) ให้เหมาะสม เช่น anonymous FTP เนื่องจากเราสามารถปรับตั้งค่าให้ทำการ chroot ก่อนจะปลุกโพรเซสที่ให้บริการนั้น ๆ ขึ้นมา ด้วยวิธีการทำงานแบบนี้ทำให้เกิดความสะดวกในการใช้งานบริการ (service) ที่ต้องแยกการทำงานของโพรเซสนั้นออกไป ยกตัวอย่างเช่น การให้บริการ finger พร็อกซีโดยการปรับตั้งค่า (configure) netacl ให้ปลุกโพรเซส fingerd ขึ้นมาทำงาน ทำการ chroot ไปที่ไดเร็กทอรีว่าง ๆ อันใดอันหนึ่ง เปรียบเสมือน fingerd เป็นโพรเซสที่ถูกแยกออกไปจากระบบก่อนที่จะเริ่มทำงาน ซึ่งถ้า fingerd มีช่องโหว่ในด้านความปลอดภัย (security hole) ก็ยังไม่เกิดความเสียหายมากนัก เนื่องจากผู้บุกรุกจะไม่สามารถใช้งานไฟล์ต่าง ๆ ของระบบ (file system) ได้

ในการปรับตั้งค่าของไฟร์วอลล์ (firewall configuration) โดยปกติแล้วจะใช้ netacl เพื่อกัน (block) โฮสต์เกือบทั้งหมด (ยกเว้นไว้เพียงบางเครื่อง) ไม่ให้สามารถล็อกอินเข้ามาในโฮสต์ไฟร์วอลล์ผ่านทางอาร์ล็อกอิน (rlogin) หรือเทลเน็ต (telnet) ได้ และใช้ในการป้องกันไม่ให้มีการเข้าถึง (access) ระบบจากโฮสต์ไคลเอนต์ที่เราไม่ไว้ใจ (untrusted)

การรักษาความปลอดภัยของ netacl ใช้ไอพีแอดเดรส (IP address) และ/หรือชื่อเครื่องคอมพิวเตอร์ (host name) เป็นสิ่งอ้างอิง สำหรับแอพพลิเคชันที่ความปลอดภัยเป็นสิ่งสำคัญสูงสุด (security-critical) จะใช้ไอพีแอดเดรสในการอ้างอิง เพื่อป้องกันการปลอมแปลง (spoof) ต้นทางโดยการใช้โดเมนเนมเซิร์ฟเวอร์ (Domain Name Server) แต่ netacl ไม่สามารถป้องกันการบุกรุกที่ใช้วิธีการปลอมแปลงไอพีแอดเดรสได้ ถ้าต้องการป้องกันการบุกรุกดังกล่าว จะต้องใช้เราเตอร์ที่สามารถตรวจสอบไอพีต้นทางได้ (screening router) และใช้งานร่วมกับ inetd

การออกแบบ netacl เน้นในเรื่องความสะดวกในการตรวจสอบความถูกต้อง โดยใช้โค้ดเป็นภาษา C จำนวน 240 บรรทัด

7.2.3 ftp-gw: เอฟทีพีพร็อกซีเซิร์ฟเวอร์ (A Proxy Server for FTP)

ในการอนุญาตให้มีการโอนย้ายไฟล์ผ่านไฟร์วอลล์ได้โดยไม่เกิดความเสี่ยงต่อความปลอดภัยของระบบ สามารถทำได้ด้วยการใช้งานเอฟทีพีพร็อกซีเซิร์ฟเวอร์ (FTP proxy server) ซึ่งเอฟทีพีพร็อกซีเซิร์ฟเวอร์นี้รองรับการควบคุมการเข้าถึงที่ใช้ไอพีแอดเดรส และ/หรือชื่อโฮสต์เป็นสิ่งอ้างอิง และยังรองรับการควบคุมการใช้งานคำสั่งต่าง ๆ ที่เข้ามาด้วย โดยสามารถเลือกที่จะกัน (block) ไม่ให้บริการ หรือให้บริการแล้วเขียนลงล็อกไฟล์ก็ได้ ส่วนโฮสต์ปลายทางของการติดต่อ (destination) ก็สามารถกำหนดได้เช่นกันว่าจะอนุญาต หรือไม่อนุญาตโฮสต์ใด การติดต่อทั้งหมด (connection) และจำนวนข้อมูลที่ได้ทำการโอนย้ายจะถูกเก็บลงล็อกไฟล์ทั้งหมด

เอฟทีพีพร็อกซีเซิร์ฟเวอร์นี้รองรับการควบคุมการเข้าถึงที่ใช้ไอพีแอดเดรส และ/หรือชื่อโฮสต์เป็นสิ่งอ้างอิง และยังรองรับการควบคุมการใช้งานคำสั่งต่าง ๆ ที่เข้ามาด้วย โดยสามารถเลือกที่จะกัน (block) ไม่ให้บริการ หรือให้บริการแล้วเขียนลงล็อกไฟล์ก็ได้ ส่วนโฮสต์ปลายทางของการติดต่อ (destination) ก็สามารถกำหนดได้เช่นกันว่าจะอนุญาต หรือไม่อนุญาตโฮสต์ใด การติดต่อทั้งหมด (connection) และจำนวนข้อมูลที่ได้ทำการโอนย้ายจะถูกเก็บลงล็อกไฟล์ทั้งหมด

ftp-gw เป็นแอพพลิเคชั่นที่ไม่ก่อให้เกิดความเสี่ยงต่อระบบรักษาความปลอดภัยของระบบไฟร์วอลล์ เนื่องจาก ftp-gw จะทำการ chroot ไปยังไดเร็กทอรีว่างก่อนการทำงาน และไม่มีการติดต่ออ่านเขียนไฟล์ใด ๆ นอกจากไฟล์ที่เกี่ยวกับค่าเริ่มต้น (configuration file) ของตัวเอง เนื่องจากการทำงานของโปรโตคอลเอฟทีพีเป็นสิ่งที่ค่อนข้างยุ่งยาก ดังนั้น ftp-gw จึงประกอบด้วยโค้ดประมาณ 1,300 บรรทัด และมีความยุ่งยากในการตรวจสอบมากกว่าส่วนประกอบซอฟท์แวร์อื่น ๆ ของทูลคิท และ ftp-gw รองรับการทำงานเกี่ยวกับการให้บริการเท่านั้น แต่ไม่ได้สนใจผู้ใช้ที่มาทำการโอนย้ายไฟล์ว่าผ่านการรับรอง (authorized) หรือไม่ ถ้าต้องการการรับรองผู้ใช้ (user authentication) ก็สามารถติดตั้งทูลไว้ที่เกตเวย์ได้ ซึ่งทำให้ผู้ใช้ต้องผ่านขั้นตอนการรับรองก่อนที่จะทำการโอนย้ายไฟล์ ผู้บริหารระบบ

(administrator) สามารถทำการติดตั้งทูลรุ่นใหม่ที่ทำให้สามารถให้บริการเอฟทีพีและเอฟทีพีพร็อกซีในเครื่องเดียวกันได้

7.2.4 telnet-gw: เทลเน็ตพร็อกซีเซิร์ฟเวอร์ (A Proxy Server for TELNET)

ในการอนุญาตให้เครื่องเทอร์มินัลที่อยู่ในระยะไกล (remote terminal) ใช้งานระบบเครือข่ายผ่านไฟร์วอลล์โดยไม่ทำให้เกิดความเสี่ยงต่อระบบรักษาความปลอดภัยของไฟร์วอลล์ สามารถทำได้ด้วยการใช้งานเทลเน็ตพร็อกซี (TELNET proxy) โดยที่เทลเน็ตพร็อกซีเซิร์ฟเวอร์นี้รองรับการควบคุมการเข้าถึงที่ใช้ไอพีแอดเดรส และ/หรือชื่อโฮสต์เป็นสิ่งอ้างอิง และยังรองรับการควบคุมการติดต่อจากโฮสต์ปลายทางใด ๆ (destination) โดยสามารถกำหนดได้ว่าจะอนุญาต หรือไม่อนุญาตโฮสต์ใด การติดต่อทั้งหมด (connection) และจำนวนข้อมูลที่ได้ทำการโอนย้ายจะถูกเก็บลงล็อกไฟล์ทั้งหมด

telnet-gw เป็นแอพพลิเคชั่นที่ไม่ก่อให้เกิดความเสี่ยงต่อระบบรักษาความปลอดภัยของระบบไฟร์วอลล์ เนื่องจาก telnet-gw ทำงานได้โดยไม่ต้องใช้ข้อมูลเกี่ยวกับการยินยอม (permission) และจะทำการ chroot ไปยังไดเร็กทอรีที่กำหนดไว้ก่อนการทำงาน

telnet-gw เขียนขึ้นด้วยโค้ดปรมาณ 1,000 บรรทัด ซึ่งสามารถตรวจสอบได้ง่าย และไม่มีการอ่านเขียนไฟล์ใด ๆ นอกจากไฟล์ที่เกี่ยวกับค่าเริ่มต้น (configuration file) ของตัวเองเท่านั้น ดังนั้น telnet-gw จึงไม่สามารถรองรับการล็อกอินเข้าไปที่ไฟร์วอลล์ได้

7.2.5 rlogin-gw: อาร์ล็อกอินพร็อกซีเซิร์ฟเวอร์ (A Proxy Server for Rlogin)

อาร์ล็อกอินพร็อกซี (rlogin proxy) สามารถรองรับการเข้าถึงจากเครื่องเทอร์มินัล (terminal) ด้วยโปรโตคอลบีเอสดีอาร์ล็อกอิน (BSD rlogin) ได้ โดยอาร์ล็อกอินพร็อกซีสามารถทำการตรวจสอบการยินยอม (permission) และควบคุมการเข้าถึงในลักษณะเดียวกับเทลเน็ตเกตเวย์ เครื่องอาร์ล็อกอินไคลเอนต์สามารถกำหนดโฮสต์ปลายทางที่ต้องการได้ตั้งแต่เริ่มต้นการติดต่อกับพร็อกซี ทำให้ผู้ใช้สามารถผ่านระบบไปได้เลยถ้าไม่มีการตรวจสอบการรับรองผู้ใช้ (user authentication)

7.2.6 plug-gw: การเชื่อมต่อพอร์ตทีซีพีโดยตรง (A TCP Plug-Board Connection Server)

ไฟร์วอลล์มักจะมีการรองรับ (support) บริการบางอย่าง เช่น ยูสเน็ตนิวส์ (Usenet News) โดยผู้บริหารระบบสามารถเลือกที่จะใช้ไฟร์วอลล์เป็นโฮสต์สำหรับให้บริการ หรือติดตั้งพร็อกซีก็ได้ เนื่องจากการให้บริการนิวส์ (news) โดยตรงบนไฟร์วอลล์อาจทำให้ระบบเกิดจุดอ่อนขึ้นได้จากซอฟท์แวร์ของตัวนิวส์ (news) เอง จึงเป็นการปลอดภัยมากกว่าถ้าจะใช้พร็อกซีเป็นเกตเวย์ไปยังโฮสต์ที่ให้บริการและจัดวางโฮสต์ดังกล่าวไว้ในเครือข่ายภายใน

plug-gw เป็นพร็อกซีที่ใช้งานได้กับแอพพลิเคชันหลายตัว โดยจะทำการเชื่อมบริการจากทั้งสองฝั่งเข้าด้วยกันโดยที่ผู้ใช้ไม่ทราบ (transparent) จุดประสงค์หลักของ plug-gw คือเพื่อให้เป็นพร็อกซีที่ใช้งานยูสเน็ตนิวส์ได้ อย่างไรก็ตาม plug-gw ก็สามารถเป็นพร็อกซีของแอพพลิเคชันอื่น ทั่ว ๆ ไปได้

โดยปกติเมื่อติดตั้ง plug-gw ไว้ในไฟร์วอลล์ แล้วมีการติดต่อมาจากโฮสต์ในระยะไกลไปยังพอร์ตเอ็นเอ็นทีพี (NNTP) ไฟร์วอลล์จะทำการติดต่อไปยังโฮสต์ที่ใช้เป็นนิวส์เซิร์ฟเวอร์ที่อยู่ภายในเครือข่ายให้ ถ้าโฮสต์ที่เป็นนิวส์เซิร์ฟเวอร์ภายในเครือข่ายทำการติดต่อไปยังพอร์ตนิวส์ (news) บนไฟร์วอลล์ plug-gw ก็จะทำการเชื่อมการติดต่อทั้งสองฝั่ง (นิวส์เซิร์ฟเวอร์ และเครือข่ายภายนอก) เข้าด้วยกัน ซึ่งการทำงานนี้ใช้ไอพีแอดเดรสของโฮสต์ต้นทางเป็นสิ่งอ้างอิง เมื่อ plug-gw ทำการเชื่อมการติดต่อของทั้งสองฝั่งเข้าด้วยกันแล้วก็จะทำการคัดลอกข้อมูลจนกว่าฝั่งใดฝั่งหนึ่งจะหยุดการติดต่อ เราสามารถปรับตั้งค่า (configure) plug-gw ให้อนุญาตเฉพาะโฮสต์บางตัวได้ด้วยการใช้ไอพีแอดเดรส หรือชื่อโฮสต์เป็นสิ่งอ้างอิง การติดต่อทุกครั้งและจำนวนข้อมูลที่ได้ทำการโอนย้ายจะถูกเก็บบันทึกลงล็อกไฟล์

plug-gw เป็นเสมือนพอร์ตที่เชื่อมระหว่างเครือข่ายภายในกับภายนอก แต่ในการใช้งานต้องทำด้วยความระมัดระวัง และเนื่องจากการทำงานของ plug-gw คล้ายกับเป็นไปป์ (pipe) ไม่มีการอ่านเขียนดิสก์ และไม่มีการเรียกใช้งานเชลล์ (shell) หรือโพรเซสใด ๆ การใช้งาน plug-gw เป็นพร็อกซีสำหรับนิวส์ ตัว plug-gw สามารถรักษาความปลอดภัยได้เป็นอย่างดี เพราะการติดต่อจากภายนอกสามารถทำได้แบบสองทิศทาง (bi-directional) ด้วยการใช้พอร์ตเพียงพอร์ตเดียวบนนิวส์เซิร์ฟเวอร์ภายในเครือข่าย และในขณะเดียวกันก็สามารถกัน (block) การติดต่ออื่น ๆ ได้

7.2.7 ส่วนที่สามารถเพิ่มเติมได้ในภายหลัง (Optional Components)

7.2.7.1 authd: การรับรองผู้ใช้ของเครือข่าย (Network Authentication Service)

ออเธนทิเคชันเซิร์ฟเวอร์ของระบบเครือข่าย (network authentication server) ทำงานด้วยโปรแกรม authd ซึ่งให้บริการเกี่ยวกับการรับรองผู้ใช้ (authentication service) ที่ต้องการใช้งานแอพพลิเคชันต่าง ๆ ของระบบเครือข่าย แต่โปรแกรมนี้เป็นออพชัน ซึ่งจะใช้เมื่อมีการใช้งานไฟร์วอลล์พร็อกซี เช่น ftp-gw หรือ tn-gw และทำการปรับตั้งค่าให้มีการเรียกใช้การรับรอง จุดประสงค์ของ authd คือการรวบรวมวิธีการรับรองผู้ใช้แบบต่าง ๆ ที่มีใช้งานอยู่ในองค์กร (โดยเฉพาะในองค์กรขนาดใหญ่) มาไว้ภายในฐานข้อมูล (database) เดียว และมีเชลล์สำหรับผู้บริหารระบบให้สามารถจัดการฐานข้อมูลผ่านระบบเครือข่ายได้ รวมทั้งมีการเข้ารหัส (encryption) ข้อมูลต่าง ๆ

ฐานข้อมูลที่ authd ใช้มีการแบ่งกลุ่มผู้ใช้ออกเป็นกลุ่มของผู้บริหารระบบ (administrator group) ของกลุ่มหนึ่ง ๆ และกลุ่มของผู้ใช้ (user group) ภายในกลุ่มนั้น ๆ โดยที่ผู้ที่อยู่ในกลุ่มของผู้บริหารระบบจะสามารถเพิ่ม , ลบ และแก้ไขข้อมูลผู้ใช้ภายในกลุ่มนั้น ๆ ได้ authd ยังเก็บข้อมูลเกี่ยวกับเวลาที่ผู้ใช้ผ่านเข้ามาทำการรับรองกับโปรแกรมครั้งสุดท้าย , จำนวนครั้งของการทำการรับรองแล้วไม่สามารถผ่านการรับรองได้ (failed attemp) และสามารถยกเลิกแอคเคาท์นั้น ๆ (disable account) หลังจากการเกิดความผิดพลาดมากครั้งกว่าที่ตั้งไว้ authd มีการสร้างล็อกไฟล์พิเศษไว้เก็บทรานแซ็คชัน (transaction) ต่าง ๆ การทำงานของ authd จะต้องอยู่บนโฮสต์ที่มีความปลอดภัยเพียงพอ เพราะอาจเกิดการบุกรุกเพื่อข้อมูลที่มีอยู่ในฐานข้อมูลได้

7.2.7.2 telnetd: บริการล็อกอินผ่านเครือข่ายมายังไฟร์วอลล์ (Network Login Service to the Firewall)

ด้วยเหตุผลบางประการ ในบางครั้งผู้บริหารระบบอาจต้องการล็อกอินเข้ามาที่โฮสต์ไฟร์วอลล์โดยตรง เพื่อจะทำการปรับปรุงแก้ไขค่าต่าง ๆ เราสามารถปรับตั้งค่าโพรเซสที่เทลเน็ตเซิร์ฟเวอร์ ที่ชื่อ telnetd ให้ทำงานอยู่บนไฟร์วอลล์ได้เพื่อให้สามารถทำการล็อกอินเข้ามาที่ไฟร์วอลล์ได้ ค่ามาตรฐานที่เป็นอยู่คือเทลเน็ตพร็อกซีเซิร์ฟเวอร์ telnet-gw ทำงานอยู่บนพอร์ต 23 ซึ่งเป็นพอร์ตทีซีพี (TCP) ที่ให้บริการเทลเน็ต และผู้บริหารระบบจะต้องล็อกอินที่คอนโซล (console) เราสามารถปรับตั้งค่า telnetd ในไฟล์ inetd.conf เพื่อให้สามารถล็อกอินได้ และสามารถรักษาความปลอดภัยของ telnetd ได้ด้วย netacl รวมทั้งจำกัดจำนวนของโฮสต์ที่สามารถติดต่อขอใช้บริการได้ในขณะหนึ่ง ในการทำงานขอแนะนำให้ใช้รหัสผ่านแบบใช้ครั้งเดียว (one-time changing password)

7.2.7.3 login: การรับรองผู้ใช้ (User Authentication)

ทูลคิทได้รวมเอาโปรแกรมล็อกอินที่ชื่อว่า login-sh ไว้เพื่อการรับรองความถูกต้องของผู้ใช้ ซึ่งรองรับวิธีการรับรองผู้ใช้แบบต่าง ๆ เช่น Security Dynamics' SecurID , Digital Pathways' Secure Net Key และ Racal Watchwood ในการทำงานจะใช้โปรแกรมล็อกอินนี้แทนเชลล์ปกติของผู้ใช้ และผู้ใช้จะต้องผ่านการรับรองก่อนที่จะเรียกใช้คอมมานด์อินเทอร์พรีเตอร์ (command interpreter) ได้ วิธีนี้ทำให้การล็อกอินมีความปลอดภัย โดยที่ไม่ต้องแก้ไขโปรแกรมล็อกอินเดิม

7.2.7.4 ftpd: บริการเอฟทีพีสำหรับผู้ใช้ที่ใช้ชื่อว่า anonoymous (Anonymous FTP Service)

ตามปกติแล้ว ftpd จะยอมให้ผู้ใช้ทำการล็อกอินด้วยชื่อ anonymous หรือ ftp เพื่อเข้ามาใช้งานเป็นเกสต์ (guest) ของระบบ ซึ่งในกรณีนี้ผู้ใช้จะต้องถูก chroot ไปที่โฮมไดเร็กทอรี (home directory) ของเอฟทีพีซึ่งเราสามารถใช้ netacl ในการควบคุมและพิจารณาการร้องขอ (request) ว่ามาจากภายนอกเครือข่ายหรือไม่ ถ้าเป็นการร้องขอจากภายนอก ftpd จะถูกปลุกขึ้นมาหลังจากที่ทำการ chroot ไปยังไดเร็กทอรีที่กำหนดไว้เรียบร้อยแล้ว เป็นการเพิ่มความปลอดภัยจากจุดบอด (bug) ที่มีอยู่ในโปรแกรม ftpd และจะสามารถป้องกันเครือข่ายภายนอกที่ไม่น่าเชื่อถือ (untrusted network) ไม่ให้สามารถติดต่อโดยตรงกับแอพพลิเคชันที่ทำงานอยู่บนระบบไฟล์ (file system) ที่ไม่ได้ทำการจำกัดไว้ (unrestricted) แต่การปรับตั้งค่าต่าง ๆ ต้องทำด้วยความรอบคอบ ถึงแม้ว่าการรวมเอา ftpd มาใช้งานร่วมกับ netacl จะไม่ต้องเปลี่ยนแปลงมาตรฐานการใช้งานเดิมก็ตาม

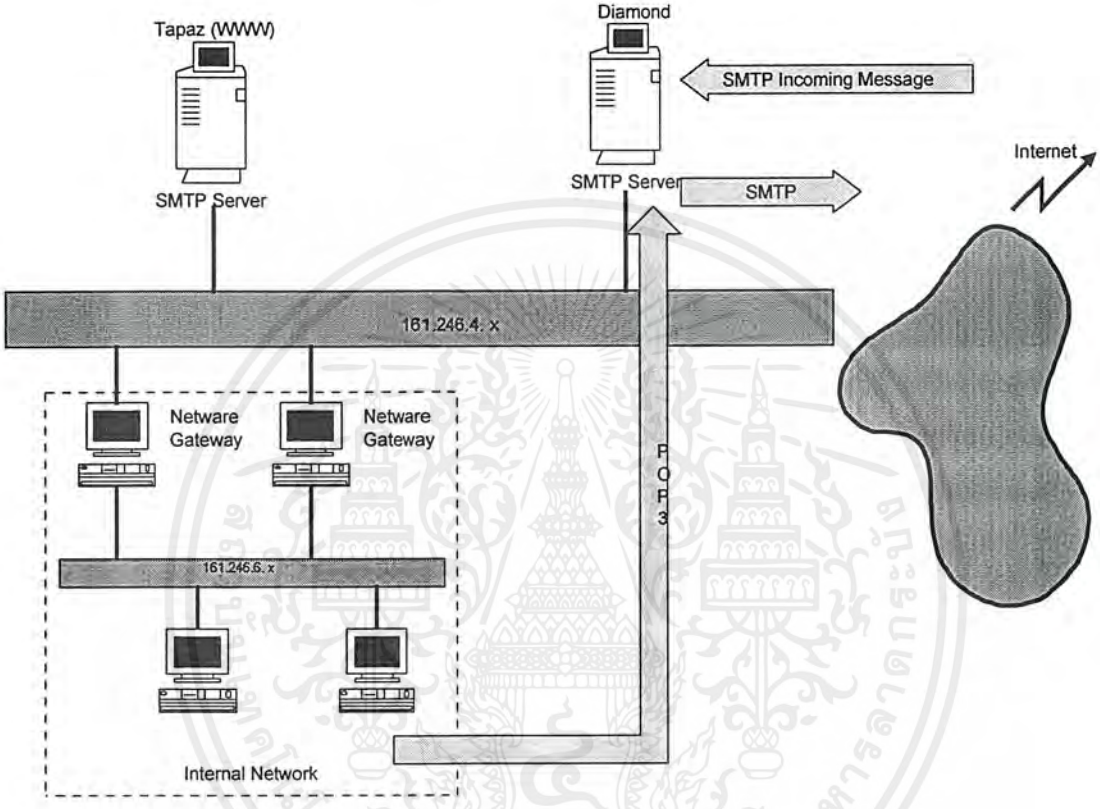
7.2.7.5 syslogd: ล็อกไฟล์ของระบบ (System Logging)

ทูลคิทของไฟร์วอลล์จะมีเดมอนที่ใช้ในการเก็บล็อกไฟล์ (logging daemon) ซึ่งยอมให้มีการค้นหารูปแบบเฉพาะ (regular expression search) ในไฟล์ และสามารถปลุกโปรแกรมใด ๆ ตามที่ระบุไว้ให้ขึ้นมาทำงาน เมื่อได้รับข้อมูลล็อกในรูปแบบที่ระบุไว้ และยังยอมให้มีการตรวจสอบล็อกไฟล์ของระบบในขณะนั้นทันที (real-time scanning) และแจ้งเตือนในขณะนั้นทันที (real-time alert) การปลุกโพรเซสใด ๆ ให้ขึ้นมาทำงานโดยใช้ข้อมูลล็อกเป็นสิ่งอ้างอิงนับเป็นวิธีที่ดีสำหรับผู้ดูแลระบบในการจัดการเมื่อมี

เหตุการณ์ใด ๆ ที่เป็นอันตรายต่อความปลอดภัยของระบบเครือข่ายเกิดขึ้น โดยอาจกำหนดให้ทำการปิดเครื่อง หรือส่งเมล์ไปยังผู้ดูแลระบบ เป็นต้น

7.3 การดำเนินการไฟร์วอลล์ตามบริการต่าง ๆ

7.3.1 บริการเอสเอ็มทีพี (SMTP Service)



รูป 7.3 ระบบการรับส่งเมล์ก่อนการติดตั้งพร็อกซี

ลักษณะโดยทั่วไปของการส่งเมล์ผ่านโปรโตคอลเอสเอ็มทีพีโดยไม่ผ่านไฟร์วอลล์

- เอสเอ็มทีพีเซิร์ฟเวอร์ ในที่นี้คือโฮสต์ไดมอนด์ (diamond) รับเมสเสจ (message) จากเครือข่าย (network) ภายนอก หรือภายในวงเดียวกัน แล้วส่งเมสเสจดังกล่าวสู่เอสเอ็มทีพีเซิร์ฟเวอร์ปลายทาง
- โพรเซสที่ทำหน้าที่รับส่งเมสเสจ (delivery agent) รับเมสเสจ แล้วทำการเขียนข้อมูลลงเมล์บ็อกซ์ (mailbox) ของผู้ใช้ที่ถูกระบุภายในเมสเสจ
- โพรเซสที่ทำหน้าที่ส่งเมสเสจไปยังผู้ใช้ (user agent) ทำการอ่านเมสเสจ โดยจะอ่านผ่านโพสต์ออฟฟิสโปรโตคอลเวอร์ชั่น 3 (POP3) แล้วจะตอบรับเมสเสจดังกล่าวผ่านโพสต์ออฟฟิสโปรโตคอลเวอร์ชั่น 3 เช่นเดิม

ด้วยวิธีการเช่นนี้มีข้อเสียคือ

- โฮสต์เอสเอ็มทีพีเซิร์ฟเวอร์ไม่สามารถตรวจสอบเมสเสจที่รับเข้ามาได้ โดยเอสเอ็มทีพีเซิร์ฟเวอร์จะเขียนข้อมูลลงสพูลไดเร็กทอรีทันที ด้วยวิธีการนี้ ผู้บุกรุก (hacker) สามารถส่งเมสเสจเป็นจำนวนเท่าใดก็ได้สู่เอสเอ็มทีพีเซิร์ฟเวอร์ปลายทาง จนเมล์บ๊อกซ์เต็ม และไม่สามารถใช้งานต่อได้
- โพรเซสที่ทำหน้าที่รับข้อมูลจำเป็นต้องได้รับการยินยอม (permission) ที่จะสามารถเขียนเมสเสจที่รับมาได้สู่เมล์บ๊อกซ์ของผู้ใช้ทุกคนที่ถูกระบุในเมสเสจ
- ผู้ใช้ทุกคนที่อยู่ภายในระบบมีสิทธิเต็มที่ในการใช้งานเอสเอ็มทีพีเซิร์ฟเวอร์ (คือโฮสต์ไดมอนด์) นั่นคือสามารถส่งข้อมูลที่มีขนาดใหญ่ได้เต็มที่ โดยไม่มีการจำกัดขอบเขตข้อมูล

โดยทั่วไป หน้าที่ในการรับส่งเมสเสจโดยใช้โปรโตคอลเอสเอ็มทีพีโปรแกรมตัวหนึ่งซึ่งได้รับความนิยมอย่างมากคือ sendmail แต่ข้อเสียของโปรแกรม sendmail มีดังนี้คือ

- ทำงานที่พอร์ต 25 ซึ่งเป็นพอร์ตสงวนสำหรับเมสเสจที่เข้าสู่ระบบ (incoming SMTP connection)
- เนื่องจาก sendmail เป็นได้ทั้งโพรเซสที่ทำหน้าที่ส่งเมสเสจไปยังผู้ใช้ (user agent) และโพรเซสที่ทำหน้าที่รับส่งเมสเสจ (delivery agent) ดังนั้น ในขณะที่ทำงานเป็นโพรเซสที่ทำหน้าที่ส่งเมสเสจไปยังผู้ใช้ โปรแกรมจำเป็นจะต้องเปลี่ยนรหัสของโพรเซส (process id) เป็นพรีวิเลจ (previlege) เดียวกับผู้ใช้ที่ใช้งานอยู่ในขณะนั้น
- ขณะที่ sendmail ทำงานเป็นโพรเซสที่ทำหน้าที่รับส่งเมสเสจ จะต้องเปลี่ยนพรีวิเลจตัวเองเป็นซูเปอร์ยูสเซอร์ (superuser) เพื่อให้สามารถทำงานได้กับทุกฟังก์ชั่นที่มีอยู่ในเคอร์เนล (kernel)

ในการใช้งานเมล์ในองค์กรหนึ่ง ๆ เช่น ภาควิชาวิศวกรรมคอมพิวเตอร์ จะมีเมล์เซิร์ฟเวอร์เป็นเซิร์ฟเวอร์หลักในการรับส่งเมล์ บุคคลภายนอกที่จะส่งเมล์มายังภาควิชา ฯ จะต้องระบุด้วยชื่อโดเมน (domain name) ของภาควิชา นั่นคือ ce.kmitl.ac.th ตัวอย่างเช่น s7014010@ce.kmitl.ac.th รีโซลเวอร์ (resolver) ที่ต้นทางจะถามโดเมนเนมเซิร์ฟเวอร์ (DNS) ปลายทางว่า โดเมนดังกล่าวมีเมล์เซิร์ฟเวอร์ตัวใดเป็นตัวรองรับการรับส่งเมล์ ซึ่งในที่นี้ โดเมนของภาควิชาวิศวกรรมคอมพิวเตอร์ มีโฮสต์ที่ชื่อไดมอนด์ (diamond) เป็นตัวรองรับการรับส่งเมล์ภายในภาควิชา ซึ่งถูกระบุในฐานข้อมูลโดเมนเนมเซิร์ฟเวอร์ (DNS database file) ดังนี้

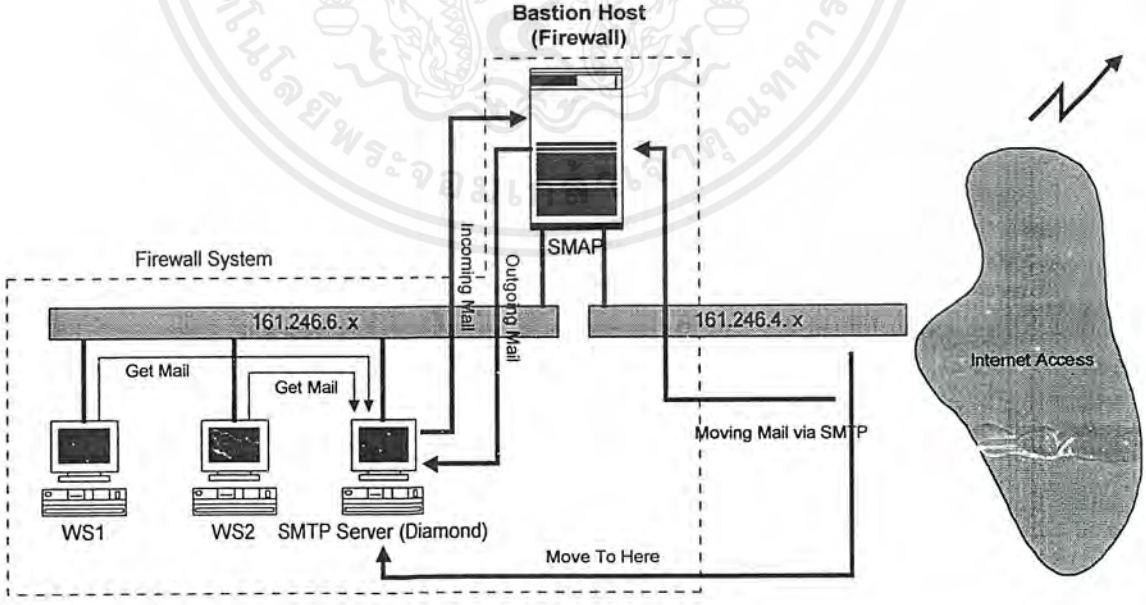
```
@    IN    SOA    diamond.ce.kmitl.ac.th. root.diamond.ce.kmitl.ac.th. (
                        19980206; Serial (yyyymmdd)
                        10800   ; Refresh every 3 hours
                        3600    ; Retry every hour
                        604800  ; Expire after a week
                        86400 ) ; Minimum ttl of 1 day
     IN    NS     diamond.ce.kmitl.ac.th.
$ORIGIN kmitl.ac.th.
ce        IN    MX    10    diamond.ce.kmitl.ac.th.
```

← แสดงว่าโดเมนของภาควิชา ฯ ถูกกำหนดให้มีโฮสต์ไดมอนด์เป็นโฮสต์รับส่งเมล์

จากฐานข้อมูลโดเมนเนมเซิร์ฟเวอร์ที่นำมาแสดงส่วนหนึ่ง เป็นการบังคับให้โฮสต์ต้นทางที่ต้องการรับส่งเมล์กับภาควิชาวิศวกรรมคอมพิวเตอร์ ให้ไปรับส่งเมล์กับโฮสต์ที่ชื่อไดมอนด์แทน โดยเรคคอร์ด MX (Mail Exchange) เป็นตัวบอกทิศทางของเมล์ที่จะรับส่งด้วย

7.3.1.1 smap และ smapd

โครงงานนี้ใช้ smap และ smapd แทน sendmail ที่เคยใช้อยู่ จากเครือข่ายต้นแบบของภาควิชาวิศวกรรมคอมพิวเตอร์ ประกอบด้วยไคลเอนต์ที่มีโพสต์ออฟฟิสโปรโตคอลเวอร์ชั่น 3 (POP3) เป็นโปรโตคอลติดต่อกับโฮสต์ที่ชื่อไดมอนด์ โดยที่ smap จะทำหน้าที่รับเมล์ แล้วเขียนไปยังสพูลไดเร็กทอรี จากนั้น smapd จะนำข้อมูลที่มีอยู่ในสพูลไดเร็กทอรีเขียนไปยังเมล์บ๊อกซ์ของผู้ใช้อีกทีหนึ่ง



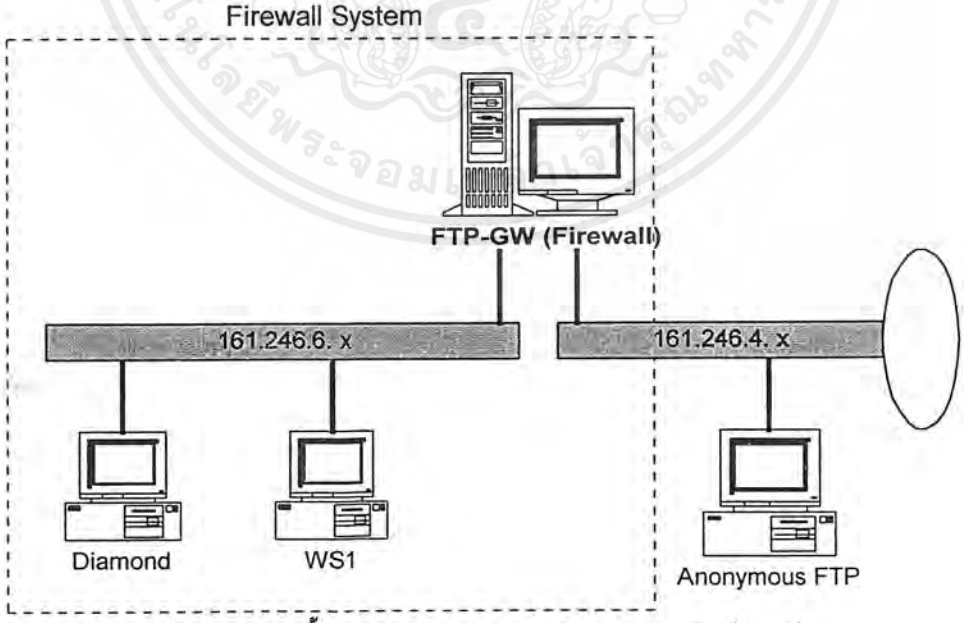
รูป 7.4 ระบบการรับส่งเมล์หลังการติดตั้งพร็อกซี

เมล์บ๊อกซ์ของผู้ใช้ในที่นี้จะหมายถึง โฮสต์ที่มีโพรเซส sendmail ทำงานอยู่ จากเครือข่ายของภาควิชาวิศวกรรมคอมพิวเตอร์ จะมีโฮสต์ไดมอนด์ที่มีโพรเซส sendmail ทำงานอยู่ ดังนั้น เราไม่จำเป็นต้องเปลี่ยนแปลงค่าที่เคยติดตั้ง (configure) ของ sendmail อีก ผู้ใช้ที่เคยใช้โฮสต์ไดมอนด์เป็นตัวส่งเมล์ ก็จะยังคงใช้ได้ตามปกติ

ข้อสังเกต : ผู้ใช้ที่จะใช้โฮสต์ไดมอนด์ เป็นตัวรับส่งเมล์ จะต้องระบุต้นทาง (นั่นคือที่อยู่ของเราเอง) ให้ชี้ไปที่โดเมนของภาควิชาวิศวกรรมคอมพิวเตอร์ (ตัวอย่างเช่น user@ce.kmitl.ac.th) เนื่องจากเรคคอร์ดที่เก็บ MX (Mail Exchange) ถูกชี้มายังแบสชันโฮสต์ (bastion host) แล้ว ดังนี้

```
@   IN   SOA   diamond.ce.kmitl.ac.th. root.diamond.ce.kmitl.ac.th. (
                19980206; Serial (yyyymmdd)
                10800  ; Refresh every 3 hours
                3600   ; Retry every hour
                604800 ; Expire after a week
                86400 ) ; Minimum ttl of 1 day
    IN   NS    diamond.ce.kmitl.ac.th.
$ORIGIN kmitl.ac.th.
ce          IN   MX   10   firewall.ce.kmitl.ac.th.
```

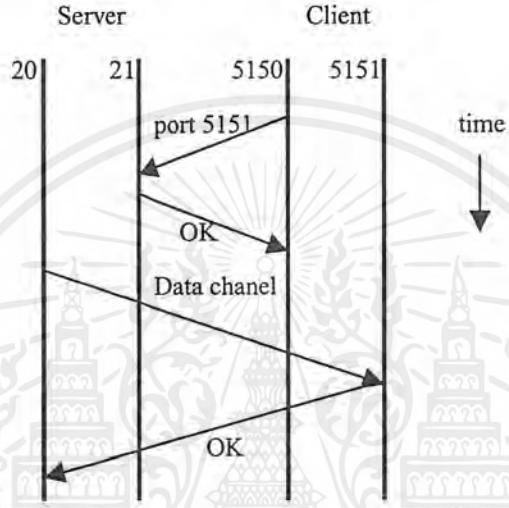
7.3.2 บริการเอฟทีพี (FTP Service)



รูป 7.5 การติดตั้ง Anonymous FTP ภายนอกระบบไฟร์วอลล์

7.3.2.1 ลักษณะการทำงานของเอฟทีพีในระบบที่ไม่มีไฟร์วอลล์

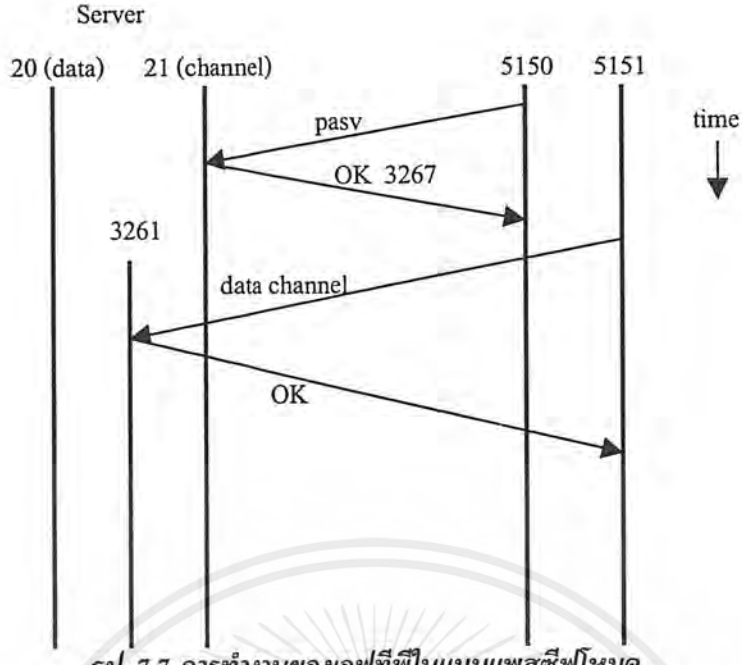
โดยทั่วไปการโอนย้ายไฟล์ข้อมูลจากโฮสต์หนึ่งไปสู่อีกโฮสต์หนึ่งนั้น เราจะอาศัยโปรโตคอลเอฟทีพีการทำงานจะมีการทำงานใน 2 โหมดด้วยกัน คือ นอร์มัลโหมด (Normal Mode) และแพสซีฟโหมด (Passive Mode) โดยในโหมดแรก คือ นอร์มัลโหมด จะอาศัยพอร์ตของทีซีพีพอร์ตที่ 20 และ 21 ในส่วนที่เป็นเซิร์ฟเวอร์โพรเซส เช่นกันกับฝั่งไคลเอนต์ ซึ่งจะอาศัยพอร์ตทีซีพีในการติดต่อกับเซิร์ฟเวอร์ โดยพอร์ตที่ใช้จะเป็นพอร์ตที่มีค่ามากกว่า 1,024 ซึ่งใช้ค่าที่สุ่มได้จำนวน 2 ค่า จากค่าพอร์ตที่กำหนด ตัวอย่างการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยใช้เอฟทีพีในแบบนอร์มัลโหมดมีดังนี้



รูป 7.6 การทำงานของเอฟทีพีในแบบนอร์มัลโหมด

การส่งในแบบนอร์มัลโหมด ในขั้นแรก ทางฝั่งโฮสต์ไคลเอนต์จะส่งหมายเลขพอร์ตที่ใช้เป็นคอมมานด์พอร์ต (command port) ให้กับฝั่งเซิร์ฟเวอร์ และฝั่งเซิร์ฟเวอร์ก็จะทำการตอบ (acknowledge) ให้ฝั่งไคลเอนต์ทราบ และจะเริ่มทำการส่งข้อมูลโดยใช้พอร์ต 20 ของฝั่งเซิร์ฟเวอร์ และพอร์ตที่ถัดขึ้นไปจากคอมมานด์พอร์ตของฝั่งไคลเอนต์เป็นช่องทางในการส่งข้อมูลต่อไป

แต่ในแบบแพสซีฟโหมด จะอาศัยหลักการที่ว่า ในขณะที่พอร์ต 20 ซึ่งเป็นพอร์ตในการส่งข้อมูลในฝั่งเซิร์ฟเวอร์ถูกใช้อยู่ โฮสต์ไคลเอนต์อื่นก็ไม่สามารถใช้พอร์ตดังกล่าวได้ ดังนั้น จึงได้ใช้วิธีการใหม่คือ เสริมให้มีพอร์ตขึ้นมาโดยไม่จำเป็นต้องเริ่มต้นทำการขอการติดต่อ (request for connecting) เลย แต่จะเป็นการส่งคอมมานด์ PASV ไปยังเซิร์ฟเวอร์ แล้วคอยการตอบรับตัวเลขพอร์ตที่เซิร์ฟเวอร์ว่างพอที่จะติดต่อด้วย ดังรูป



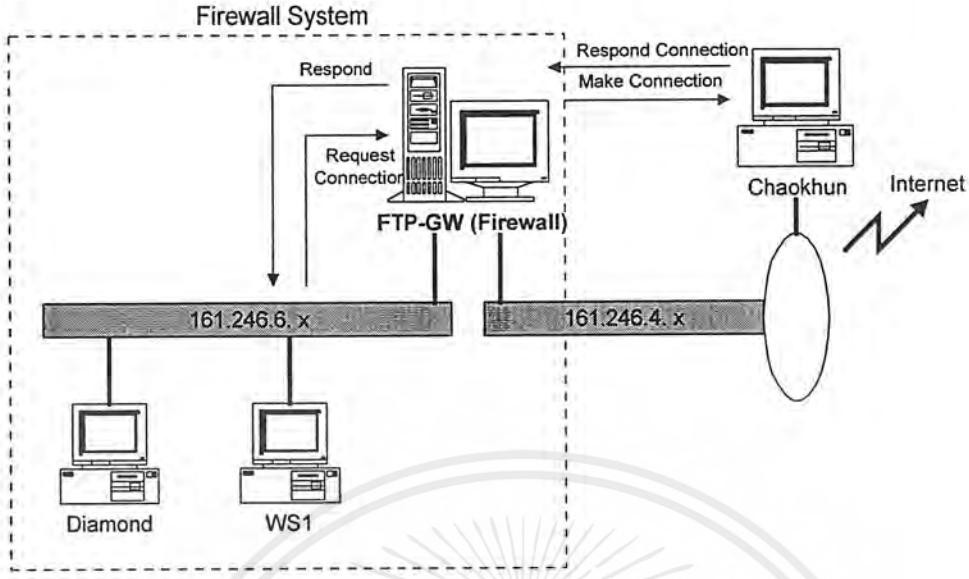
รูป 7.7 การทำงานของเอฟทีพีในแบบแพสซีฟโหมด

จากรูป เมื่อไคลเอนต์ร้องขอโหมดการส่งด้วย PASV ทางด้านเซิร์ฟเวอร์จะตอบสนองด้วยพอร์ตที่ตนเองคิดว่าว่างที่จะใช้ในการส่งข้อมูล (data) คือพอร์ต 3,261 เมื่อไคลเอนต์ทราบก็จะใช้พอร์ต 5,151 และพอร์ต 3,261 เป็นช่องทางติดต่อแทนพอร์ต 20 โปรแกรมที่สนับสนุนการทำงานแบบแพสซีฟโหมดได้แก่โปรแกรมเน็ตสเคป (Netscape) , โมเสค (MOSAIC) หรือดับเบิ้ลยูเอส-เอฟทีพี (WS-FTP) เป็นต้น

โอเปอร์เรชันที่เกิดขึ้นระหว่างการติดต่อในทั้งสองแบบ เมื่อดูแล้วจะพบว่าไม่ได้มีการควบคุมเลย นั่นคือว่าโอเปอเรชันที่เกิดระหว่างพอร์ตที่ใช้รับส่งข้อมูล (data port) หรือคอมมานด์พอร์ต (command port) เราสามารถควบคุมการเข้าถึงข้อมูลในฝั่งเซิร์ฟเวอร์ได้ โดยส่วนใหญ่จะควบคุมได้ใน 2 ระดับด้วยกัน คือ

- ระดับแพ็กเก็ตข้อมูล (proxy server)
- ระดับผู้ใช้ (anonymous login)

7.3.2.2 เอฟทีพีที่มีการทำพร็อกซี (proxy FTP)



รูป 7.8 การติดต่อระหว่างเอฟทีพีพร็อกซีกับเอฟทีพีไคลเอนต์

จากรูป แสดงบางส่วนของระบบเครือข่ายของภาควิชาวิศวกรรมคอมพิวเตอร์ ในส่วนของการติดตั้งพร็อกซีเซิร์ฟเวอร์ ผู้ใช้ทั้งหมดที่จะทำการโอนย้ายไฟล์ภายในระบบไฟร์วอลล์ สามารถกระทำได้โดยตรง เช่น เครื่องเวิร์คสเตชั่น 1 (WS1) ติดต่อกับโฮสต์ไดมอนด์ (diamond) แต่จะเกิดกรณีที่เครื่องเวิร์คสเตชัน (workstation) ที่อยู่ภายในระบบไฟร์วอลล์ต้องการโอนย้ายไฟล์กับโฮสต์ที่ไม่ได้อยู่ในเครือข่าย ซึ่งในกรณีเช่นนี้ จะต้องทำการเอฟทีพีไปที่เอฟทีพีพร็อกซีเซิร์ฟเวอร์ ก่อนที่จะทำการติดต่อ (make connection) ไปยังโฮสต์อื่น ๆ ที่อยู่นอกเหนือจากโฮสต์ภายในระบบไฟร์วอลล์ ตัวอย่างเช่น เครื่องเวิร์คสเตชั่น 1 (WS1) ต้องการติดต่อกับโฮสต์ที่ชื่อ chaokhun.kmitl.ac.th ขั้นแรกจะต้องใช้คำสั่งเอฟทีพีไปยังเอฟทีพีพร็อกซีเซิร์ฟเวอร์ก่อน ถ้าพร็อกซีเซิร์ฟเวอร์อนุญาตให้โฮสต์ของเราใช้เอฟทีพีพร็อกซีเซิร์ฟเวอร์ ก็จะแสดงเมนูเพื่อให้ผู้ใช้พิมพ์โฮสต์ปลายทางที่ต้องการจะติดต่อด้วย

```
WS1> ftp firewall

                    Welcome to Firewall System

FTP-GW> c s7014010@chaokhun.kmitl.ac.th

Connecting host …
```

7.3.2.2.1 ขั้นตอนการติดตั้งโปรแกรมเอฟทีพีพร็อกซี

ภายในตัวเอฟทีพีพร็อกซีเซิร์ฟเวอร์ จะมีการใช้โพรเซสที่ชื่อ ftp-gw รออยู่ที่พอร์ต 21 ซึ่งปกติแล้วพอร์ตดังกล่าวคือพอร์ตเฉพาะของเอฟทีพี (well-known port) ที่มีโพรเซส ftpd รออยู่ และจะถูกเรียกให้ขึ้นมาทำงานจากโพรเซส /sbin/inetd

- ที่เอฟทีพีพร็อกซีเซิร์ฟเวอร์
  - การติดตั้ง ftp-gw จะใช้วิธีการแทนที่โปรแกรม ftpd ด้วยโปรแกรม ftp-gw โดยทั่วไปจะแก้ไขที่ไฟล์ /etc/inetd.conf โดยทำการแก้ไขบรรทัดที่มี

    ftp stream tcp nowait root /usr/sbin/tcpd wu.ftpd

    ไปเป็น

    ftp stream tcp nowait root /usr/local/etc/ftp-gw ftp-gw
  - ค่าเริ่มต้น (default) การทำงานจะอยู่ที่ไฟล์ /usr/local/etc/netperm-table ตัวอย่างของไฟล์ /usr/local/etc/netperm-table ที่ได้ทำการแก้ไขเพื่อให้ใช้งานได้ ดังนี้

    ftp-gw: permit-hosts 161.246.6.*

    ftp-gw: permit-hosts 161.246.5.*

    เป็นการบอกว่า จะยินยอมให้โฮสต์เฉพาะที่อยู่ภายในเครือข่ายวง 161.246.6.X และ 161.246.5.X เท่านั้นที่สามารถใช้งานเอฟทีพีพร็อกซีได้ ถ้าโฮสต์ที่มาจากเครือข่ายวงอื่น ๆ นอกเหนือจากที่ระบุไว้ในกฎดังกล่าวแล้ว จะไม่สามารถผ่านเข้ามาใช้งานเอฟทีพีพร็อกซีได้ เช่น โฮสต์ที่มาจากเครือข่ายวง 161.246.4.X เป็นต้น
  - เมื่อเพิ่มข้อมูลตามที่ได้กำหนดไว้เรียบร้อยแล้ว จะต้องทำการส่งสัญญาณชื่อ SIGHUP ไปให้กับโพรเซส inetd โดยใช้คำสั่งดังนี้

    kill –HUP `ps –ef | grep inetd | cut –f 2`
- ที่โฮสต์ไคลเอนต์

  เราสามารถใช้โปรแกรมเอฟทีพีที่มีอยู่เดิมกับเอฟทีพีพร็อกซีเซิร์ฟเวอร์ได้ โดยไม่ต้องมีการแก้ไขใด ๆ ทั้งสิ้น

7.3.2.2.2 รูปแบบของคำสั่งที่ใช้

ftp-gw: [*รูปแบบ*]

โดยที่สามารถกำหนด *รูปแบบ* ได้ดังต่อไปนี้

- userid *user*

  ระบุเป็นรหัสผู้ใช้ (user-id) หรือชื่อที่อยู่ในไฟล์รหัสผ่าน (password file) ถ้ามีการใช้งานออพชั่นนี้ โปรแกรม ftp-gw จะทำการเปลี่ยนรหัสผู้ใช้ไปเป็นรหัสที่กำหนดไว้ ก่อนที่จะเริ่มทำงาน
- directory *pathname*

  เป็นไดเร็กทอรีที่โปรแกรมจะทำการ chroot ไปก่อนจะเริ่มทำงาน

- denial-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดงให้ผู้ใช้ทราบ ถ้าผู้ใช้คนนั้นไม่มีสิทธิในการใช้งานพร็อกซี ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ (default)

- welcome-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดงให้ผู้ใช้เมื่อการติดต่อเป็นผลสำเร็จ (successful) ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- help-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดง เมื่อผู้ใช้ใช้คำสั่ง help ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- denydest-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดง เมื่อผู้ใช้ทำการติดต่อไปยังโฮสต์ที่ระบบไม่อนุญาต ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- timeout *secondvalue*

  ถ้าไม่มีการป้อนข้อมูลหลังจากระยะเวลาเท่ากับ *secondvalue* วินาที จะทำการปิดการติดต่อ (disconnect) ค่าปกติคือ 60 นาที

- permit-hosts *host-pattern* [*host-pattern*...] [*options*]

  deny-hosts *host-pattern* [*host-pattern*...] [*options*]

  เป็นการระบุโฮสต์ และการยินยอมให้ใช้งาน (access permission) โดยสามารถใช้รูปแบบของพารามิเตอร์ได้ดังนี้

  - *host-pattern*
    - ระบุเป็นหมายเลขของเครือข่ายได้ เช่น 161.246.4.*
    - ระบุเป็นไอพีแอดเดรสของโฮสต์ได้ เช่น 161.246.4.3
    - ระบุเป็นชื่อโฮสต์ เช่น diamond.ce.kmitl.ac.th

    ตามรูปแบบของคำสั่ง จะเห็นได้ว่าสามารถระบุโฮสต์ได้มากกว่าหนึ่ง

  - *options*
    - -log *operation*

      -log { *operation* [*operation*...] }

      ระบุว่าให้ระบบทำการเก็บข้อมูลลงล็อกไฟล์ เมื่อมีการทำ *operation* ผ่านพร็อกซี
    - -authall

      ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งานโอเปอเรชั่นใด ๆ ได้
    - -auth *operation*

      -auth { *operation* [*operation*] }

      ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งาน *operation* ที่กำหนดไว้ได้

ข้อดีของการทำเอฟทีพีพร็อกซีเซิร์ฟเวอร์

- สามารถควบคุมการเข้าออกของโฮสต์ในเครือข่ายที่กำหนดได้ ในที่นี้จะมีการแจ้งผลการเข้าออก , ขนาดของข้อมูลที่ได้ทำการโอนย้าย , อัตราการใช้พร็อกซีทั้งที่สำเร็จ (success) และล้มเหลว (fail) จากโฮสต์ที่ไม่ได้รับอนุญาตให้ใช้ (deny) ด้วย
- เราสามารถใช้งานเอฟทีพีพร็อกซีเซิร์ฟเวอร์ร่วมกับการทำการรับรองผู้ใช้ (authentication) ได้ ในที่นี้จะเป็นออพชันที่เราสามารถเลือกได้ โดยการปรับตั้งค่าที่ไฟล์ /usr/local/etc/netperm-table
- ป้องกันการบุกรุกเข้ามาใช้ไฟล์จากโฮสต์ที่มาจากเครือข่ายที่ไม่น่าไว้วางใจ (untrusted network) ได้

ข้อเสียของการทำเอฟทีพีพร็อกซีเซิร์ฟเวอร์

- เมื่อเทียบความเร็วระหว่างการเชื่อมต่อโดยตรงโดยไม่ผ่านเอฟทีพีพร็อกซี กับแบบที่ต้องผ่านเอฟทีพีพร็อกซี แล้ว แบบไม่ผ่านจะให้อัตราการโอนย้ายข้อมูลได้สูงกว่า เนื่องจากไม่มีการพักข้อมูลไว้ที่เอฟทีพีพร็อกซี
- เพื่อเพิ่มความสะดวกในขณะใช้งานเอฟทีพีพร็อกซี ก็จะต้องมีโปรแกรมสนับสนุนการทำงานผ่านไฟร์วอลล์ ตัวอย่างเช่น ดับเบิ้ลยู-เอฟทีพี (WS-FTP) , เน็ตสเคป (Netscape) หรือโมเสค (MOSAIC) เป็นต้น

7.3.2.3 การทำเอฟทีพีโดยผู้ใช้ที่ใช้ชื่อว่า anonymous (Anonymous FTP)

เนื่องจากการโอนย้ายข้อมูลโดยใช้เอฟทีพีนั้น โดยทั่วไปจะมีลักษณะการโอนย้ายของผู้ใช้อยู่ 2 ประเภท คือ โอนย้ายโดยที่ผู้โอนย้ายที่ชื่อ (account) อยู่ในระบบ กับอีกประเภทหนึ่งคือการโอนย้ายข้อมูลโดยเราไม่ทราบว่าผู้ใช้เป็นใคร (anonymous)

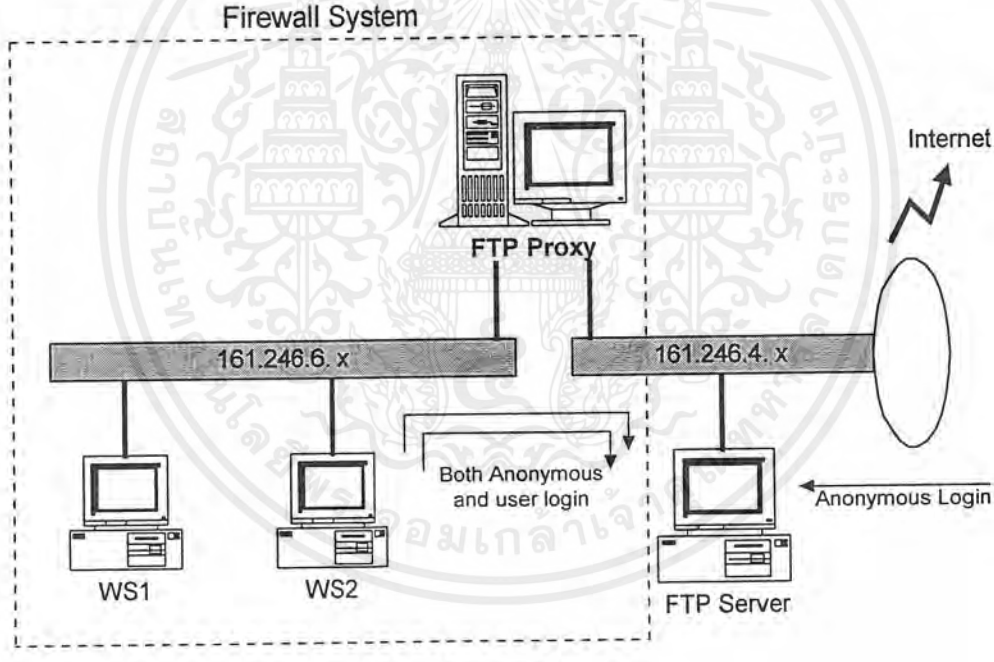
การโอนย้ายในลักษณะแรก เมื่อผู้โอนย้ายข้อมูลสามารถล็อกอินเข้าสู่ระบบได้ ผู้ใช้ดังกล่าวจะมีสิทธิในการอ่านเขียนแฟ้มข้อมูลตามที่ตนเป็นเจ้าของอยู่ และตามที่การยินยอม (permission) ของไฟล์นั้นกำหนดให้ ลักษณะการทำงานโดยทั่วไป ผู้ใช้งานจะมีสิทธิในการเปลี่ยนไดเร็กทอรีที่เป็นราก (root directory) ไปที่ใดก็ได้ภายในระบบ ตามที่สิทธิของตนเองจะเข้าถึง ซึ่งต่างจากลักษณะที่สอง คือ การโอนย้ายไฟล์โดยผู้ใช้ที่ใช้ชื่อว่า anynomous ที่จะต้องมีการกำหนดไดเร็กทอรีที่เป็นรากที่ผู้ใช้ที่เป็น anynomous จะสามารถเข้าถึงข้อมูลได้ (โดยการทำ chroot ก่อน)

ปัญหาที่ตามมาคือ ในกรณีที่เอฟทีพีเซิร์ฟเวอร์นั้นต้องการให้มีทั้งการล็อกอินทั้งสองแบบ คือ ต้องการให้ผู้ใช้ที่มีชื่อ (account) อยู่ในระบบ กับผู้ใช้ที่ใช้ชื่อว่า anynomous ล็อกอินได้ในเครื่องเดียวกัน โปรแกรม ftpd เดิมจะไม่ยอมให้มีการใช้งานในสองลักษณะดังกล่าวพร้อม ๆ กันในเครื่องเดียว

7.3.2.3.1 โปรแกรม wu.ftpd (wuarchive FTP daemon)

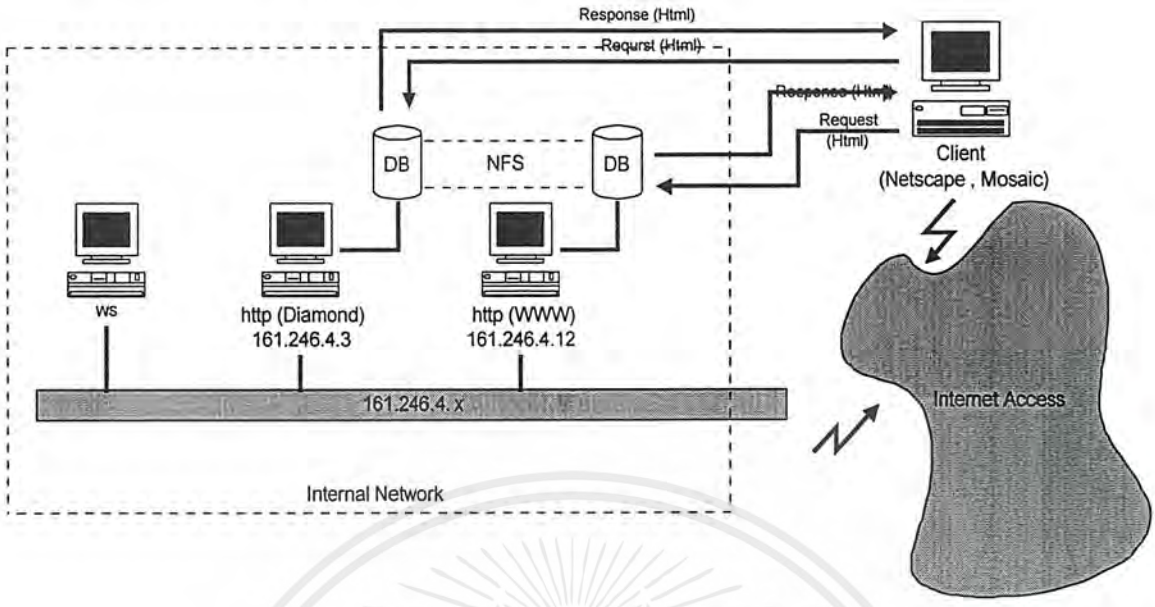
โปรแกรมตัวหนึ่งซึ่งเป็นที่นิยมใช้งานเป็นโปรแกรมเอฟทีพีสำหรับผู้ใช้ทั้งที่ใช้ชื่อว่า anonymous และผู้ใช้ปกติ คือ wu.ftpd (Washington University FTP) เป็นโปรแกรมที่มีคุณสมบัติมากมายที่เหมาะสมจะนำมาใช้เป็นโพรเซสเอฟทีพีภายใต้การใช้งานเป็นโปรแกรมเอฟทีพีสำหรับผู้ใช้ทั้งสองแบบ ซึ่งมีรายละเอียดดังนี้

- สามารถทำล็อกไฟล์ได้ โดยจะล็อกทั้งคำสั่ง (command) ที่ผู้ใช้ใช้ และขนาดของข้อมูลที่ได้ทำการโอนย้าย (ทั้ง upload และ download)
- สามารถใช้เป็นกระดานข่าว (bulletin board) ในขณะที่ผู้ใช้ทำการล็อกอินเข้ามาในระบบ
- สามารถแบ่งกลุ่มผู้ใช้ที่จะเข้ามาใช้งานเอฟทีพีเซิร์ฟเวอร์ได้
- สามารถจำกัดจำนวนการเชื่อมต่อ (connection) ที่ผู้ใช้ทำการล็อกอินเข้ามาพร้อม ๆ กันได้ เช่น จะไม่ยอมให้มีการล็อกอินเข้าระบบเกินกว่า 5 คน เป็นต้น
- สามารถกำหนดระดับของการบีบอัดไฟล์ที่จะใช้ในการส่งได้
- สามารถกำหนดไดเร็กทอรีเริ่มต้นสำหรับผู้ใช้ปกติ และผู้ใช้ที่ใช้ชื่อว่า anonymous



รูป 7.9 การติดต่อระหว่าง Anonymous FTP กับเอฟทีพีไคลเอนต์

7.3.3 บริการเวิลด์ไวด์เว็บ (WWW)



รูป 7.10 การติดต่อระหว่างเว็บเซิร์ฟเวอร์กับโฮสต์ไคลเอนต์

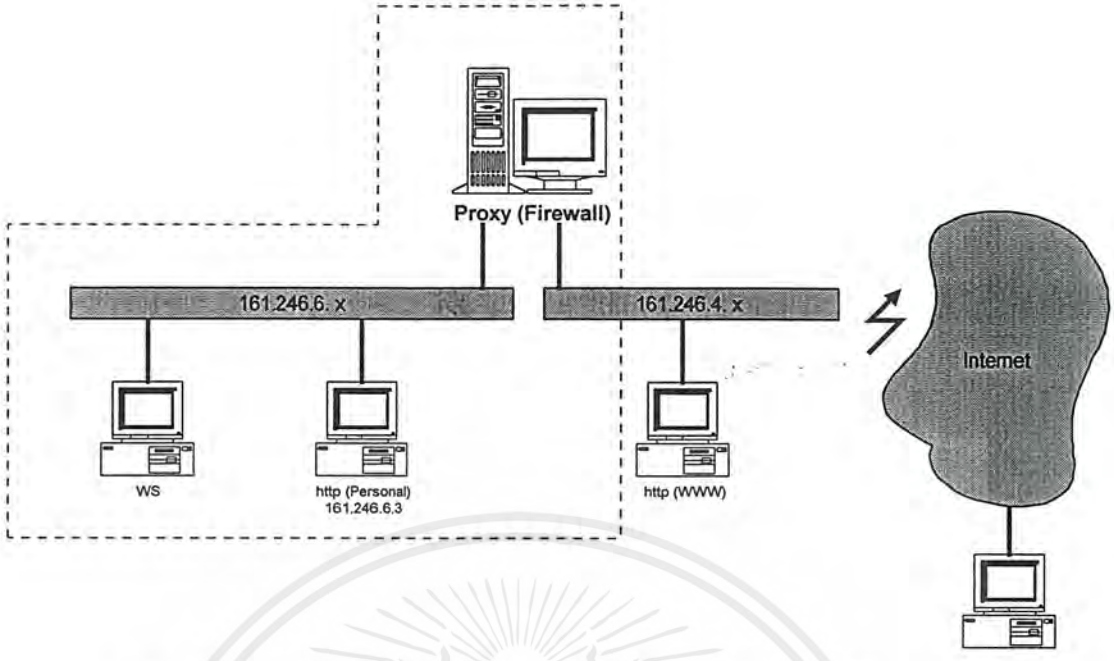
การให้บริการโฮมเพจ (homepage) ของภาควิชาวิศวกรรมคอมพิวเตอร์ แบ่งเป็น 2 ส่วน คือ

- ให้บริการสำหรับผู้ใช้ภายในระบบเครือข่าย เพื่อใช้พื้นที่ในไดเร็กทอรีที่มีอยู่สร้างโฮมเพจของตน
- ให้บริการสำหรับไคลเอนต์ที่อยู่ภายนอกระบบเครือข่าย (Internet) รวมถึงโฮสต์ไคลเอนต์ที่อยู่ภายในระบบเครือข่ายให้สามารถใช้งานข้อมูลผ่านโปรแกรมเบราเซอร์ (browser) ต่าง ๆ ได้ เช่นเน็ตสเคปหรือโมเสค เป็นต้น

การใช้บริการทั้งสองอย่าง จะเกิดปัญหามากที่สุดที่บริการประเภทที่สอง นั่นคือ การยอมให้มีการใช้งานข้อมูลจากภายนอกโดยไม่มีการควบคุมว่าจะให้เครือข่ายวงใดผ่านเข้ามาใช้งานข้อมูล รวมถึงการป้องกันไม่ให้ผู้ใช้ออกไปใช้งานไซต์ (site) ที่เราไม่ต้องการให้ไป (ซึ่งใช้ในกรณีสถานศึกษาหรือองค์กรต่าง ๆ)

7.3.3.1 เอชทีทีพีพร็อกซีเซิร์ฟเวอร์

วิธีการหนึ่งซึ่งเป็นที่นิยมใช้ในการตรวจสอบการใช้งานเอชทีทีพีเซิร์ฟเวอร์ คือ การทำพร็อกซีเซิร์ฟเวอร์ กล่าวคือ การที่โฮสต์ไคลเอนต์จะใช้งานข้อมูลที่เอชทีทีพีเซิร์ฟเวอร์ จะต้องกระทำผ่านพร็อกซีเท่านั้น ถ้าผู้ใช้ไม่กระทำผ่านพร็อกซี จะไม่สามารถใช้งานข้อมูลใด ๆ ของเอชทีทีพีเซิร์ฟเวอร์ปลายทางได้เลย



รูป 7.11 ระบบเครือข่ายที่มีเว็บเซิร์ฟเวอร์ทั้งภายในและภายนอกระบบไฟร์วอลล์

เมื่อทำการติดตั้งระบบไฟร์วอลล์เข้าสู่ระบบเครือข่ายของทางภาควิชา ฯ เพื่อใช้ตรวจสอบการเข้าออกของการใช้บริการเอชทีทีพีจากรูป แสดงให้เห็นบริการเอชทีทีพีหลัก ๆ ซึ่งมีอยู่ 2 ตัวคือ

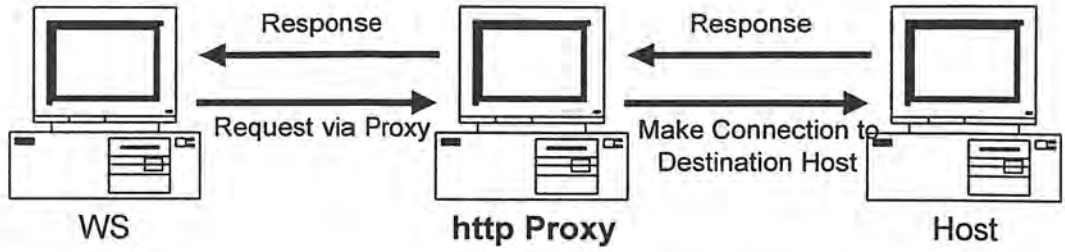
- เวิลด์ไวด์เว็บ (WWW) ของภาควิชา ฯ ที่ใช้แสดงเมนูหลัก
- โฮสต์เพอร์ซันนัล (personal) จะเป็นที่เก็บข้อมูลของนักศึกษาแต่ละคน

ไคลเอนต์ที่ทำการร้องขอบริการจากเอชทีทีพีหลัก (WWW) จะถูกให้บริการทันที เนื่องจากไม่มีการเราท์ (route) แพ็กเก็ตผ่านไฟร์วอลล์แต่อย่างใด พิจารณากรณีต่อไปนี้

- เมื่อโฮสต์ไคลเอนต์ภายในไฟร์วอลล์ต้องการร้องขอบริการจากภายนอกไฟร์วอลล์
- เมื่อโฮสต์ไคลเอนต์จากภายนอกต้องการใช้งานข้อมูล (โฮมเพจ) ภายในโฮสต์ Personal โดยไม่ต้องมีการรับรอง (authentication) จากไฟร์วอลล์

7.3.3.1.1 ในกรณีที่โฮสต์ไคลเอนต์ภายในไฟร์วอลล์ต้องการใช้งานข้อมูลภายนอกไฟร์วอลล์

กรณีนี้เราจะใช้พร็อกซีเป็นตัวผ่านการติดต่อทั้งหมด ระหว่างโฮสต์ไคลเอนต์ที่ร้องขอบริการและโฮสต์ปลายทางที่คอยให้บริการอยู่



รูป 7.12 การติดต่อระหว่างเอชทีทีพีพร็อกซีเซิร์ฟเวอร์กับเอชทีทีพีไคลเอนต์

การติดตั้ง http-gw ให้ใช้ได้กับกรณีแรก มีขั้นตอนดังนี้

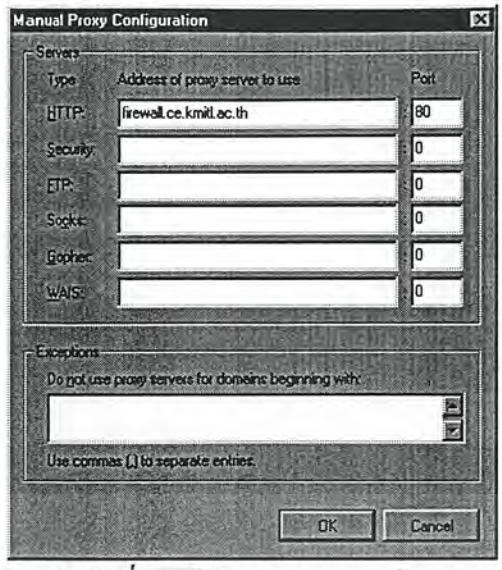
- ที่โฮสต์พร็อกซีเซิร์ฟเวอร์
  - เพิ่มบริการชื่อ http ที่ไฟล์ /etc/services ดังนี้

    http TCP/80 WWW
  - ทำการลบบริการที่ใช้งานพอร์ตหมายเลข 80 (เช่น httpd) ซึ่งส่วนใหญ่บริการดังกล่าวมักจะถูกเรียกให้ทำงานผ่าน /etc/RC.d/RC.*
  - ทำการเพิ่มบรรทัดต่อไปนี้ที่ไฟล์ /etc/inetd.conf

    http stream tcp nowait root /usr/local/etc/http-gw http-gw
  - เพิ่มบรรทัดดังต่อไปนี้ในไฟล์ /usr/local/etc/netperm-table

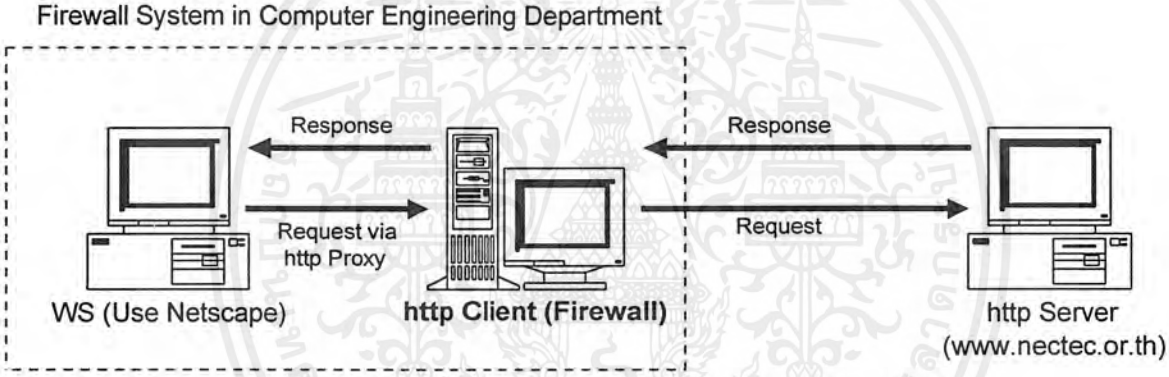
    http-gw: hosts *
  - เมื่อเพิ่มข้อมูลตามที่ได้กำหนดไว้เรียบร้อยแล้ว จะต้องทำการส่งสัญญาณชื่อ SIGHUP ไปให้กับโพรเซส inetd โดยใช้คำสั่งดังนี้

    kill –HUP `ps –ef | grep inetd | cut –f 2`
- ที่โฮสต์ไคลเอนต์
  - กรณีที่ใช้โปรแกรมเบราเซอร์เป็นโปรแกรมไคลเอนต์ จะทำการติดตั้งค่าให้กับเบราเซอร์ เพื่อให้โปรแกรมเปลี่ยนทิศทางการทำงานไปทำงานกับพร็อกซี
  - จากตัวอย่างต่อไปนี้ เป็นการกำหนดให้โปรแกรมเบราเซอร์เน็ตสเคปใช้งานเอชทีทีพีพร็อกซี ที่ถูกติดตั้งที่โฮสต์ชื่อไฟร์วอลล์ (firewall) จะสังเกตเห็นว่า เราจะใช้พอร์ตของเอชทีทีพีพร็อกซีเป็นพอร์ตหมายเลข 80 ซึ่งเป็นพอร์ตเดียวกับพอร์ตเวิลด์ไวด์เว็บ (WWW) แต่ในกรณีเช่นนี้โฮสต์ไฟร์วอลล์จะต้องไม่มีโพรเซสอื่นใดใช้งานพอร์ตหมายเลขดังกล่าวอยู่ นอกจากโปรแกรม http-gw เท่านั้น



รูป 7.13 หน้าต่างเซ็ตอัพที่ทำให้โปรแกรมเบราเซอร์อ่านข้อมูลจากพร็อกซี

7.3.3.1.2 วิธีการทดสอบการทำงานของเอชทีทีพีพร็อกซี



รูป 7.14 เวิร์คสเตชั่นกำลังติดต่อกับเอชทีทีพีเซิร์ฟเวอร์ผ่านพร็อกซี

จากรูป จะกำหนดให้เวิร์คสเตชันใช้โปรแกรมเน็ตสเคป แล้วทำการร้องขอข้อมูลเอกสาร (document) ไปที่เว็บไซต์ (web site) http://www.nectec.or.th ถ้าการร้องขอนั้นได้รับการตอบสนอง (successful) จะต้องมีการรายงานกลับไปที่ล็อกไฟล์ที่ชื่อ /usr/adm/message ดังนี้

Mar 14 11:50:33 firewall http-gw[150]: permit host=ws.ce.kmitl.ac.th/161.246

.4.15 use of gateway (V2.0beta)

Mar 14 11:50:33 firewall http-gw[150]: log host=ws.ce.kmitl.ac.th/161.246.4.

15 protocol=HTTP cmd=get desi=nectec.or.th path=/images/footer600_new.gif

Mar 14 11:50:34 firewall http-gw[148]: exit host=ws.ce.kmitl.ac.th/161.246.4

.15 cmds=1 in=0 out=0 user=unauth duration=2

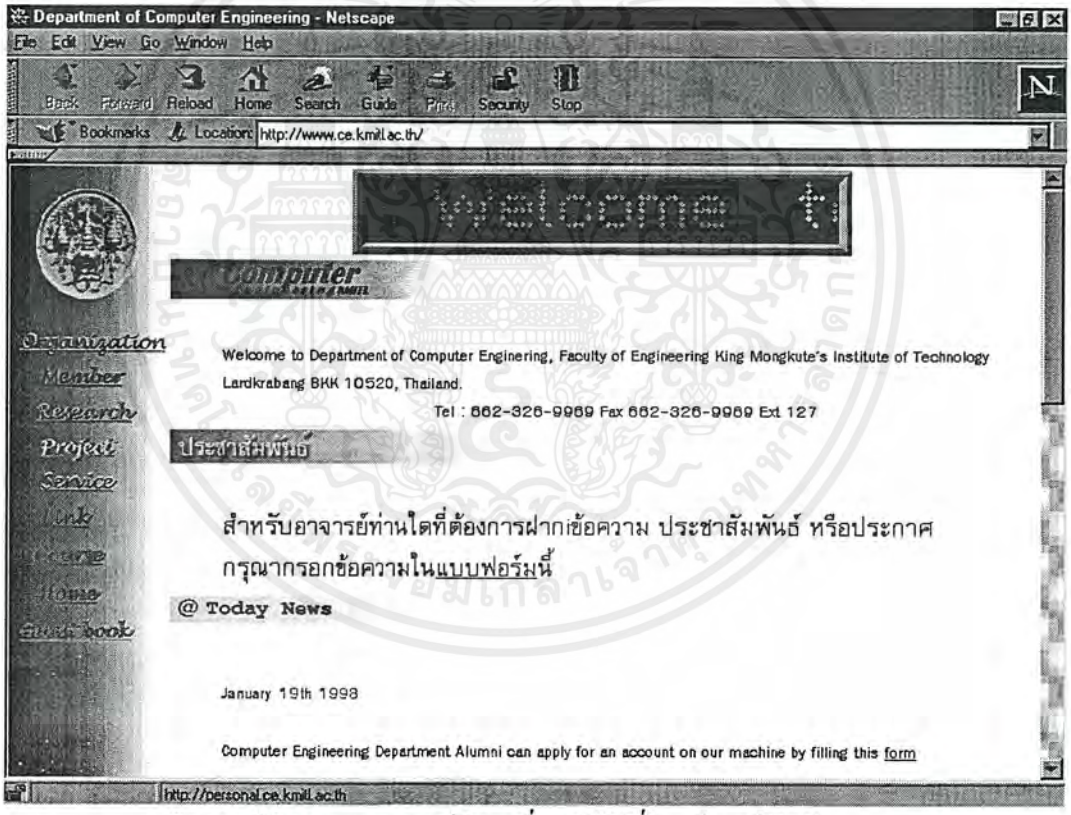
Mar 14 11:50:34 firewall http-gw[150]: exit host=ws.ce.kmitl.ac.th/161.246.4

.15 cmds=1 in=0 out=0 user=unauth duration=1

7.3.3.1.3 กรณีที่โฮสต์ไคลเอนต์ภายนอกไฟร์วอลล์ต้องการใช้งานข้อมูลภายในไฟร์วอลล์

เนื่องจากเมื่อทำการติดตั้งไฟร์วอลล์ให้กับภาควิชา ฯ แล้ว จะมีโฮสต์อยู่ตัวหนึ่งซึ่งมีบัญชีรายชื่อ (account) ของนักศึกษาทุกคนอยู่ นั่นคือโฮสต์เพอร์ซันนัล (personal) แต่โฮสต์ดังกล่าวอยู่ภายในไฟร์วอลล์ ซึ่งบุคคลภายนอกที่จะเข้ามาใช้งานข้อมูลกับโฮสต์ดังกล่าว (กรณีต้องการดูข้อมูลโฮมเพจที่นักศึกษาแต่ละคนได้สร้างไว้) จะต้องกระทำผ่านเอชทีทีพีพร็อกซีเซิร์ฟเวอร์เท่านั้น แต่ในกรณีเช่นนี้ผู้ใช้จากภายนอกจะไม่ทราบว่าเอชทีทีพีพร็อกซีเซิร์ฟเวอร์ของทางภาควิชา ฯ มีชื่อว่าอย่างไร หรือถึงแม้ว่าผู้ใช้จากภายนอกจะรู้จักชื่อของเอชทีทีพีพร็อกซีเซิร์ฟเวอร์ก็ตาม เราก็ไม่ควรให้ผู้ใช้ดังกล่าวใช้งานเอชทีทีพีพร็อกซีเป็นตัวหาเส้นทาง ด้วยสาเหตุที่ว่าผู้ใช้อยู่ภายนอกระบบไฟร์วอลล์ ซึ่งเราจะถือว่าเป็นการใช้งานข้อมูลข้ามเครือข่าย และโดยทั่วไปไม่นิยมที่จะยอมให้มีการใช้งานพร็อกซีเซิร์ฟเวอร์ข้ามเครือข่าย

การแก้ไขจะเป็นการแก้ไขโดยทางอ้อม โดยบุคคลภายนอกจะรู้จักโฮมเพจของทางภาควิชา ฯ ในชื่อ www.ce.kmitl.ac.th ภายในเมนูหลักจะมีการเชื่อมข้อมูลไปที่โฮสต์เพอร์ซันนัล ดังนี้



รูป 7.15 โปรแกรมเบราเซอร์มีการเชื่อมต่อไปที่โฮสต์เพอร์ซันนัล

โปรแกรม http-gw ที่ทำงานที่โฮสต์ไฟร์วอลล์จะมีพารามิเตอร์ (parameter) ตัวหนึ่งชื่อ http-default ใช้กำหนดว่า ถ้ามีการเรียกใช้งานไฟร์วอลล์โดยไม่ใช้วิธีการพร็อกซี (คือไม่กำหนดให้เบราเซอร์ใช้พร็อกซีในการหาเส้นทาง) http-gw จะแสดงโฮมเพจที่ถูกชี้ด้วยค่า http-default เป็นค่าเริ่มต้น ซึ่งใน

การทำโครงงานครั้งนี้ได้กำหนดให้ไฟร์วอลล์มีค่า http-default ชี้ไปยังโฮสต์ไดมอนด์ (diamond) ภายในไฟล์ /usr/local/etc/netperm-table โดยเพิ่มเติมบรรทัดต่อไปนี้

http-gw: http-default diamond.ce.kmitl.ac.th

ด้วยวิธีการนี้เราจึงไม่จำเป็นต้องเปลี่ยนแปลงค่าใด ๆ ภายในโปรแกรม httpd ที่อยู่ในโฮสต์ไดมอนด์ และผู้ใช้ที่มีอยู่ในไดมอนด์ก็ยังคงสามารถสร้างโฮมเพจของตนได้ตามปกติ แต่เราจำเป็นต้องเปลี่ยนเรคคอร์ดของชื่อที่ใช้แทน (alias) ภายในไฟล์ฐานข้อมูลในโดเมนเนมเซิร์ฟเวอร์ ดังนี้

personal IN CNAME firewall

ข้อจำกัดที่เกิดจากทั้งสองกรณีคือ โฮสต์ที่ทำหน้าที่เป็นเอชทีทีพีพร็อกซีเซิร์ฟเวอร์จะต้องรับภาระในการหาเส้นทางแต่เพียงผู้เดียว

7.3.3.1.4 รูปแบบของคำสั่งที่ใช้

http-gw: [*รูปแบบ*]

โดยที่สามารถกำหนด *รูปแบบ* ได้ดังต่อไปนี้

- userid *user*

  ระบุเป็นรหัสผู้ใช้ (user-id) หรือชื่อที่อยู่ในไฟล์รหัสผ่าน (password file) ถ้ามีการใช้งานออพชั่นนี้ โปรแกรม http-gw จะทำการเปลี่ยนรหัสผู้ใช้ไปเป็นรหัสที่กำหนดไว้ ก่อนที่จะเริ่มทำงาน

- directory *pathname*

  เป็นไดเร็กทอรีที่โปรแกรมจะทำการ chroot ไปก่อนจะเริ่มทำงาน

- timeout *secondvalue*

  ถ้าไม่มีการป้อนข้อมูลหลังจากระยะเวลาเท่ากับ *secondvalue* วินาที จะทำการปิดการติดต่อ (disconnect) ค่าปกติ (default) คือ 60 นาที

- default-httpd *server*

  ถ้ามีการระบุการเข้าถึง (access) แบบเอชทีทีพี จะเปลี่ยนการเข้าถึงไปที่เซิร์ฟเวอร์ที่ระบุด้วย *server*

  เป็นการระบุโฮสต์ และการยินยอมให้ใช้งาน (access permission) โดยสามารถใช้รูปแบบของพารามิเตอร์ได้ดังนี้

  ➢ *host-pattern*

  - ระบุเป็นหมายเลขของเครือข่ายได้ เช่น 161.246.4.*
  - ระบุเป็นไอพีแอดเดรสของโฮสต์ได้ เช่น 161.246.4.3
  - ระบุเป็นชื่อโฮสต์ เช่น diamond.ce.kmitl.ac.th

  ตามรูปแบบของคำสั่ง จะเห็นได้ว่าสามารถระบุโฮสต์ได้มากกว่าหนึ่ง

➢ *options*

- -log *function*

  -log { *function* [*function*...] }

  ระบุว่าให้ระบบทำการเก็บข้อมูลลงล็อกไฟล์ เมื่อมีการทำ *function* ผ่านพร็อกซี

- -authall

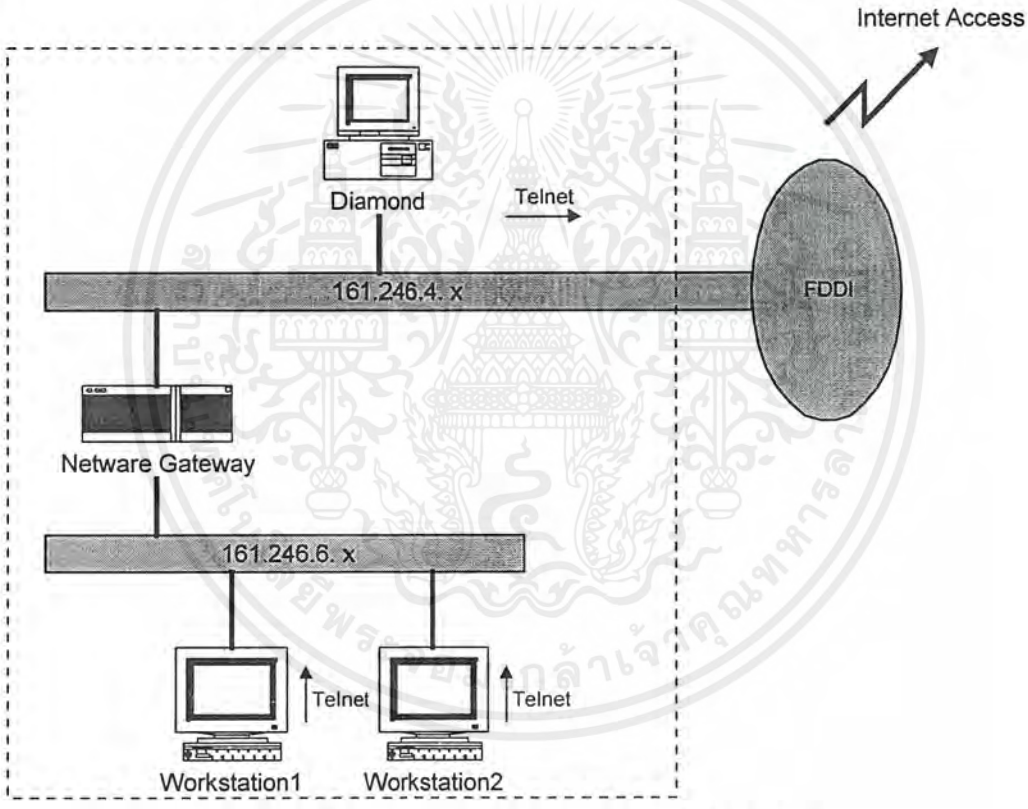
  ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งานฟังก์ชั่นใด ๆ ได้

- -auth *function*

  -auth { *function* [*function*] }

  ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งาน *function* ที่กำหนดไว้ได้

7.3.4 บริการเทลเน็ต (TELNET Service)



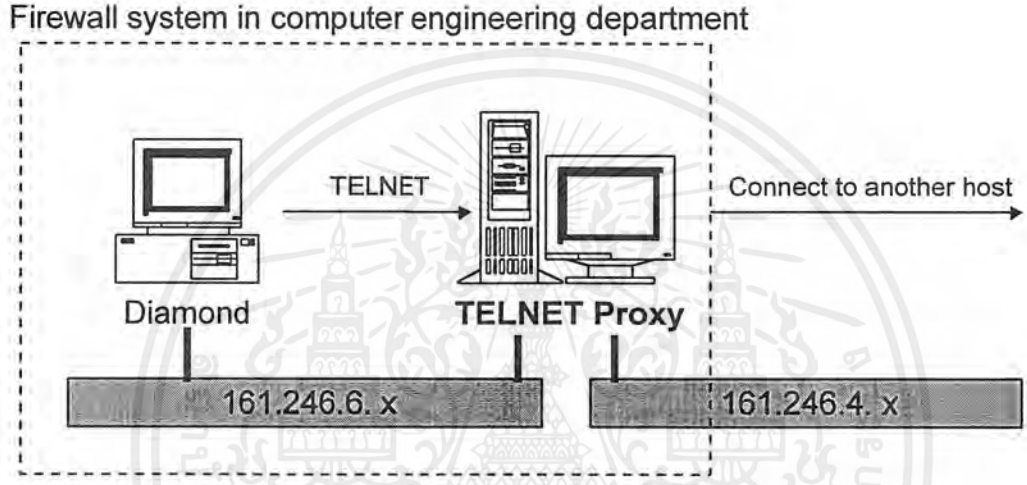
รูป 7.16 โฮสต์ทุกตัวสามารถใช้โปรแกรมเทลเน็ตได้อย่างเป็นอิสระ

เครื่องเทอร์มินัล (terminal) หรือเวิร์คสเตชั่น (workstation) ที่มีโปรแกรมประเภทเทอร์มินัลอิมูเลเตอร์ (terminal emulator) ทุกเครื่องจะสามารถทำการรีโมตล็อกอิน (remote login) ไปยังโฮสต์ใด ๆ ได้โดยไม่จำกัดว่าจะอยู่ที่เครือข่ายวงใด รวมถึงบุคคลภายนอกที่จะทำการรีโมตล็อกอินเข้ามาใช้โฮสต์ภายในก็สามารถทำได้เช่นเดียวกัน

7.3.4.1 ปัญหาที่เกิดขึ้นกับระบบเดิม

ระบบเดิมที่ยอมให้มีการทำรีโมตล็อกอินได้จากทุก ๆ จุด จะมีข้อเสียอยู่ว่า ถ้ามีกรณีที่ผู้ใช้ทำการล็อกอินจากระยะไกล เช่น ซูเปอร์ยูสเซอร์ (superuser) ทำการล็อกอินจากโฮสต์เจ้าคุณ (chaokhun) ไปยังโฮสต์ไดมอนด์ (diamond) จะต้องมีการส่งแพ็กเก็ตข้อมูลไปตามสาย ซึ่งข้อมูลดังกล่าวอาจถูกดักจับได้ระหว่างทางที่แพ็กเก็ตข้อมูลผ่านไป โดยการใช้โปรแกรมประเภทสนิฟเฟอร์ (sniffer) หรือในกรณีที่มีการทำรีโมตล็อกอินจากโฮสต์ที่เราไม่ไว้ใจ เราสามารถแก้ปัญหาดังกล่าวได้ดังนี้

- กรณีใช้โปรแกรมเทลเน็ตจากภายในไฟร์วอลล์
- กรณีที่บุคคลภายนอกทำการรีโมตล็อกอินเข้ามาใช้โฮสต์ภายในภาควิชา ฯ

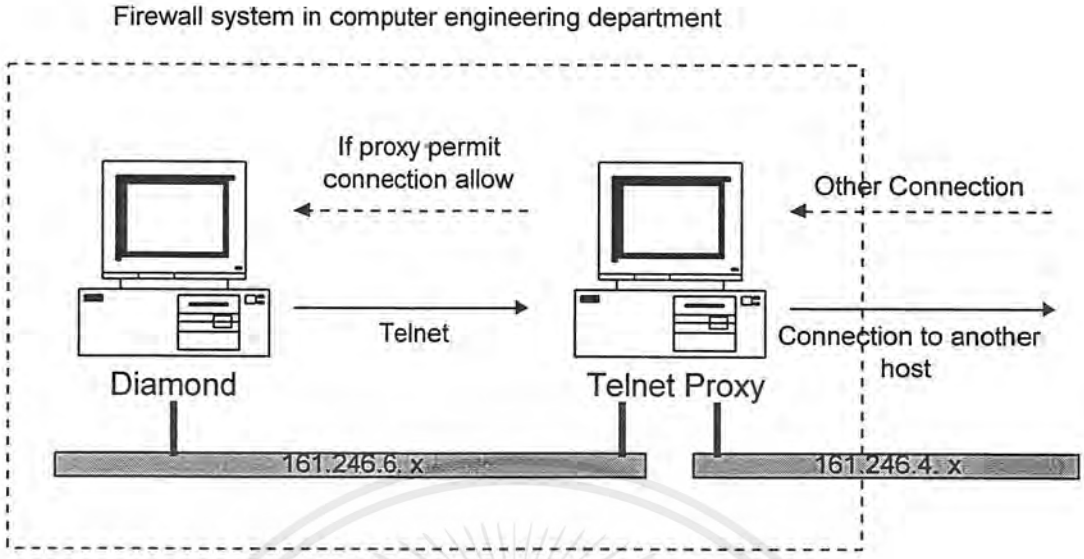


รูป 7.17 การใช้โปรแกรมเทลเน็ตผ่านพร็อกซี

การใช้โปรแกรมเทลเน็ตจากภายนอกเข้ามาสู่ภายในภาควิชา ฯ จะมีกรณีที่ผู้บุกรุกทราบชื่อของผู้ใช้ภายในโฮสต์ (user name) และทราบรหัสผ่าน (password) ของผู้ใช้นั้น แล้วทำการเทลเน็ตจากระยะไกลเพื่อก่อการร้ายกับโฮสต์ที่ทำการรีโมตล็อกอินมา โดยกรณีดังกล่าวจะเกิดจากผู้ใช้ภายในระบบเองที่ไม่ใช้แอคเคาท์ (account) นั้นมาเป็นเวลานานจนผู้บุกรุกสามารถเดารหัสผ่านและใช้รหัสผ่านนั้นในการก่อการร้าย ซึ่งส่วนใหญ่แล้วผู้บุกรุกมักจะมาจากเครือข่ายอื่น ๆ ที่มีผู้ใช้อยู่มาก จนไม่สามารถจับกุมได้

วิธีการป้องกันการบุกรุกจากเครือข่ายทุกวงที่เราไม่ไว้ใจ (untrusted network) เราจะใช้วิธีการที่เรียกว่าเทลเน็ตพร็อกซีเซิร์ฟเวอร์ (TELNET proxy server) คือ ทำการตรวจสอบแพ็กเก็ตข้อมูลก่อนว่าแพ็กเก็ตข้อมูลดังกล่าวเป็นแพ็กเก็ตที่มาจากเครือข่ายที่เราไว้ใจหรือไม่เท่านั้น ถ้าแพ็กเก็ตนั้นมาจากเครือข่ายที่เราไว้ใจ (trusted network) ก็จะทำการส่งแพ็กเก็ตนั้นเข้าสู่เครือข่ายของเรา (forwarding packet) ถ้าเป็นแพ็กเก็ตที่มาจากเครือข่ายที่เราไม่ไว้ใจ ก็จะส่งข้อความปฏิเสธกลับไปยังโฮสต์ต้นทาง โปรแกรมที่จะนำมาประยุกต์ใช้กับทางภาควิชา ฯ คือ tn-gw

7.3.4.2 วิธีการติดตั้ง tn-gw



รูป 7.18 การเทลเน็ตจากภายนอกซึ่งต้องได้รับอนุญาตจากพร็อกซี

- ที่พร็อกซีเซิร์ฟเวอร์
  - ทำการแทนที่โปรแกรม telnetd ด้วยโปรแกรม tn-gw ที่ไฟล์ /etc/inetd.conf ดังนี้

    telnet stream tcp nowait root /usr/local/etc/tn-gw tn-gw
  - เพิ่มบรรทัดต่อไปนี้ที่ไฟล์ /usr/local/etc/netperm-table เป็นการปรับตั้งค่าให้ระบบทราบว่ามีการเริ่มต้นใช้งานเทลเน็ตเกตเวย์

    tn-gw: permit-hosts *

    ในกรณีที่ยอมให้โฮสต์ที่มาจากทุก ๆ เครือข่ายสามารถใช้งานพร็อกซีได้ (พารามิเตอร์ที่ใช้กับ tn-gw ได้อธิบายเพิ่มเติมในตอนท้าย)
  - ทำการส่งสัญญาณที่ชื่อ SIGHUP ไปให้กับโพรเซส inetd โดยใช้คำสั่งดังนี้

    kill –HUP `ps –ef | grep inetd | cut –f 2`
- ที่โฮสต์ไคลเอนต์

  เราสามารถใช้โปรแกรมเทลเน็ตที่มีอยู่เดิมกับเทลเน็ตพร็อกซีเซิร์ฟเวอร์ได้ โดยไม่ต้องมีการแก้ไขใด ๆ ทั้งสิ้น

7.3.5 รูปแบบของคำสั่งที่ใช้

tn-gw: [*รูปแบบ*]

โดยที่สามารถกำหนด *รูปแบบ* ได้ดังต่อไปนี้

- userid *user*

  ระบุเป็นรหัสผู้ใช้ (user-id) หรือชื่อที่อยู่ในไฟล์รหัสผ่าน (password file) ถ้ามีการใช้งานออพชั่นนี้ โปรแกรม tn-gw จะทำการเปลี่ยนรหัสผู้ใช้ไปเป็นรหัสที่กำหนดไว้ ก่อนที่จะเริ่มทำงาน

- directory *pathname*

  เป็นไดเร็กทอรีที่โปรแกรมจะทำการ chroot ไปก่อนจะเริ่มทำงาน

- prompt *string*

  กำหนดเครื่องหมายพร้อมรับคำสั่ง (prompt) ที่โปรแกรม tn-gw จะใช้แสดงบนหน้าจอขณะทำงาน

- timeout *secondvalue*

  ถ้าไม่มีการป้อนข้อมูลหลังจากระยะเวลาเท่ากับ *secondvalue* วินาที จะทำการปิดการติดต่อ (disconnect) ค่าปกติ (default) คือ 60 นาที

- denial-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดงให้ผู้ใช้ทราบ ถ้าผู้ใช้คนนั้นไม่มีสิทธิในการใช้งานพร็อกซี ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- welcome-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดงให้ผู้ใช้เมื่อการติดต่อเป็นผลสำเร็จ (successful) ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- help-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดง เมื่อผู้ใช้ใช้คำสั่ง help ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- authserver *hostname*

  ระบุชื่อโฮสต์ที่ tn-gw จะทำการรับรองผู้ใช้ (authentication) ด้วย (รายละเอียดของการรับรองผู้ใช้จะอธิบายภายหลัง)

- permit-hosts *host-pattern* [*host-pattern...*] [*options*]

  deny-hosts *host-pattern* [*host-pattern...*] [*options*]

  เป็นการระบุโฮสต์ และการยินยอมให้ใช้งาน (access permission) โดยสามารถใช้รูปแบบของพารามิเตอร์ได้ดังนี้

  - *host-pattern*
    - ระบุเป็นหมายเลขของเครือข่ายได้ เช่น 161.246.4.*
    - ระบุเป็นไอพีแอดเดรสของโฮสต์ได้ เช่น 161.246.4.3

- ระบุเป็นชื่อโฮสต์ เช่น diamond.ce.kmitl.ac.th

ตามรูปแบบของคำสั่ง จะเห็นได้ว่าสามารถระบุโฮสต์ได้มากกว่าหนึ่ง

➢ *options*

- -dest *hostpattern*

  ระบุโฮสต์ปลายทางที่สามารถติดต่อด้วยได้ ถ้าไม่มีการระบุคำสั่งนี้ไว้จะสามารถติดต่อกับโฮสต์ได้ทุกเครื่อง สามารถใช้นิเสธได้ด้วยเครื่องหมาย !

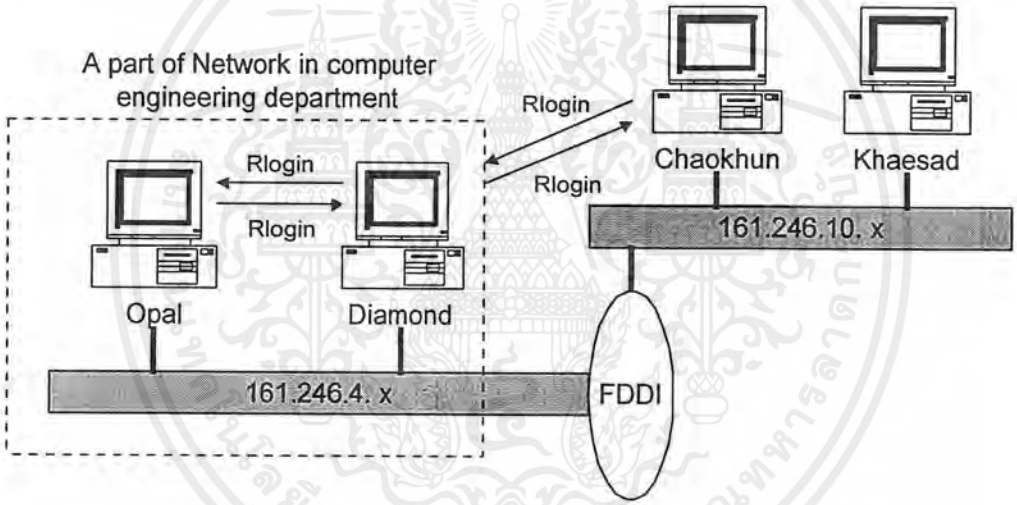
- -auth

  ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งานพร็อกซีได้

- -passok

  ยอมให้มีการเปลี่ยนรหัสผ่านของผู้ใช้ที่ออเธนทิเคชั่นเซิร์ฟเวอร์ได้

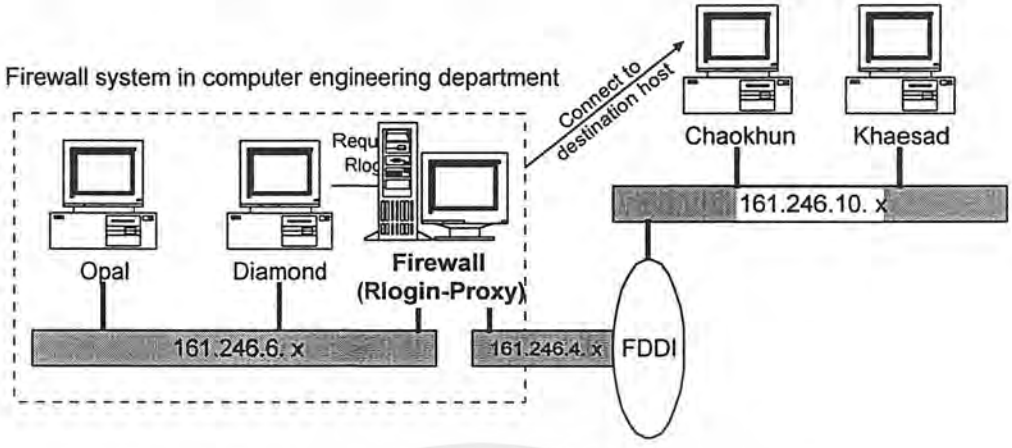
7.3.6 บริการอาร์ล็อกอิน (rlogin Service)



รูป 7.19 อาร์ล็อกอินที่ไม่มีระบบไฟร์วอลล์

การใช้งานโฮสต์ที่ต่อบนระบบเครือข่ายท้องถิ่น (LAN) ส่วนใหญ่นิยมติดต่อเพื่อทำรีโมทล็อกอินแบบอาร์ล็อกอิน (rlogin) แทนการใช้งานแบบเทลเน็ต สาเหตุที่ผู้ใช้เลือกใช้การทำรีโมทล็อกอินแบบอาร์ล็อกอิน เนื่องจากผลของการทำโปรแกรมบางโปรแกรมส่งผลถึงเครื่องเทอร์มินัลที่เรียกเข้าหาระบบ ตัวอย่างเช่น โฮสต์ที่ชื่อแคแสด (khaesad) ไม่มีโปรแกรม SLIP (Serial Line Internet Protocol) แต่โปรแกรมดังกล่าวมีอยู่ที่โฮสต์เจ้าคุณ (chaokhun) เราสามารถทำอาร์ล็อกอินจากโฮสต์แคแสดไปที่โฮสต์เจ้าคุณ แล้วเรียกโปรแกรม SLIP ที่โฮสต์เจ้าคุณ แต่ผลที่เกิดขึ้นทั้งหมดจะมีการผ่านค่าไปที่โฮสต์แคแสดแทน แต่ถ้าเราทำการเทลเน็ตจากโฮสต์แคแสดไปที่โฮสต์เจ้าคุณ แล้วเรียกโปรแกรม SLIP ที่

โฮสต์เจ้าคุณ ผลที่เกิดขึ้นจะให้ผลกับหมายเลขเครื่องเทอร์มินัลที่เรียกเข้าหาโฮสต์เจ้าคุณเท่านั้น ไม่ส่งผลกลับไปยังโฮสต์แคแสด



รูป 7.20 การทำอาร์ล็อกอินผ่านพร็อกซีหลังติดตั้งระบบไฟร์วอลล์

จากรูป แสดงการเชื่อมต่อโดยคำสั่งอาร์ล็อกอิน จะเห็นได้ว่าเราไม่สามารถใช้คำสั่งอาร์ล็อกอินโดยตรงไปยังโฮสต์ปลายทางได้ เนื่องจากต้องผ่านไฟร์วอลล์ และไฟร์วอลล์จะต้องตรวจสอบว่าโฮสต์ต้นทางได้รับความยินยอมที่จะติดต่อแบบอาร์ล็อกอินภายนอกไฟร์วอลล์หรือไม่ แต่จะมีกรณีที่โฮสต์ภายในระบบใช้คำสั่งอาร์ล็อกอินติดต่อกันเอง กรณีนี้จะไม่มีการผ่านไฟร์วอลล์ ดังนั้น คำสั่งอาร์ล็อกอินที่เกิดขึ้นภายในเครือข่ายเดียวกันจะได้รับความยินยอมเสมอ ไม่เกี่ยวข้องกับไฟร์วอลล์แต่อย่างใด

7.3.6.1 อาร์ล็อกอินพร็อกซีเซิร์ฟเวอร์

วิธีการหนึ่งที่ใช้กำหนดการใช้คำสั่งอาร์ล็อกอินข้ามเครือข่าย ซึ่งเราสามารถควบคุมได้ว่าเราจะยินยอมให้มีการใช้อาร์ล็อกอินข้ามเครือข่ายก็ต่อเมื่อ โฮสต์ที่เรียกเข้าหาอาร์ล็อกอินพร็อกซีเซิร์ฟเวอร์อยู่ในเครือข่ายที่เรากำหนดให้ยินยอมให้ใช้คำสั่งอาร์ล็อกอิน โปรแกรมที่เราใช้คือ rlogin-gw ที่มากับไฟร์วอลล์ทูลคิท (FWTK)

7.3.6.1.1 การติดตั้ง rlogin-gw

- ที่พร็อกซีเซิร์ฟเวอร์
  - ➢ แทนที่โปรแกรม rlogind ด้วยโปรแกรม rlogin-gw ที่ไฟล์ /etc/inetd.conf ดังนี้

    ```
    rlogin stream tcp nowait root /usr/local/rlogin-gw rlogin-gw
    ```
  - ➢ เพิ่มบรรทัดต่อไปนี้ในไฟล์ /usr/local/etc/netperm-table

    ```
    rlogin-gw: hosts *
    ```
  - ➢ ทำการส่งสัญญาณที่ชื่อ SIGHUP ไปให้กับโพรเซส inetd โดยใช้คำสั่งดังนี้

    ```
    kill –HUP `ps –ef | grep inetd | cut –f 2`
    ```

- ที่โฮสต์ไคลเอนต์
  - โปรแกรมอาร์ล็อกอินที่เคยใช้งานอยู่ จะยังคงใช้งานได้ตามปกติ แต่จะไม่สามารถทำการอาร์ล็อกอินไปที่โฮสต์ปลายทางได้โดยตรง แต่จะต้องกระทำผ่านอาร์ล็อกอินพร็อกซีเซิร์ฟเวอร์เท่านั้น ตัวอย่างการใช้อาร์ล็อกอินจากโฮสต์ไคมอนด์ไปที่โฮสต์เจ้าคุณ จะแสดงผลดังนี้

```
# rlogin firewall
Connected to firewall.
**********************************************************
                Welcome to Firewall System
**********************************************************
rlogin-gw> c diamond
login: s7014010
passwd:
....
.......
$
```

7.3.6.1.2 รูปแบบของคำสั่งที่ใช้

rlogin-gw: [*รูปแบบ*]

โดยที่สามารถกำหนด *รูปแบบ* ได้ดังต่อไปนี้

- directory *pathname*

  เป็นไดเร็กทอรีที่โปรแกรมจะทำการ chroot ไปก่อนจะเริ่มทำงาน

- prompt *string*

  กำหนดเครื่องหมายพร้อมรับคำสั่ง (prompt) ที่โปรแกรม tn-gw จะใช้แสดงบนหน้าจอขณะทำงาน

- timeout *secondvalue*

  ถ้าไม่มีการป้อนข้อมูลหลังจากระยะเวลาเท่ากับ *secondvalue* วินาที จะทำการปิดการติดต่อ (disconnect) ค่าปกติ (default) คือไม่มี

- denial-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดงให้ผู้ใช้ทราบ ถ้าผู้ใช้คนนั้นไม่มีสิทธิในการใช้งานพร็อกซี ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- help-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดง เมื่อผู้ใช้ใช้คำสั่ง help ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- denydest-msg *filename*

  ระบุไฟล์ที่ต้องการให้แสดง เมื่อผู้ใช้ทำการติดต่อไปยังโฮสต์ที่ระบบไม่อนุญาต ถ้าไม่มีการระบุค่าไว้ ระบบจะแสดงไฟล์ที่เป็นค่าปกติ

- authserver *hostname*

  ระบุชื่อโฮสต์ที่ rlogin-gw จะทำการรับรองผู้ใช้ (authentication) ด้วย (รายละเอียดของการรับรองผู้ใช้จะอธิบายภายหลัง)

- hosts *host-pattern* [*host-pattern...*] [*options*]

  deny-hosts *host-pattern* [*host-pattern...*] [*options*]

  เป็นการระบุโฮสต์ และการยินยอมให้ใช้งาน (access permission) โดยสามารถใช้รูปแบบของพารามิเตอร์ได้ดังนี้

  - *host-pattern*
    - ระบุเป็นหมายเลขของเครือข่ายได้ เช่น 161.246.4.*
    - ระบุเป็นไอพีแอดเดรสของโฮสต์ได้ เช่น 161.246.4.3
    - ระบุเป็นชื่อโฮสต์ เช่น diamond.ce.kmitl.ac.th

    ตามรูปแบบของคำสั่ง จะเห็นได้ว่าสามารถระบุโฮสต์ได้มากกว่าหนึ่ง
  - *options*
    - -dest *hostpattern*

      ระบุโฮสต์ปลายทางที่สามารถติดต่อด้วยได้ ถ้าไม่มีการระบุคำสั่งนี้ไว้จะสามารถติดต่อกับโฮสต์ได้ทุกเครื่อง สามารถใช้นิเสธได้ด้วยเครื่องหมาย !
    - -auth

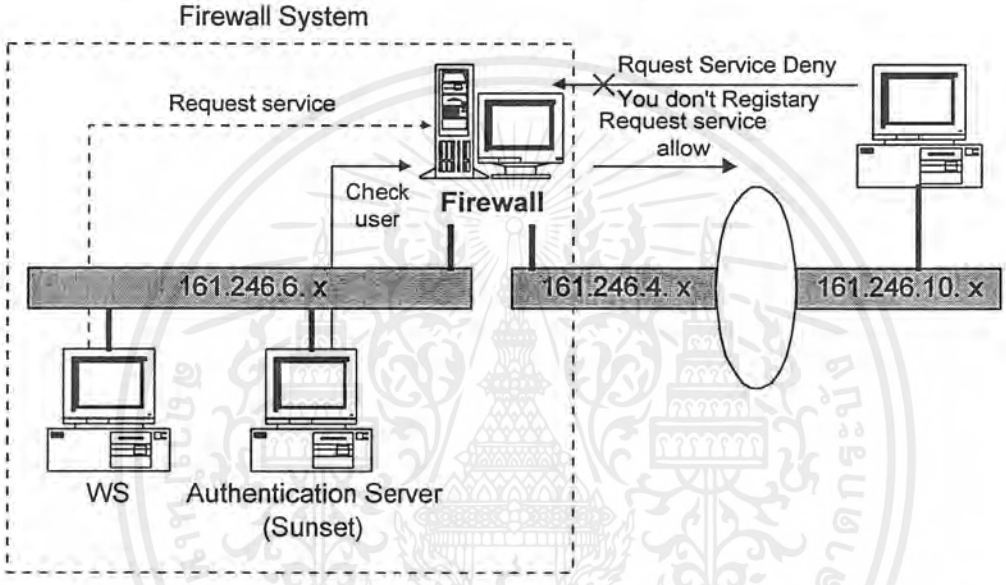
      ผู้ใช้ต้องผ่านการรับรองก่อนที่จะสามารถใช้งานพร็อกซีได้
    - -passok

      ยอมให้มีการเปลี่ยนรหัสผ่านของผู้ใช้ที่ออเธนทิเคชั่นเซิร์ฟเวอร์ได้

7.3.7 การรับรองผู้ใช้ก่อนการใช้งาน (Authentication Server)

เนื่องจากในบางกรณีนั้น บริการที่ถูกเรียกใช้ภายในภาควิชา ฯ ซึ่งได้แก่การเทลเน็ต , เอฟทีพี , อาร์ล็อกอิน รวมไปถึงเอชทีทีพี เป็นการเรียกใช้ทั้งจากผู้ใช้ภายในระบบเอง และผู้ใช้ที่อยู่ภายนอกระบบที่มีความจำเป็นจะต้องผ่านไฟร์วอลล์ เพื่อเข้ามาใช้บริการต่าง ๆ จากการติดตั้งโปรแกรมที่ผ่านมา เป็นการติดตั้งโปรแกรมในลักษณะที่มีการควบคุมการผ่านเข้าออกของโฮสต์ โดยดูจากแพ็กเก็ตที่ระบุต้นทางของข้อมูล ถ้าต้นทางได้รับอนุญาตให้ใช้บริการได้ จะมีการส่งต่อแพ็กเก็ตดังกล่าวไปยังโฮสต์ปลายทางต่อไป จะเห็นได้ว่าการควบคุมดังกล่าวให้ความปลอดภัยในระดับหนึ่งเท่านั้น ในกรณีที่แพ็กเก็ตที่ผ่านมายังไฟร์วอลล์มีการใส่ไอพีแอดเดรส (IP address) ของต้นทางเป็นค่าไอพีแอดเดรสที่ได้รับอนุญาตให้สามารถผ่านไฟร์วอลล์ได้ แต่ในความเป็นจริงแพ็กเก็ตดังกล่าวอาจไม่ได้มาจากโฮสต์ที่มีไอพีแอดเดรสนั้น ๆ จริง ๆ นั่นคือมีการปลอมไอพีแอดเดรส (IP spoofing) ซึ่งเราจะไม่สามารถตรวจสอบได้เลยว่า

แพ็กเก็ตนั้นมาจากโฮสต์ที่มีไอพีแอดเดรสนั้น ๆ จริงหรือไม่ แต่เราสามารถทำการแก้ไขได้โดยการตรวจสอบรายชื่อของผู้ใช้บริการอีกทีหนึ่ง นั่นคือ ถ้าเป็นบุคคลที่ได้รับอนุญาตให้สามารถใช้บริการใดบริการหนึ่งได้แล้ว ผู้ดูแลระบบ (administrator) จะต้องทำการสร้างแอคเคาท์ (account) ของผู้ใช้นั้น ๆ อีกแอคเคาท์หนึ่งที่ไฟร์วอลล์ ถ้าผู้ใช้อยู่ภายในระบบและต้องการใช้บริการใด ๆ ก็จะสามารถใช้ได้ตามปกติ แต่ถ้าผู้ใช้ต้องการใช้งานไฟร์วอลล์พร็อกซี ไฟร์วอลล์จะถามชื่อผู้ใช้ (user name) ของผู้ใช้บริการนั้นก่อน ซึ่งชื่อผู้ใช้ที่ไฟร์วอลล์กับชื่อผู้ใช้สำหรับใช้บริการในเครือข่ายอาจเหมือนหรือต่างกันก็ได้ (ขึ้นอยู่กับผู้ดูแลระบบ) ด้วยวิธีการนี้ ทำให้เราสามารถตรวจสอบได้ว่า ผู้ที่จะผ่านไฟร์วอลล์พร็อกซีเป็นผู้ใช้ที่อยู่ภายในระบบจริง (เพราะได้ผ่านการตรวจสอบผู้ใช้แล้วที่ไฟร์วอลล์ นอกเหนือไปจากการล็อกอินและรายชื่อของผู้ใช้ปรากฏอยู่ทั้งสองที่) ขั้นตอนนี้คือ การรับรองผู้ใช้ (authentication)



รูป 7.21 การใช้งานออเธนทิเคชั่นเซิร์ฟเวอร์

7.3.7.1 วิธีการติดตั้งออเธนทิเคชั่นเซิร์ฟเวอร์เพื่อให้บริการในการรับรองผู้ใช้

โปรแกรมที่มาจากไฟร์วอลล์ทูลคิททุกตัวที่เป็นลักษณะของพร็อกซี จะมีออพชั่นที่บอกว่าให้ทำการรับรองผู้ใช้ก่อนใช้งาน โดยทั่วไปแบ่งการติดตั้งออกเป็นสามส่วนดังนี้

- ที่ออเธนทิเคชั่นเซิร์ฟเวอร์

  เราจะติดตั้งออเธนทิเคชั่นเซิร์ฟเวอร์ให้มีความปลอดภัยสูงสุด จะต้องกำหนดให้ออเธนทิเคชั่นเซิร์ฟเวอร์ทำหน้าที่ในการรับรองผู้ใช้เพียงอย่างเดียวเท่านั้น ไม่มีบริการอื่น ๆ อีก เช่น เทลเน็ต หรือเอฟทีพี เป็นต้น ดังนั้นจะมีขั้นตอนการติดตั้งดังนี้

  - ทำการยกเลิกบริการที่ได้ทำการติดตั้งมากับระบบปฏิบัติการ (Operationg System) แต่ละตัวให้หมด (ในที่นี้เราใช้โซลาริสเวอร์ชั่น 5 (Solaris version 5) เป็นระบบปฏิบัติการของออเธนทิเคชั่นเซิร์ฟเวอร์ และใช้ไลนุกซ์ (Linux) เป็นระบบปฏิบัติการของเครื่องพร็อกซีเซิร์ฟเวอร์) ดังนี้

```
#echo    stream  tcp   nowait  root   internal
#echo    dgram   udp   wait    root   internal
#discard stream  tcp   nowait  root   internal
#discard dgram   udp   wait    root   internal
#daytime stream  tcp   nowait  root   internal
#chargen stream  tcp   nowait  root   internal
...
...
```

➢ ทำการกำหนดหมายเลขบริการของ auth ที่ไฟล์ /etc/services ดังนี้

```
auth    113/tcp    ident
```

➢ ทำการติดตั้งโปรแกรมของออเธนทิเคชั่นเซิร์ฟเวอร์ ชื่อ authsrv ที่ไฟล์ /etc/inetd.conf ดังนี้

```
auth    stream  tcp  wait   root  /usr/sbin/tcpd  in.identd -w -t120
authsrv stream  tcp  nowait root  /usr/local/etc/authsrv  authsrv
```

➢ ทำการส่งสัญญาณ SIGHUP ไปที่โพรเซส inetd เพื่อให้ inetd ทำการอ่านไฟล์ /etc/inetd.conf ใหม่

ข้อสังเกต : จะเห็นว่าบริการ auth จะถูกตรวจสอบโดยโปรแกรม tcp wrapper อีกทีหนึ่ง (tcpd) เพื่อบันทึกความเปลี่ยนแปลงของบริการ auth

■ การรับรองผู้ใช้ที่บริการแต่ละตัว

➢ ในบริการทุกตัวของไฟร์วอลล์ทูลคิทที่สามารถทำหน้าที่เป็นพร็อกซีได้นั้น เราสามารถเพิ่มออพชั่น -auth เพื่อเป็นการบอกว่าบริการดังกล่าวจะต้องทำการรับรองผู้ใช้ที่ไฟร์วอลล์ก่อนทำการให้บริการ โดยกระทำการเพิ่มบรรทัดต่อไปนี้ที่ไฟล์ /usr/local/etc/netperm-table หลังบริการแต่ละตัว ดังนี้

```
http-gw:      host  *   -auth
ftp-gw:       host  *   -auth
tn-gw:        host  *   -auth
rlogin-gw:    host  *   -auth
```

แต่ที่ไฟล์ /etc/inetd.conf จะต้องมีบรรทัดต่อไปนี้ด้วย คือ

```
auth    stream  tcp  wait   root  /usr/sbin/tcpd  in.identd -w -t120
```

➢ ทำการส่งสัญญาณ SIGHUP ไปยังโพรเซส inetd

- ที่โฮสต์ไคลเอนต์
  - จะไม่มีการเปลี่ยนแปลงใด ๆ ทั้งสิ้น เนื่องจากโปรแกรมเดิมที่ทำหน้าที่ให้บริการยังคงสามารถใช้งานพร็อกซีได้ (ซึ่งจะขึ้นกับพร็อกซีอีกทีหนึ่งว่า จะให้ทำการรับรองผู้ใช้ด้วยหรือไม่)

7.3.7.2 การเพิ่มเติมชื่อผู้ใช้ภายในออเธนทิเคชั่นเซิร์ฟเวอร์

เราจะทำการเพิ่มชื่อผู้ใช้เข้าไปในออเธนทิเคชั่นเซิร์ฟเวอร์ได้ 2 ทาง คือ กระทำที่โฮสต์ที่ทำหน้าที่เป็นออเธนทิเคชั่นเซิร์ฟเวอร์โดยตรง หรือกระทำผ่านเครือข่าย ทั้งนี้ต้องได้รับความยินยอมจากไฟร์วอลล์ก่อน ดังนี้

- เพิ่มเติมชื่อผู้ใช้ที่ออเธนทิเคชั่นเซิร์ฟเวอร์โดยตรง

  โปรแกรมที่ใช้ทำการเพิ่มเติมรายชื่อผู้ใช้คือ /usr/local/etc/authsrv ซึ่งก่อนใช้งาน ผู้ที่จะทำการเพิ่มเติมรายชื่อจะต้องล็อกอิน หรือเปลี่ยนพรีวิเลจ (previlege) ของตนเองเป็นซูเปอร์ยูสเซอร์ (superuser) ก่อนเสมอ

  เมื่อเรียกคำสั่ง authsrv แล้วหน้าจอจะปรากฏดังนี้

  #

```
# authsrv
authsrv# list
authsrv# adduser admin "Auth DB admin"
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# list
Report for users in database
user   group  longname          ok?   proto   last
------ ------ ------------------ ----- ------  -----
admin          Auth DB admin     ena   passw   never
authsrv# display admin
Report for user admin (Auth DB admin)
```

Authentication protocol: password

Flags: WIZARD

authsrv# ^D

EOT

จากตัวอย่างเป็นการเพิ่มผู้ใช้ ชื่อ admin เข้าสู่ระบบออเธนทิเคชั่นเซิร์ฟเวอร์ ซึ่งผู้ใช้ที่ชื่อ admin มีสิทธิในการเปลี่ยนแปลงข้อมูลในฐานข้อมูลภายในออเธนทิเคชั่นเซิร์ฟเวอร์ได้ ทั้งจากโฮสต์ตัวอื่นที่เป็นออเธนทิเคชั่นไคลเอนต์ หรือจากออเธนทิเคชั่นเซิร์ฟเวอร์โดยตรง

- เพิ่มเติมผู้ใช้ผ่านเครือข่ายจากออเธนทิเคชั่นไคลเอนต์

เราสามารถทำการลบ , เพิ่มเติม หรือทำการแก้ไขเปลี่ยนแปลงข้อมูลภายในฐานข้อมูลของออเธนทิเคชั่นเซิร์ฟเวอร์ผ่านเครือข่ายจากออเธนทิเคชั่นไคลเอนต์ได้จากโปรแกรม authmgr แต่ผู้ใช้ที่จะมีสิทธิเช่นนั้นได้จะต้องถูกกำหนดสิทธิเป็น wizard ที่ออเธนทิเคชั่นเซิร์ฟเวอร์ก่อน และก่อนใช้งานทุกครั้งจะต้องใช้คำสั่งล็อกอินก่อน จึงจะทำการจัดการกับฐานข้อมูลในออเธนทิเคชั่นเซิร์ฟเวอร์ได้

ตัวอย่างการเพิ่มรายชื่อของผู้ใช้ชื่อ s7014010 เข้าไปในฐานข้อมูล หลังจากเรียกใช้คำสั่ง authmgr จะปรากฏข้อความบนจอภาพดังนี้

```
# authmgr
  connecting host
  authmgr> login
  login name: admin
  password: *********
  USER logged in.
  authmgr> adduser s7014010 “Kawita”
  user add OK.
  authmgr> ena s7014010
  enable
  authmgr> quit
```

7.3.7.3 คำสั่งที่ใช้ในการจัดการออเธนทิเคชั่นเซิร์ฟเวอร์และออเธนทิเคชั่นไคลเอนต์

มีคำสั่งที่ใช้ดังนี้

- login

ล็อกอินเข้าสู่ออเธนทิเคชั่นเซิร์ฟเวอร์ เพื่อทำการแก้ไขข้อมูลผู้ใช้งาน จะใช้ในกรณีเรียกจากโฮสต์ออเธนทิเคชั่นไคลเอนต์

- adduser *username* [*longname*]
  เพิ่มเติมผู้ใช้ลงในฐานข้อมูลของออเธนทิเคชั่นเซิร์ฟเวอร์ โดยที่ *username* คือ ชื่อที่ผู้ใช้ใช้ทำการล็อกอิน และ *longname* คือชื่อจริงของผู้ใช้ ถ้าชื่อจริงของผู้ใช้มีช่องว่าง จะต้องใส่ไว้ในเครื่องหมายคำพูด
- deluser *username*
  ลบผู้ใช้ที่ใช้ชื่อในการล็อกอินว่า *username* ออกจากฐานข้อมูล
- display *username*
  แสดงข้อมูลทั้งหมดของผู้ใช้นั้น ๆ ที่มีอยู่ในฐานข้อมูลออกทางหน้าจอ ซึ่งประกอบไปด้วย สถานะ (status) , โปรโตคอลที่ใช้ในการติดต่อระหว่างออเธนทิเคชั่นเซิร์ฟเวอร์กับออเธนทิเคชั่นไคลเอนต์ (authentication protocol) และเวลาที่ผู้ใช้นั้นผ่านการรับรองครั้งสุดท้าย
- list [*group*]
  แสดงผู้ใช้ทั้งหมดที่มีอยู่ระบบ หรือผู้ใช้ที่อยู่ใน *group* นั้น ๆ
- enable *username*
  ทำให้ผู้ใช้นั้น ๆ สามารถใช้งานได้ (enable)
- disable *username*
  ทำให้ผู้ใช้นั้น ๆ ไม่สามารถใช้งานได้ (disable)
- group *username groupname*
  กำหนดกลุ่มให้กับผู้ใช้
- rename user *newname* [*longname*]
  เปลี่ยนชื่อที่ผู้ใช้ใช้ในการล็อกอิน
- proto *username auth-proto*
  กำหนดโปรโตคอลที่ใช้ในการติดต่อระหว่างออเธนทิเคชั่นไคลเอนต์ กับออเธนทิเคชั่นเซิร์ฟเวอร์ให้กับผู้ใช้ ซึ่งได้แก่ Security Dynamics' SecurID , Digital Pathways' SecureNet Key และ Racal Watchword
- wiz *username*
  เปลี่ยนสถานะของผู้ใช้นั้น ๆ เป็นผู้จัดการระบบ (wizard)
- unwiz *username*
  เปลี่ยนสถานะของผู้ใช้นั้น ๆ เป็นผู้ใช้ปกติ
- password [*username* [*pass*]]
  เปลี่ยนรหัสผ่านของผู้ใช้นั้น ๆ
- quit หรือ exit
  ออกจากระบบ
- ? หรือ help
  แสดงคำสั่งที่สามารถใช้งานได้ในออเธนทิเคชั่นเซิร์ฟเวอร์พร็อมพ์ (prompt)

บทที่ 8
การทดสอบการใช้งานไฟร์วอลล์ (Firewall)

เมื่อกล่าวถึงระบบไฟร์วอลล์ หัวใจสำคัญคือการทำแพ็กเก็ตฟิลเตอร์ริ่ง และการทำพร็อกซี เพื่อให้ได้มาซึ่งการควบคุมการไหลเข้าออกของข้อมูล (flow control) ภายในเครือข่าย แต่ในอีกแง่มุมหนึ่งเราจำเป็นต้องพิจารณาว่าระบบไฟร์วอลล์ให้การตอบสนองต่อความเร็วของข้อมูล (transfer rate) เท่าใดเมื่อเปรียบเทียบกับระบบเดิมที่ไม่มีการติดตั้งไฟร์วอลล์ ซึ่งเป็นที่แน่นอนว่าอัตราเร็วที่ได้ย่อมต้องลดลง แต่เพื่อให้ได้ความปลอดภัยสูงสุด ก็จำเป็นต้องยอมรับกับความเร็วที่ได้ ระบบไฟร์วอลล์ที่ดีจำเป็นต้องมีองค์ประกอบหลายอย่างรองรับ ทั้งในด้านฮาร์ดแวร์และซอฟท์แวร์ประกอบกัน

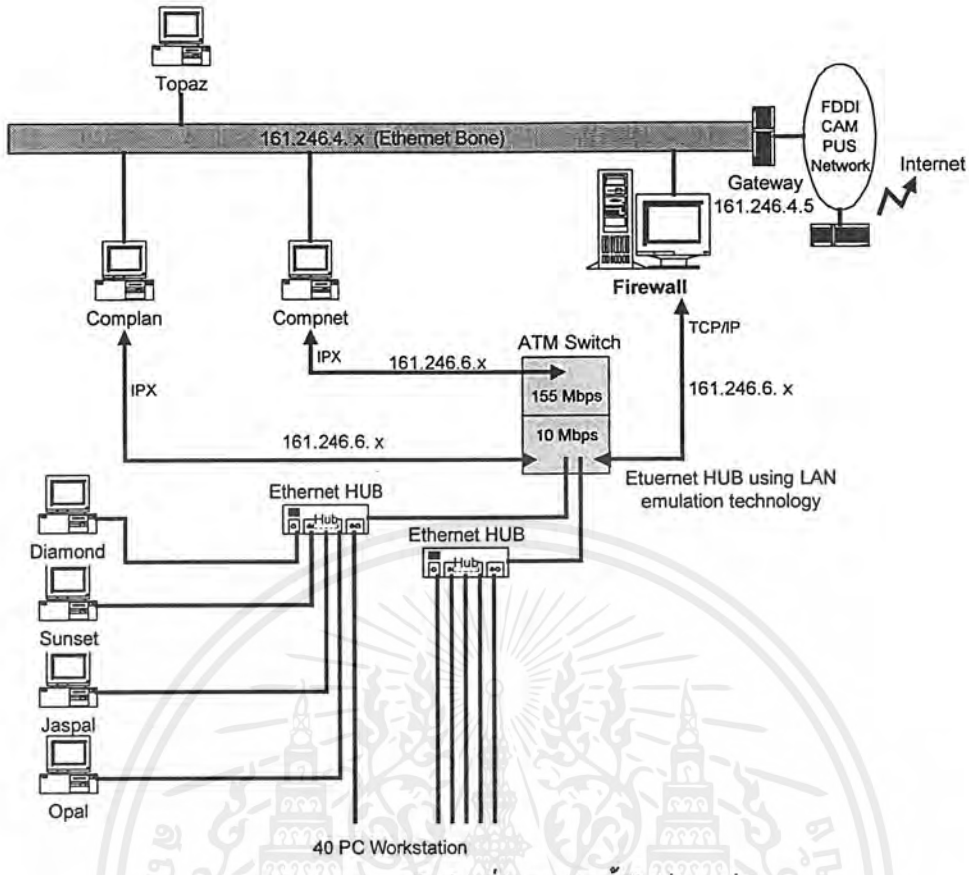
8.1 ปัจจัยที่ส่งผลถึงความเร็วของระบบไฟร์วอลล์

8.1.1 ฮาร์ดแวร์

ในด้านฮาร์ดแวร์ เราแบ่งปัจจัยที่มีผลต่อความเร็วออกเป็นสองอย่าง คือ

- ระบบเครือข่าย (network system)
  ถ้าเรากล่าวถึงความเร็วของระบบเครือข่ายแล้ว โดยทั่วไปจะพิจารณาถึงระบบการเชื่อมต่อที่มีในเครือข่ายนั้น ๆ เป็นหลัก เช่น เป็นระบบเครือข่าย FDDI (Fiber Distributed Data Interface) ให้ความเร็วในการโอนถ่ายข้อมูล 100 เมกะบิตต่อวินาที หรือ ATM (Asynchronous Transfer Mode) ให้ความเร็ว 155 เมกะบิตต่อวินาที โทเคนริง (token ring) ให้ความเร็ว 16 เมกะบิตต่อวินาที และอีเธอร์เน็ต (ethernet) ให้ความเร็ว 10 เมกะบิตต่อวินาที เป็นต้น

- หน่วยประมวลผลกลาง (microprocessor)
  เนื่องจากระบบไฟร์วอลล์จำเป็นต้องพึ่งซอฟท์แวร์ในการทำงาน ดังนั้น อัตราความเร็วในการประมวลผลซอฟท์แวร์จึงต้องขึ้นกับหน่วยประมวลผลเป็นหลัก กล่าวคือ ถ้าเปรียบเทียบซอฟท์แวร์ตัวเดียวกัน ทำงานบนแพล็ตฟอร์ม (platform) เดียวกัน ความเร็วในการประมวลผลเพื่อให้ได้ผลลัพธ์ของข้อมูลจะดูที่หน่วยประมวลผลกลางเป็นหลักเสมอ (ทั้งนี้รวมถึงปัจจัยรอบข้างด้วย เช่น หน่วยความจำหลัก , บัส (bus) ที่ใช้เชื่อมต่อภายใน , ความเร็วของฮาร์ดดิสก์ เป็นต้น) รูป 8.1 และ ตารางที่ 8.1 บอกให้ทราบถึงหน่วยประมวลผลที่มีอยู่ในโฮสต์แต่ละตัว



รูป 8.1 แสดงเครือข่ายที่ทำการติดตั้งไฟร์วอลล์

Hostname	OS Type	IP Address	Processor	NIC
Diamond	HP-UX 9.0	161.246.4.3	N/A	Ethernet 10Mbps
Opal	HP-UX 9.0	161.246.4.7	N/A	Ethernet 10Mbps
Sunset	Solaris 2.6	161.246.4.15	Pentium Pro	Ethernet 10Mbps
Topaz	Linux	161.246.4.12	486DX-33	Ethernet 10Mbps
Jasper	Linux	161.246.4.4	Pentium 133	Ethernet 10Mbps
Complan	Netware 3.21	161.246.4.1	486DXL-66	Ethernet 10Mbps
		161.246.6.1		Ethernet 10Mbps
Compnet	Netware 4.10	161.246.4.6	Pentium Pro	Ethernet 10Mbps
		161.246.6.6		ATM 155Mbps

ตาราง 8.1 แสดงรายละเอียดของโฮสต์แต่ละตัว

8.1.2 ซอฟท์แวร์

เนื่องจากระบบไฟร์วอลล์โดยทั่วไป ต้องอาศัยซอฟท์แวร์ในการจัดการระบบเครือข่าย โดยมีฮาร์ดแวร์เป็นตัวรองรับอีกทีหนึ่ง การที่เราจะตัดสินว่าซอฟท์แวร์ที่นำมาใช้ในระบบไฟร์วอลล์ของเรามีประสิทธิภาพมากหรือน้อย ขึ้นอยู่กับระบบเครือข่ายที่เรามีอยู่ และขนาดของระบบเครือข่ายนั้น ๆ ด้วย

โดยส่วนใหญ่ซอฟท์แวร์ที่จะนำมาดำเนินการเป็นระบบไฟร์วอลล์จะเป็นซอฟท์แวร์ที่มาในลักษณะของแพ็กเก็ต โครงงานนี้ใช้ซอฟท์แวร์ที่เป็นแชร์แวร์ของบริษัท TIS (Trusted Information Systems) ชื่อ FWTK (Firewall Toolkit) ซึ่งภายใน FWTK จะประกอบไปด้วยซอฟท์แวร์ย่อย ๆ ดังนี้

- tn-gw ทำหน้าที่เป็นเทลเน็ตพร็อกซีเซิร์ฟเวอร์ (TELNET proxy server)
- ftp-gw ทำหน้าที่เป็นเอฟทีพีพร็อกซีเซิร์ฟเวอร์ (FTP proxy server)
- rlogin-gw ทำหน้าที่เป็นอาร์ล็อกอินพร็อกซีเซิร์ฟเวอร์ (rlogin proxy server)
- http-gw ทำหน้าที่เป็นเอชทีทีพีพร็อกซีเซิร์ฟเวอร์ (http proxy server)
- plug-gw ทำหน้าที่เป็นพร็อกซีของบริการอื่น ๆ ที่ไม่ได้อยู่ในแพ็กเก็ตโปรแกรม
- x-gw ทำหน้าที่เป็นเอ็กซ์โปรโตคอลพร็อกซีเซิร์ฟเวอร์ (X-protocol proxy server)
- smap และ smapd ทำหน้าที่เป็นเมล์พร็อกซีเซิร์ฟเวอร์ (mail proxy server)
- netacl ทำหน้าที่ควบคุมการใช้งาน (access control) ให้กับโปรแกรม telnetd และ ftpd (จะสังเกตเห็นว่าทั้งโปรแกรม telnetd และ ftpd ไม่ใช่เดมอน (daemon) ของพร็อกซีเซิร์ฟเวอร์)

8.2 ทดสอบระบบไฟร์วอลล์ที่ได้ทำการติดตั้ง

เราจะแบ่งการทดสอบระบบไฟร์วอลล์ออกเป็นบริการย่อย ๆ โดยจะมีบริการบางตัวที่จะต้องทำการทดสอบอัตราการถ่ายเทข้อมูล (transfer rate) นั่นคือ บริการเอฟทีพี และบริการเอชทีทีพี ทั้งก่อนและหลังจากการติดตั้งไฟร์วอลล์ โดยที่บริการอื่น ๆ นอกเหนือจากบริการดังกล่าว เราจะไม่พิจารณาด้วยสาเหตุที่ว่า บริการที่เหลือโดยส่วนใหญ่แล้วเราจะไม่สนใจเวลาในการตอบสนอง (response time) บริการดังกล่าวคือ เทลเน็ต , เมล์ , อาร์ล็อกอิน

8.2.1 วิธีการทดสอบ tn-gw

- ปัญหาที่เกิดขึ้นกับบริการเทลเน็ตก่อนทำการติดตั้งระบบไฟร์วอลล์
  - ➢ ผู้ใช้จากภายนอกสามารถทำการเทลเน็ตเข้ามาในเครือข่ายได้อย่างอิสระ ไม่มีการป้องกันใด ๆ ทั้งสิ้น เป็นช่องทางให้ผู้ไม่ประสงค์ดีบุกรุกเข้ามาในระบบเครือข่ายของเราได้ง่าย
  - ➢ ไม่มีการทำรายงานการใช้บริการเทลเน็ตของผู้ใช้ต่าง ๆ ทั้งภายนอกและภายในเครือข่าย ทำให้ผู้ดูแลระบบเครือข่ายไม่ทราบความเป็นไปของระบบเกี่ยวกับบริการดังกล่าว

- เป้าหมายในการทดสอบ

- ต้องการให้โฮสต์และไคลเอนต์ภายในไฟร์วอลล์ทั้งหมด (เครือข่ายหมายเลข 161.246.6.X) สามารถเทลเน็ตไปยังโฮสต์อื่น ๆ ภายนอกระบบได้
- ไม่ต้องการให้โฮสต์จากภายนอกไฟร์วอลล์ใช้เทลเน็ตเข้ามาในไฟร์วอลล์

ข้อมูลในไฟล์ /usr/local/etc/netperm-table ได้กำหนดไว้ดังนี้

*tn-gw: permit-hosts 161.246.6.**

*tn-gw: deny-hosts **

- ทดสอบผลที่เกิดขึ้นในกรณีอนุญาตการเชื่อมต่อ
  - หลังจากใช้คำสั่งเทลเน็ต จากโฮสต์ในเครือข่ายหมายเลข 161.246.6.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

```
$ telnet firewall
Trying...
Connected to firewall.ce.kmitl.ac.th.
Escape character is '^]'.
Local flow control off
##########################################################
##                Welcome to FireWall System            ##
##########################################################
tn-gw-> c chaokhun.kmitl.ac.th
Trying 161.246.10.21 port 23...
Connected to chaokhun.kmitl.ac.th.
UNIX(r) System V Release 4.0 (Chaokhun)
login:
```

ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

Mar 19 03:00:58 firewall tn-gw[214]: permit host=diamond.ce.kmitl.ac.th/161.246.6.3 use of gateway

Mar 19 03:01:44 firewall tn-gw[214]: permit host=diamond.ce.kmitl.ac.th/161.246.6.3 destination=161.246.10.21

Mar 19 03:01:44 firewall tn-gw[214]: connected host=diamond.ce.kmitl.ac.th/161.246.6.3 destination=chaokhun.kmitl.ac.th

Mar 19 03:03:44 firewall tn-gw[214]: exit host=diamond.ce.kmitl.ac.th/161.246.6.3 dest=chaokhun.kmitl.ac.th in=106 out=44 user=unauth duration=166

- ทดสอบผลที่เกิดขึ้นในกรณีที่ไม่อนุญาตให้มีการใช้งาน
  - หลังจากใช้คำสั่งเทลเน็ต จากโฮสต์ในเครือข่ายหมายเลข 161.246.10.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

$ telnet firewall.ce.kmitl.ac.th

Trying 161.246.4.160...

Connected to firewall.ce.kmitl.ac.th.

Escape character is '^]'.

##

 # This System not allow you to access without authenticated #

##

Connection closed by foreign host.

ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

Mar 19 03:07:57 firewall tn-gw[231]: deny host=chaokhun.kmitl.ac.th/161.246.10.2

1 use of gateway

- หลังจากทำการติดตั้งเทลเน็ตเกตเวย์แล้ว เราสามารถแก้ปัญหาได้ดังต่อไปนี้
  - สามารถกำหนดหมายเลขเครือข่ายที่เราไม่ต้องการให้ล่วงล้ำเข้ามาใช้บริการเทลเน็ตภายในเครือข่ายของเรา โดยการใช้คำสั่งดังนี้ในไฟล์ /usr/local/etc/netperm-table

*tn-gw: permit-hosts 161.246.4.**

tn-gw: deny-hosts unknown

คำสั่งดังกล่าวคือ ยอมให้เครือข่ายหมายเลข 161.246.4.X ทำการเทลเน็ตเข้ามาในเครือข่ายได้ แต่ไม่อนุญาตให้โฮสต์ที่ไม่มีชื่ออยู่ในโดเมนเนมเซิร์ฟเวอร์ทำการเทลเน็ตเข้ามาในเครือข่าย

  - มีการบันทึกผลการเข้าออกของบริการเทลเน็ต ของหมายเลขเครือข่ายทุกเครือข่าย ทั้งภายในและภายนอก ทำให้ผู้ดูแลระบบสามารถนำข้อมูลเหล่านั้นมาจัดทำเป็นรายงานได้ และสามารถดูความเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นกับระบบได้
  - สามารถกำหนดผู้ใช้บริการเทลเน็ตที่เข้ามาภายในระบบเครือข่าย นอกเหนือไปจากชื่อผู้ใช้ที่มีอยู่ในโฮสต์แต่ละตัวภายในระบบเครือข่าย ซึ่งวิธีดังกล่าวเราเรียกว่า การทำการรับรองผู้ใช้ (user authentication)

8.2.2 วิธีการทดสอบ ftp-gw

- ปัญหาที่เกิดขึ้นกับบริการเอฟทีพีก่อนทำการติดตั้งระบบไฟร์วอลล์
  - ผู้ใช้จากภายนอกสามารถทำการเอฟทีพีได้อย่างอิสระ ไม่มีการป้องกันใด ๆ ทั้งสิ้น ถ้าโฮสต์ที่อยู่ในเครือข่ายของเราไม่มีความปลอดภัยเพียงพอ อาจเป็นช่องทางให้ผู้ไม่ประสงค์ดีบุกรุกเข้ามาทำการเปลี่ยนแปลงแก้ไขข้อมูลภายในระบบของเราได้
  - ไม่มีการทำรายงานการใช้บริการเอฟทีพีของผู้ใช้ต่าง ๆ ทั้งภายนอกและภายในเครือข่าย ทำให้ผู้ดูแลระบบเครือข่ายไม่ทราบความเป็นไปของระบบเกี่ยวกับบริการดังกล่าว

- เป้าหมายในการทดสอบ
  - ต้องการให้โฮสต์และไคลเอนต์ภายในไฟร์วอลล์ทั้งหมด (เครือข่ายหมายเลข 161.246.6.X) สามารถเอฟทีพีไปยังโฮสต์อื่น ๆ ภายนอกระบบได้
  - ไม่ต้องการให้โฮสต์จากภายนอกไฟร์วอลล์ใช้เอฟทีพีเข้ามาในไฟร์วอลล์ ข้อมูลในไฟล์ /usr/local/etc/netperm-table ได้กำหนดไว้ดังนี้

    *ftp-gw: permit-hosts 161.246.6.**

    *ftp-gw: deny-hosts **

- ทดสอบผลที่เกิดขึ้นในกรณีอนุญาตการเชื่อมต่อ
  - หลังจากใช้คำสั่งเอฟทีพี จากโฮสต์ในเครือข่ายหมายเลข 161.246.6.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

    $ ftp firewall

    Connected to firewall.ce.kmitl.ac.th.

    220-##

    220-## Welcome to FireWall System ##

    220 ##

    Name (firewall:s9013265): s9013265@diamond.ce.kmitl.ac.th

    331-(----GATEWAY CONNECTED TO diamond.ce.kmitl.ac.th----)

    331-(220-Welcome to diamond FTP server.)

    331-(220-)

    331-(220-This server does not permit anonymous FTP.)

    331-(220-Our anonymous FTP server is ftp.ce.kmitl.ac.th.)

    331-(220-)

    331-(220 diamond.ce.kmitl.ac.th FTP server (Version wu-2.4.2-academ [BETA-13](1)

    Wed Mar 12 22:05:14 TST 1997) ready.)

331 Password required for s9013265.

Password:

230 User s9013265 logged in.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp>

ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

Mar 19 03:15:14 firewall ftp-gw[259]: permit host=topaz.ce.kmitl.ac.th/161.246
.6.12 use of gateway
Mar 19 03:15:34 firewall ftp-gw[259]: permit host=topaz.ce.kmitl.ac.th/161.246
.6.12 connect to diamond.ce.kmitl.ac.th
Mar 19 03:17:06 firewall ftp-gw[259]: exit host=topaz.ce.kmitl.ac.th/161.246.6
.12 cmds=5 in=0 out=0 user=unauth duration=112

- ทดสอบผลที่เกิดขึ้นในกรณีที่ไม่อนุญาตให้มีการใช้งาน
  - หลังจากใช้คำสั่งเอฟทีพี จากโฮสต์ในเครือข่ายหมายเลข 161.246.10.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

    $ ftp firewall.ce.kmitl.ac.th

    Connected to firewall.ce.kmitl.ac.th.

    500-##

    500-# This System not allow you to access without authenticated #

    500-##

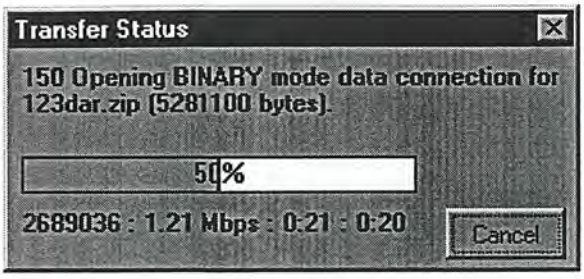
    500

    ftp>

  ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

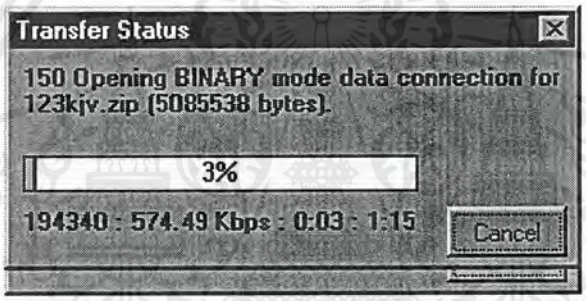
  Mar 19 03:22:23 firewall ftp-gw[267]: deny host=chaokhun.kmitl.ac.th/161.246.10.
  21 use of gateway

- ทดสอบอัตราการถ่ายเทข้อมูลด้วยโปรแกรมดับเบิ้ลยูเอส-เอฟทีพี (WS-FTP)
  - ในกรณีที่ไม่มีการติดตั้งไฟร์วอลล์ สามารถให้ความเร็วในการถ่ายเทข้อมูลจากโฮสต์ไคลเอนต์ไปยังโฮสต์เอฟทีพีของเนคเทค 1.2 เมกะบิตต่อวินาที



รูป 8.2 แสดงความเร็วในการถ่ายเทข้อมูลจากโฮสต์ไคลเอนต์ไปยังโฮสต์เอฟทีพีของเนคเทคกรณีที่ไม่มีการติดตั้งไฟร์วอลล์

- ในกรณีที่มีการติดตั้งไฟร์วอลล์ และโปรแกรมจำเป็นต้องใช้ไฟร์วอลล์เป็นพร็อกซี สามารถให้ความเร็วในการถ่ายเทข้อมูลจากโฮสต์ไคลเอนต์ไปยังโฮสต์เอฟทีพีของเนคเทค 575 กิโลบิตต่อวินาที



รูป 8.3 แสดงความเร็วในการถ่ายเทข้อมูลจากโฮสต์ไคลเอนต์ไปยังโฮสต์เอฟทีพีของเนคเทคกรณีที่มีการติดตั้งไฟร์วอลล์

- หลังจากทำการติดตั้งเอฟทีพีเกตเวย์แล้ว เราสามารถแก้ปัญหาได้ดังต่อไปนี้
  - สามารถกำหนดหมายเลขเครือข่ายที่เราไม่ต้องการให้ล่วงล้ำเข้ามาใช้บริการเอฟทีพีภายในเครือข่ายของเรา โดยการใช้คำสั่งดังนี้ในไฟล์ /usr/local/etc/netperm-table

    ftp-gw: *permit-hosts 161.246.6.**

    ftp-gw: *deny-hosts **

  - มีการบันทึกผลการเข้าออกของบริการเอฟทีพี ของหมายเลขเครือข่ายทุกเครือข่าย ทั้งภายในและภายนอก ทำให้ผู้ดูแลระบบสามารถนำข้อมูลเหล่านั้นมาจัดทำเป็นรายงานได้ และสามารถดูความเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นกับระบบได้ โดยใช้ออพชั่นดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

    ftp-gw: *deny-hosts * log { retr stor }*

- สามารถกำหนดผู้ใช้บริการเอฟทีพีที่เข้ามาภายในระบบเครือข่าย นอกเหนือไปจากชื่อผู้ใช้ที่มีอยู่ในโฮสต์แต่ละตัวภายในระบบเครือข่าย ซึ่งวิธีดังกล่าวเราเรียกว่า การทำการรับรองผู้ใช้ (user authentication) โดยใช้ออพชั่นดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

ftp-gw: permit-hosts 161.246.6. -authall*

8.2.3 วิธีการทดสอบเอชทีทีพีพร็อกซี

- ปัญหาที่เกิดขึ้นกับบริการเอชทีทีพีก่อนทำการติดตั้งระบบไฟร์วอลล์
  - ผู้ใช้จากภายในสามารถใช้งานเว็บภายนอกได้อย่างอิสระ ไม่สามารถจำกัดโฮสต์ได้ ทำให้เกิดความหนาแน่นของระบบเครือข่าย
  - ผู้ใช้ภายนอกสามารถใช้งานเว็บภายในได้อย่างเป็นอิสระเช่นกัน ซึ่งข้อมูลบางอย่างอาจเป็นข้อมูลที่ใช้เป็นการภายในเท่านั้น
  - ไม่มีการทำรายงานการใช้บริการเอชทีทีพีของผู้ใช้ต่าง ๆ ทั้งภายนอกและภายในเครือข่าย ทำให้ผู้ดูแลระบบเครือข่ายไม่ทราบความเป็นไปของระบบเกี่ยวกับบริการดังกล่าว
- เป้าหมายในการทดสอบ
  - ต้องการควบคุมการเข้าถึงข้อมูลของโฮสต์ กล่าวคือเราสามารถควบคุมได้ว่าโฮสต์ที่จะใช้งานพร็อกซีจะต้องเป็นโฮสต์ที่ที่เราอนุญาตไว้ในไฟล์ /usr/local/etc/netperm-table
  - ข้อมูลในไฟล์ /usr/local/etc/netperm-table

    http-gw: hosts 161.246.4.*
- ทดสอบผลที่เกิดขึ้นในกรณีอนุญาตการเชื่อมต่อ
  - หลังจากใช้โปรแกรมเบราเซอร์ จากโฮสต์ในเครือข่ายหมายเลข 161.246.4.X ไปที่ไฟร์วอลล์ ไฟล์ /usr/adm/syslog จะมีการรายงานผลดังนี้

    Mar 19 04:01:35 firewall http-gw[358]: log host=jackdawson.ce.kmitl.ac.th/161.246.4.151 protocol=HTTP cmd=get dest=www.yahoo.com path=/ad-imx/theglobe/theglobe-burn_anim-02-05.gif

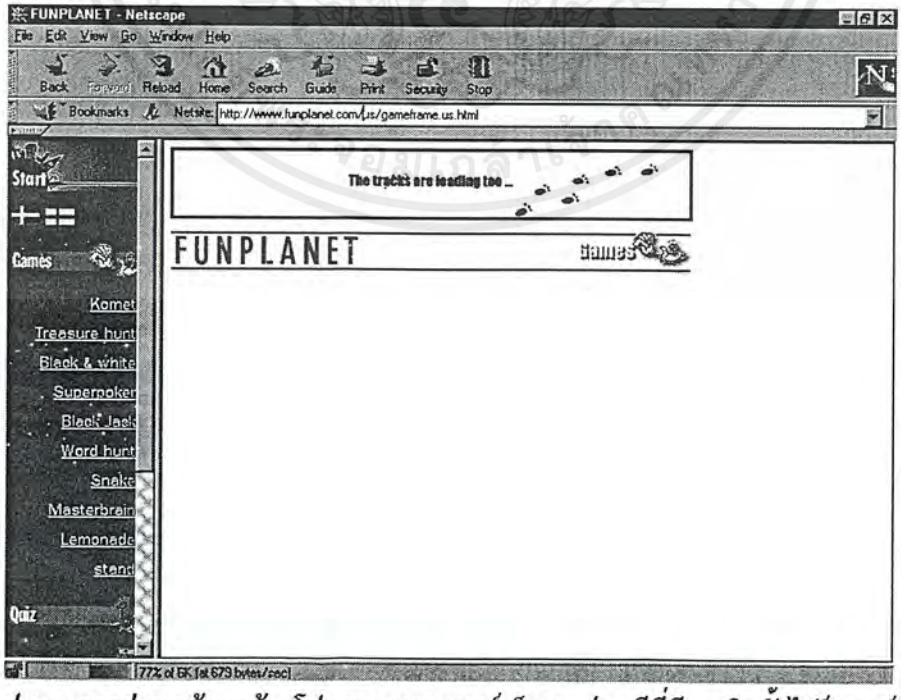
- ทดสอบผลที่เกิดขึ้นในกรณีที่ไม่อนุญาตให้มีการใช้งาน
  - หลังจากใช้โปรแกรมเบราเซอร์ จากโฮสต์ในเครือข่ายหมายเลข 161.246.34.X ไปที่ไฟร์วอลล์ จะมีรายงานมายังไฟล์ /usr/adm/syslog ดังนี้

Mar 19 04:03:39 firewall http-gw[365]: deny host=blue03.crsc.kmitl.ac.th/161.246.34.38 use of gateway

- ทดสอบอัตราการถ่ายเทข้อมูลด้วยโปรแกรมเบราเซอร์เน็ตสเคป
  - ➢ ในกรณีที่ไม่มีการติดตั้งไฟร์วอลล์ สามารถให้ความเร็วในการถ่ายเทข้อมูลจากโฮสต์ funplanet.com มายังไคลเอนต์ด้วยความเร็วประมาณ 700 ไบต์ต่อวินาที

รูป 8.4 การถ่ายเทข้อมูลด้วยโปรแกรมเบราเซอร์เน็ตสเคปกรณีที่ไม่มีการติดตั้งไฟร์วอลล์

- ในกรณีที่มีการติดตั้งไฟร์วอลล์ สามารถให้ความเร็วในการถ่ายเทข้อมูลจากโฮสต์เดียวกันด้วยความเร็วประมาณ 600 ไบต์ต่อวินาที



รูป 8.5 การถ่ายเทข้อมูลด้วยโปรแกรมเบราเซอร์เน็ตสเคปกรณีที่มีการติดตั้งไฟร์วอลล์

- หลังจากทำการติดตั้งเอชทีทีพีเกตเวย์แล้ว เราสามารถแก้ปัญหาได้ดังต่อไปนี้
  - สามารถกำหนดโฮสต์ไคลเอนต์ภายในเครือข่ายที่จะสามารถทำการใช้งานเว็บจากภายนอกได้ ทำให้ความหนาแน่นของเครือข่ายลดลง โดยการใช้คำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

    http-gw: hosts 161.246.4.15 161.246.4.16 161.246.4.17
  - มีการบันทึกผลการเข้าออกของบริการเอชทีทีพี ของหมายเลขเครือข่ายทุกเครือข่าย ทั้งภายในและภายนอก ทำให้ผู้ดูแลระบบสามารถนำข้อมูลเหล่านั้นมาจัดทำเป็นรายงานได้ และสามารถดูความเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นกับระบบได้
  - สามารถกำหนดเอชทีทีพีเซิร์ฟเวอร์ภายในเครือข่ายที่บุคคลภายนอกสามารถใช้งานได้ โดยกำหนดด้วยคำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

    http-gw: default-httpd diamond.ce.kmitl.ac.th

8.2.4 วิธีการทดสอบ rlogin-gw

- ปัญหาที่เกิดขึ้นกับบริการอาร์ล็อกอินก่อนทำการติดตั้งระบบไฟร์วอลล์
  - ผู้ใช้จากภายนอกสามารถใช้งานอาร์ล็อกอินได้อย่างอิสระ ไม่มีการป้องกันใด ๆ ทั้งสิ้น อาจเป็นช่องทางให้ผู้ไม่ประสงค์ดีบุกรุกเข้ามาในเครือข่ายของเราได้
  - ไม่มีการทำรายงานการใช้บริการอาร์ล็อกอินของผู้ใช้ต่าง ๆ ทั้งภายนอกและภายในเครือข่าย ทำให้ผู้ดูแลระบบเครือข่ายไม่ทราบความเป็นไปของระบบเกี่ยวกับบริการดังกล่าว
- เป้าหมายในการทดสอบ
  - ต้องการให้โฮสต์และไคลเอนต์ภายในไฟร์วอลล์ทั้งหมด (เครือข่ายหมายเลข 161.246.6.X) สามารถอาร์ล็อกอินไปยังโฮสต์อื่น ๆ ภายนอกระบบได้
  - ไม่ต้องการให้โฮสต์จากภายนอกไฟร์วอลล์ทำการอาร์ล็อกอินเข้ามาในไฟร์วอลล์ ข้อมูลในไฟล์ /usr/local/etc/netperm-table ได้กำหนดไว้ดังนี้

    *rlogin-gw: permit-hosts 161.246.6.**

    *rlogin-gw: deny-hosts **
- ทดสอบผลที่เกิดขึ้นในกรณีอนุญาตการเชื่อมต่อ
  - หลังจากใช้คำสั่งอาร์ล็อกอิน จากโฮสต์ในเครือข่ายหมายเลข 161.246.6.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

$ rlogin firewall

##

Welcome to FireWall System

##

rlogin-gw-> c s9013265@chaokhun.kmitl.ac.th

Trying s9013265@161.246.10.21...

Password:

ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

Mar 19 04:31:09 firewall rlogin-gw[444]: permit host=diamond.ce.kmitl.ac.th/161.

246.4.3 connect to 161.246.10.21

- ทดสอบผลที่เกิดขึ้นในกรณีที่ไม่อนุญาตให้มีการใช้งาน
  - หลังจากใช้คำสั่งอาร์ล็อกอิน จากโฮสต์ในเครือข่ายหมายเลข 161.246.10.X ไปที่ไฟร์วอลล์ จะปรากฏข้อความดังนี้

$ rlogin firewall.ce.kmitl.ac.th

Trying 161.246.4.160...

Connected to firewall.ce.kmitl.ac.th.

Escape character is '^]'.

##

 # This System not allow you to access without authenticated #

##

Connection closed by foreign host.

ผลที่รายงานกลับไปยังไฟล์ /usr/adm/syslog มีดังนี้

Mar 19 04:38:02 firewall rlogin-gw[444]: deny host=blue03.crsc.kmitl.ac.th/161.

246.34.38 connect to 161.246.10.21 use of gateway

- หลังจากทำการติดตั้งอาร์ล็อกอินเกตเวย์แล้ว เราสามารถแก้ปัญหาได้ดังต่อไปนี้
  - สามารถกำหนดหมายเลขเครือข่ายที่เราไม่ต้องการให้ล่วงล้ำเข้ามาใช้บริการอาร์ล็อกอินภายในเครือข่ายของเรา โดยการใช้คำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

*rlogin-gw: permit-hosts 161.246.10.**

rlogin-gw: deny-hosts unknown

- มีการบันทึกผลการเข้าออกของบริการอาร์ล็อกอิน ของหมายเลขเครือข่ายทุกเครือข่าย ทั้งภายในและภายนอก ทำให้ผู้ดูแลระบบสามารถนำข้อมูลเหล่านั้นมาจัดทำเป็นรายงานได้ และสามารถดูความเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นกับระบบได้ โดยการใช้ออพชั่นดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

  rlogin-gw: permit-hosts 161.246.10. -log*

- สามารถกำหนดผู้ใช้บริการอาร์ล็อกอินที่เข้ามาภายในระบบเครือข่าย นอกเหนือไปจากชื่อผู้ใช้ที่มีอยู่ในโฮสต์แต่ละตัวภายในระบบเครือข่าย ซึ่งวิธีดังกล่าวเราเรียกว่า การทำการรับรองผู้ใช้ (user authentication) โดยการใช้คำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

  rlogin-gw: permit-hosts 161.246.10. -auth*

8.2.5 วิธีการทดสอบ smap และ smapd

- ปัญหาที่เกิดขึ้นก่อนการติดตั้งไฟร์วอลล์
  - ไม่สามารถควบคุมจำนวน และขนาดของเมล์ที่จะเข้ามาสู่ระบบ ของผู้ใช้แต่ละคนได้ บุคคลภายนอกอาจทำการละเมิดสิทธิโดยการส่งเมล์มาหาผู้ใช้คนใดคนหนึ่งเป็นจำนวนมาก (mailbomb) จนผู้ใช้นั้นไม่สามารถรับเมล์จากบุคคลอื่นได้ (quota exceed)
  - โปรแกรมที่ใช้ในการส่งเมล์ในโฮสต์แต่ละตัวอาจมีข้อผิดพลาด (bug) อยู่ โดยส่วนใหญ่จะใช้โปรแกรม sendmail ซึ่งโปรแกรมดังกล่าวในปัจจุบันยังคงมีข้อผิดพลาดอยู่บางจุด (มีฟังก์ชั่นภายใน sendmail มาก เป็นช่องทางให้ผู้ไม่ประสงค์ดีใช้ฟังก์ชั่นที่มีอยู่เป็นประโยชน์ในการบุกรุกและทำลายระบบได้)
- เป้าหมายในการทดสอบ
  - ต้องการให้โฮสต์และไคลเอนต์ภายนอกไฟร์วอลล์ทั้งหมด สามารถส่งเมล์ไปยังโฮสต์อื่น ๆ ภายในไฟร์วอลล์ได้
  - ตัวที่ทำหน้าที่รับเมล์จริง ๆ คือ โฮสต์ไดมอนด์
- ทดสอบผลที่เกิดขึ้นเมื่อเกิดการติดต่อ

  ทำการส่งเมล์จากโฮสต์เจ้าคุณ ไปที่โฮสต์ไดมอนด์ โดยเราจะทำการเทลเน็ตไปที่พอร์ตของไฟร์วอลล์แทน แล้วจะดูผลว่าไฟร์วอลล์ทำการฟอร์เวิร์ด (forward) เมล์ไปที่โฮสต์ไดมอนด์จริงหรือไม่

  $ telnet firewall smtp

  Trying...

  Connected to firewall.ce.kmitl.ac.th.

  Escape character is '^]'.

  220 firewall.ce.kmitl.ac.th SMTP/smap Ready.

HELO

250 Charmed, I'm sure.

MAIL from:s7014010@kmitl.ac.th

250 s7014010@kmitl.ac.th... Sender Ok

RCPT to:s9013265@diamond.ce.kmitl.ac.th

250 s9013265@diamond.ce.kmitl.ac.th OK

DATA

354 Enter mail, end with "." on a line by itself

hello

.

250 Mail accepted

QUIT

221 Closing connection

Connection closed by foreign host.

$

ผลที่รายงานไปที่ไฟล์ /usr/adm/syslog ของไฟร์วอลล์

Feb 3 18:06:14 firewall smap[100]: connect host=chaokhun.kmitl.ac.th/161.246.10.21

Feb 3 18:06:14 firewall smap[100]: host=chaokhun.ce.kmitl.ac.th/161.246.10.21 bytes=2

268 from=<s7014010@.kmitl.ac.th> to=<s9013265@diamond.ce.kmitl.ac.th>

Feb 3 18:06:14 firewall smap[100]: exiting host=chaokhun.kmitl.ac.th/161.246.5.3

bytes=2268

Feb 3 18:06:20 firewall smapd[101]: delivered file=sma000100 pid=102 code=0

หลังจากที่โฮสต์ไฟร์วอลล์ได้รับเมล์จากโฮสต์เจ้าคุณแล้ว จะทำการฟอร์เวิร์ดเมล์ดังกล่าวไปที่โฮสต์ไดมอนด์อย่างถูกต้อง โดยเฮดเดอร์ของเมล์ที่ถูกส่งมาจะมีรายละเอียดดังนี้

$ more s9013265

From s7014010@kmitl.ac.th Thu Mar 19 11:13:41 TST 1998

Received: from firewall.ce.kmitl.ac.th (firewall.ce.kmitl.ac.th [161.246.4.160])

by diamond.ce.kmitl.ac.th (8.8.8/8.8.8) with ESMTP id LAA22652

for <s9013265@diamond.ce.kmitl.ac.th>; Thu, 19 Mar 1998 11:13:40 +0700 (

TST)

From: s7014010@kmitl.ac.th

Received: (from mail@localhost)

by firewall.ce.kmitl.ac.th (8.8.5/8.8.5) id LAA01080

for s9013265@diamond.ce.kmitl.ac.th; Thu, 19 Mar 1998 11:17:54 -0200

Date: Thu, 19 Mar 1998 11:17:54 -0200

Message-Id: <199803191317.LAA01080@firewall.ce.kmitl.ac.th>

X-Authentication-Warning: firewall.ce.kmitl.ac.th: mail set sender to s7014010@k

mitl.ac.th using -f

Received: from chaokhun.kmitl.ac.th(161.246.10.21) by firewall.ce.kmitl.ac.th vi

a smap (V2.0)

id xma001078; Thu, 19 Mar 98 11:17:43 -0200

Status: RO

X-Status:

Hello

จะเห็นได้ว่า เฮดเดอร์ของเมล์มีการระบุว่าโฮสต์เจ้าคุณส่งเมล์มาที่ไฟร์วอลล์ แต่ไฟร์วอลล์จะฟอร์เวิร์ดเมล์ไปที่ไดมอนด์อีกทีหนึ่ง

- หลังจากทำการติดตั้งโปรแกรม smap และ smapd แล้ว เราสามารถแก้ปัญหาได้ดังนี้
  - สามารถจำกัดขนาดของข้อมูลสูงสุดที่สามารถส่งเข้ามาในระบบได้ โดยการใช้คำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

    smap: maxbyte 1048576
  - สามารถกำหนดจำนวนสูงสุดในการรับเมล์แต่ละครั้งได้ ช่วยให้ความหนาแน่นภายในเครือข่ายลดลง โดยการใช้คำสั่งดังนี้ที่ไฟล์ /usr/local/etc/netperm-table

    smap: maxrecip 4000
  - โปรแกรม smap และ smapd มีขนาดไม่ใหญ่มาก และได้ทำการตัดฟังก์ชั่นที่ไม่จำเป็นออกไป ทำให้ข้อผิดพลาดของโปรแกรมที่มีอยู่ลดลง

บทที่ 9
การเขียนสคริปต์เพื่อเปลี่ยนแปลง ค่าการทำงานของระบบไฟร์วอลล์

เนื่องจากการทำงานกับระบบไฟร์วอลล์ จำเป็นจะต้องมีการเปลี่ยนแปลงแก้ไขการทำงานของระบบอยู่ตลอดเวลา โครงงานนี้ใช้โปรแกรมไฟร์วอลล์ทูลคิท (Firewall Toolkit) ซึ่งโปรแกรมดังกล่าวทำการแก้ไขการทำงานของระบบที่ไฟล์ /usr/local/etc/netperm-table ในบางครั้ง การแก้ไขอาจเป็นไปด้วยยากลำบาก และไม่สะดวกต่อผู้ดูแลระบบ เนื่องจากจะต้องทราบไวยากรณ์ (syntax) ของไฟล์ดังกล่าว จึงได้ทำการพัฒนาโปรแกรมสคริปต์เพื่อช่วยให้ผู้ดูแลระบบสามารถทำการแก้ไขค่าการทำงาน (configuration) ได้สะดวกยิ่งขึ้น

9.1 การทำงานโดยรวมของเมนูหลัก

เนื่องจากบริการต่าง ๆ ซึ่งได้แก่ เทลเน็ตพร็อกซี , เอฟทีพีพร็อกซี , อาร์ล็อกอินพร็อกซี , เอชทีทีพีพร็อกซี , เทลเน็ต และเอฟทีพีกับไฟร์วอลล์ จะมีการทำงานโดยรวม 3 ประการหลัก คือ การเพิ่มเติม (add) , การลบ (delete) และการแก้ไข (edit) เช่น เพิ่มเติมรายชื่อโฮสต์ หรือหมายเลขเครือข่ายที่สามารถใช้บริการเทลเน็ตพร็อกซีลงในไฟล์ /usr/local/etc/netperm-table หรือทำการลบเครือข่ายที่ไม่ต้องการให้ใช้บริการเอฟทีพีพร็อกซีออกจากไฟล์ /usr/local/etc/netperm-table จากเมนูหลัก ให้เลือกบริการที่ต้องการทำการเปลี่ยนแปลงแก้ไข โดยหน้าจอจะปรากฏข้อความดังต่อไปนี้

Main Menu
1. Telnet-gw
2. FTP-gw
3. Rlogin-gw
4. HTTP-gw
5. Telnet
6. FTP
7. EXIT

Choice (1-7) : 1

รูป 9.1 เมนูหลัก

เมื่อได้ทำการเลือกบริการใดบริการหนึ่งแล้ว จะมีข้อความดังต่อไปนี้

Telnet-gw
1. *Add*
2. *Delete*
3. *Exit*
4. *Retuen to Main Menu*

Choice (1-4) :

รูป 9.2 เมนูย่อยของบริการแต่ละตัว

9.2 การเพิ่มเติม (add)

จากเมนูย่อย ดังรูป 9.2 เมื่อทำการเลือกหมายเลข 1 (add) จะเป็นการเพิ่มหมายเลขเครือข่ายหรือชื่อโฮสต์ที่ต้องการให้ไฟร์วอลล์อนุญาตการใช้งานบริการที่ได้ทำการเลือกไว้จากเมนูหลัก ตัวอย่างเช่น ถ้าเราทำการเลือกบริการเทลเน็ตพร็อกซี (รูป 9.1 หมายเลข 1) จะปรากฏเมนูย่อย แล้วเลือกหมายเลข 1 จะปรากฏตารางแสดงรายชื่อของโฮสต์ และหมายเลขเครือข่ายที่มีอยู่เดิม ในไฟล์ /usr/local/etc/netperm-table ดังนี้

Network	Permitted	Authentication	Log
1. *Diamond*	*No*	*No*	*No*
2. *161.246.4.**	*Yes*	*No*	*No*

Add to line number :

รูป 9.3 ตารางแสดงรายชื่อโฮสต์ และหมายเลขเครือข่ายเดิมก่อนทำการเพิ่ม

ผู้ใช้จะสามารถเลือกได้ว่า ต้องการเพิ่มเติมข้อมูลดังกล่าวไปยังบรรทัดที่เท่าใดของข้อมูลเดิม เนื่องจากกฎ (rule) ในการพิจารณาโฮสต์ หรือเครือข่าย บางอย่างจะต้องเรียงตามลำดับ เมื่อเลือกเลขบรรทัดได้แล้วจะปรากฏข้อความถามถึงชื่อโฮสต์ หรือหมายเลขเครือข่ายที่ต้องการทำการเพิ่มเติม และออพชั่นต่าง ๆ คือ การรับรอง (authentication) และล็อกไฟล์ ตัวอย่างเช่น ต้องการทำการอนุญาตเครือข่ายหมายเลข 161.246.10.X ไปที่บรรทัดที่ 1 (ซึ่งเดิมเป็นโฮสต์ไดมอนด์) โดยเครือข่ายดังกล่าวต้องทำการรับรองผู้ใช้ก่อนใช้งาน

Network	Permitted	Authentication	Log
3. *Firewall*	*No*	*No*	*No*
4. *161.246.4.**	*Yes*	*No*	*No*

Add to line number : 1

*Network : 161.246.10.**

Permitted (y/n) : y

Authentication (y/n) : y

รูป 9.4 การใส่ข้อมูลเพื่อเพิ่มเติมหมายเลขเครือข่าย

เมื่อทำการใส่ข้อมูลดังรูป 9.4 แล้ว จะมีการรายงานผลกลับมาดังต่อไปนี้

Network	Permitted	Authentication	Log
5. *161.246.1..**	*Yes*	*Yes*	*No*
6. *Firewall*	*No*	*No*	*No*
7. *161.246.4.**	*Yes*	*No*	*No*

Add another record (y/n) :

รูป 9.5 รายงานผลหลังจากทำการเพิ่มเติมหมายเลขเครือข่าย

ถ้าต้องการทำการเพิ่มเติมมากกว่า 1 เรคคอร์ด สามารถตอบ y ได้จากจุดนี้

9.3 การลบ (delete)

จากเมนูย่อย ดังรูป 9.2 เมื่อทำการเลือกหมายเลข 2 (delete) จะเป็นการลบหมายเลขเครือข่ายหรือชื่อโฮสต์ของบริการนั้น ๆ ออกจากไฟล์ /usr/local/etc/netperm-table ตัวอย่างเช่น ถ้าเราทำการเลือกบริการเอฟทีพีพร็อกซี (จากรูป 9.1 เลือกหมายเลข 2) จะปรากฏเมนูย่อย แล้วเลือกหมายเลข 2 จะปรากฏตารางแสดงรายชื่อของโฮสต์ และหมายเลขเครือข่ายที่มีอยู่เดิม ในไฟล์ /usr/local/etc/netperm-table ดังนี้

Network	Permitted	Authentication	Log
8. *161.246.4.**	*Yes*	*Yes*	*Yes*
9. *	*No*	*No*	*Yes*

Enter record number to delete or press e to exit to mainmenu :

รูป 9.6 ตารางแสดงรายชื่อโฮสต์ และหมายเลขเครือข่ายเดิมก่อนทำการลบ

ผู้ใช้จะต้องทำการเลือกเลขบรรทัดของเรคคอร์ดที่ต้องการลบ ตัวอย่างเช่น ต้องการทำการลบเครือข่ายหมายเลข 161.246.4.* ที่อยู่บรรทัดที่ 1 ออกจากไฟล์ /usr/local/etc/netperm-table หลังจากใส่เลขบรรทัดแล้วจะปรากฏผลดังนี้

Network	Permitted	Authentication	Log
1. *	*No*	*No*	*Yes*

Delete another record (y/n) :

รูป 9.7 การใส่ข้อมูลเพื่อเพิ่มเติมหมายเลขเครือข่าย

ถ้าต้องการทำการลบมากกว่า 1 เรคคอร์ด สามารถตอบ y ได้จากจุดนี้

9.4 การแก้ไข (edit)

จากเมนูย่อย ดังรูป 9.2 เมื่อทำการเลือกหมายเลข 3 (edit) จะเป็นการแก้ไขข้อมูลในเรคคอร์ดนั้น ซึ่งประกอบด้วย หมายเลขเครือข่าย หรือชื่อโฮสต์ การอนุญาต (permit) หรือปฏิเสธการใช้งาน (deny) บริการที่ได้ทำการเลือกไว้จากเมนูหลัก การรับรองผู้ใช้ (authentication) ในบริการเทลเน็ตพร็อกซี , เอฟทีพีพร็อกซี , อาร์ล็อกอินพร็อกซี และเอชทีทีพีพร็อกซี และการทำล็อกไฟล์ ในกรณีของเอฟทีพีพร็อกซี ตัวอย่างเช่น ถ้าเราทำการเลือกบริการอาร์ล็อกอินพร็อกซี (รูป 9.1 หมายเลข 3) จะปรากฏเมนูย่อย แล้วเลือกหมายเลข 3 จะปรากฏตารางแสดงรายชื่อของโฮสต์ และหมายเลขเครือข่ายที่มีอยู่เดิมในไฟล์ /usr/local/etc/netperm-table ดังนี้

Network	Permitted	Authentication	Log
*10. 161.246.4.**	*Yes*	*Yes*	*No*
11. *161.246.5.**	*Yes*	*Yes*	*No*
12. *Diamond*	*Yes*	*Yes*	*No*
13. *Jade*	*Yes*	*Yes*	*No*
14. *Opal*	*Yes*	*Yes*	*No*

Enter record whice you want to edit e to exit :

รูป 9.8 ตารางแสดงรายชื่อโฮสต์ และหมายเลขเครือข่ายเดิมก่อนทำการแก้ไข

หลังจากที่มีการแสดงรายชื่อของโฮสต์และหมายเลขเครือข่ายแล้ว (รูป 9.8) เราจะทำการเลือกหมายเลขเรคคอร์ดที่ต้องการทำการแก้ไข ตัวอย่างเช่น ต้องการแก้ไขเรคคอร์ดที่ 3 ซึ่งมีข้อมูลชื่อโฮสต์ไดมอนด์ มีการอนุญาตให้ใช้งานอาร์ล็อกอิน โดยต้องทำการรับรองผู้ใช้ และไม่ยอมให้ผู้ใช้เปลี่ยนรหัสผ่านที่ออเธนทิเคชั่นเซิร์ฟเวอร์ เราจะทำการเปลี่ยนฟิลด์หลัง โดยจะยอมให้ผู้ใช้เปลี่ยนรหัสผ่านที่ออเธนทิเคชั่นได้ ขั้นตอนต่อไปเราจะใส่หมายเลข 3 และข้อมูลต่าง ๆ ดังนี้

Network	Permitted	Authentication	Log
*15. 161.246.4.**	*Yes*	*Yes*	*No*
16. *161.246.5.**	*Yes*	*Yes*	*No*
17. *Diamond*	*Yes*	*Yes*	*No*
18. *Jade*	*Yes*	*Yes*	*No*
19. *Opal*	*Yes*	*Yes*	*No*

Enter record whice you want to edit e to exit : 3

Enter new value network number [diamond]:

Permit [y]: (y/n):

Authentication [y]: (y/n):

Can change password [n]: (y/n): y

รูป 9.9 การทำการแก้ไขข้อมูล

เมื่อทำการใส่ข้อมูลดังรูป 9.9 แล้ว จะมีการรายงานผลกลับมาดังต่อไปนี้

Network	Permitted	Authentication	Log
20. *161.246.4.**	*Yes*	*Yes*	*No*
21. *161.246.5.**	*Yes*	*Yes*	*No*
22. *Diamond*	*Yes*	*Yes*	*Yes*
23. *Jade*	*Yes*	*Yes*	*No*
24. *Opal*	*Yes*	*Yes*	*No*

Edit another record (y/n) :

รูป 9.10 รายงานผลหลังจากทำการแก้ไข

ถ้าต้องการทำการแก้ไขมากกว่า 1 เรคคอร์ด สามารถตอบ y ได้จากจุดนี้

บทที่ 10
สรุป

ระบบไฟร์วอลล์ถือเป็นระบบรักษาความปลอดภัยแก่ระบบเครือข่ายระบบหนึ่ง ซึ่งปัจจุบันมีระบบไฟร์วอลล์อยู่มากมาย แต่ละระบบมีความเหมาะสมกับเครือข่ายแตกต่างกันไป ขึ้นอยู่กับสถาปัตยกรรมของเครือข่ายนั้น ๆ แต่จุดประสงค์หลักเดียวกันคือ ต้องการรักษาความปลอดภัยของข้อมูลภายในระบบเครือข่าย แต่ในขณะเดียวกันภายในระบบเครือข่ายยังคงต้องสามารถติดต่อสื่อสารกับระบบภายนอกได้เช่นเดิมด้วย นอกจากนั้นการติดตั้งระบบไฟร์วอลล์ยังอาจทำให้การใช้งานของผู้ใช้ยุ่งยากมากขึ้น และประสิทธิภาพในการเคลื่อนย้ายข้อมูลลดลง

10.1 แนวทางการนำไปใช้งาน

ระบบไฟร์วอลล์ที่ได้นำเสนอในรายงานฉบับนี้ เป็นระบบไฟร์วอลล์ที่มีความสมบูรณ์และสามารถนำไปประยุกต์ใช้กับระบบเครือข่ายใด ๆ ที่ต้องการความปลอดภัยได้ การใช้งานมีความยุ่งยากเกิดขึ้นเล็กน้อย แต่เพื่อให้ได้มาซึ่งความปลอดภัยของข้อมูลก็จำเป็นจะต้องยอมสูญเสียความสะดวกบางประการ ระบบนี้เหมาะสมกับการนำไปประยุกต์ใช้กับระบบเครือข่ายที่มีขนาดใหญ่ ๆ แต่ถ้าเป็นเครือข่ายขนาดเล็กที่มีผู้ใช้ไม่มากนัก และไม่ต้องการการรักษาความปลอดภัยที่เข้มงวดมาก สามารถเลือกใช้งานระบบไฟร์วอลล์ที่เป็นสถาปัตยกรรมแบบอื่นได้ ดังที่ได้นำเสนอไปในรายงาน

10.2 ปัญหาและวิธีแก้ไข

- ความไม่สะดวกในการใช้งานของผู้ใช้
  - เลือกเปลี่ยนแปลงสถาปัตยกรรมของไฟร์วอลล์ให้เหมาะสมกับระบบเครือข่ายที่มีอยู่ โดยสถาปัตยกรรมบางแบบจะมีความสะดวกในการใช้งานมากกว่า แต่ระดับการรักษาความปลอดภัยจะลดลง เช่น เปลี่ยนจากระบบพร็อกซีเป็นระบบแพ็กเก็ตฟิลเตอร์ริ่ง
  - ทำการประชาสัมพันธ์ให้ผู้ใช้เข้าใจถึงการใช้งานไฟร์วอลล์
- การลดลงของอัตราการถ่ายเทข้อมูล
  - ปรับเปลี่ยนฮาร์ดแวร์ให้มีประสิทธิภาพมากขึ้น รวมทั้งโครงสร้างของระบบเครือข่าย
  - เลือกใช้ซอฟท์แวร์ที่มีการพัฒนามาอย่างดี จะช่วยเพิ่มประสิทธิภาพของการถ่ายเทข้อมูล
  - เลือกเปลี่ยนแปลงสถาปัตยกรรมของไฟร์วอลล์ให้เหมาะสมกับระบบเครือข่ายที่มีอยู่ เช่น เปลี่ยนจากสถาปัตยกรรมแบบดูอัลโฮมโฮสต์เป็นสถาปัตยกรรมแบบสกรีนสับเน็ต
- ปัญหาในด้านความปลอดภัยของซอฟท์แวร์

- เลือกใช้ซอฟท์แวร์ที่เป็นคอมเมอร์เชียลแวร์ เนื่องจากซอฟท์แวร์ที่เป็นแชร์แวร์มักจะมีข้อผิดพลาดอยู่มาก และข้อผิดพลาดดังกล่าวมักจะเป็นช่องทางให้ผู้บุกรุกเข้ามาทำลายข้อมูลของเรา

10.3 แนวทางการศึกษาต่อ

เนื่องจากซอฟท์แวร์ที่ได้มาเป็นโค้ดที่ยังไม่ได้ทำการคอมไพล์ เราสามารถนำโค้ดดังกล่าวมาทำการพัฒนา เพื่อให้เข้ากับระบบเครือข่ายของเรามากกว่าที่เป็นอยู่ เราสามารถเพิ่มเติมส่วนที่ขาดหายไป หรือสามารถตัดส่วนที่ไม่จำเป็นของซอฟท์แวร์ออก โดยไม่มีปัญหาทางด้านลิขสิทธิ์

เราสามารถออกแบบสถาปัตยกรรมที่แตกต่างจากสถาปัตยกรรมตามทฤษฎีที่กล่าวมาได้ โดยทำการประยุกต์จากทฤษฎีเดิมให้เหมาะสมกับเครือข่ายที่มีอยู่ หรืออาจนำทฤษฎีที่มีอยู่มาพัฒนาระบบไฟร์วอลล์ให้ทำงานบนโปรโตคอลอื่น ๆ นอกเหนือจากโปรโตคอลทีซีพีไอพี เช่น ไอพีเอ็กซ์ เป็นต้น รวมทั้งอาจทำการพัฒนาระบบไฟร์วอลล์บนตัวฮาร์ดแวร์โดยตรงได้

ภาคผนวก

โปรแกรมแสดงเมนูหลัก

```
#!/bin/sh
until false
do
printf "\n\n\n"
printf " ------------------------------------------------------------\n"
printf "                        Main Menu                           \n"
printf " ------------------------------------------------------------\n"
printf "          1. Telnet-gw\n"
printf "          2. FTP-gw\n"
printf "          3. Rlogin-gw\n"
printf "          4. HTTP-gw\n"
printf "          5. Telnet\n"
printf "          6. FTP\n"
printf "          7. EXIT\n"
printf "\n"
printf "          Choice (1-7) : \c"
read ch
        if [ "$ch" = "7" ]
        then
                exit
        fi
submenu $ch
#clear
done
```

โปรแกรมแสดงเมนูย่อย

```
#!/bin/sh
printf " ----------------------------------------\n"
case $1 in
1) ch=tn-gw
printf "          Telnet-gw\n";;
2) ch=ftp-gw
printf "          FTP-gw\n";;
3) ch=rlogin-gw
printf "          Rlogin-gw\n";;
4) ch=http-gw
printf "          HTTP-gw\n";;
5) ch=netacl-in.telnetd
printf "          Telnet\n";;
6) ch=netacl-wu.ftpd
printf "          FTP\n";;
esac
printf " ----------------------------------------\n\n"
printf "  1. Add\n"
printf "  2. Delete\n"
printf "  3. Edit\n"
printf "  4. Return to Main Menu\n\n"
printf "  Choice (1-4) : \c"
read choice
case $choice in
1)
        addmenu $ch;;
2)
        delmenu $ch;;
3)
        editmenu $ch;;
4)
        exit;;
esac
```

โปรแกรมทำการเพิ่มกฎบนบริการเอฟทีพี (FTP)

```
#!/bin/sh
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
read perm
        if [ "$perm" = "y" ];
        then perm="permit-hosts"
        else perm="hosts"
        fi
declare -i count=1
declare -i j=$line
while read bg permit network execute
do
        if test $count -eq $j
        then
                if [ "$perm" = "permit-hosts" ];
                then
                        echo "$bg $permit $network -exec /usr/bin/wu.ftpd" >> /tmp/tempfile.$$
                        echo "$bg $perm $netw -exec /usr/bin/wu.ftpd" >> /tmp/tempfile.$$
                else
                        echo "$bg $permit $network -exec /bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
                        echo "$bg $perm $netw -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
                fi
        else
                if [ "$perm" = "permit-hosts" ];
                then
                        echo "$bg $permit $network -exec /usr/bin/wu.ftpd" >> /tmp/tempfile.$$
                else
```

```
                    echo "$bg $permit $network -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$

                fi
        fi
        count=count+1
done < /tmp/tempfile
if test $j -eq $count
then
        if [ "$perm" = "permit-hosts" ];
        then
                echo "netacl-wu.ftpd: $perm $netw -exec /usr/bin/wu.ftpd" >> /tmp/tempfile.$$
        else
                echo "netacl-wu.ftpd: $perm $netw -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
        fi
fi
mv /tmp/tempfile.$$ /tmp/tempfile
grep -v "netacl-wu.ftpd" /usr/local/etc/netperm-table | grep -v "FTP rules" >> /tmp/tempfile.2
echo "#               FTP rules" >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
rm /tmp/tempfile
mv /tmp/tempfile.2 /
```

โปรแกรมทำการเพิ่มกฎบนบริการเอชทีทีพี (HTTP)

```
#!/bin/sh
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
read perm
        if [ "$perm" = "y" ];
```

```
        then perm="hosts"
        else perm="deny-hosts"
        fi
declare -i count=1
declare -i l=$line
while read bg permit network
do
        if test $count -eq $l
        then
                echo "$bg $perm $netw" >> /tmp/tempfile.$$
                echo "$bg $permit $network" >> /tmp/tempfile.$$
        else
                echo "$bg $permit $network" >> /tmp/tempfile.$$
        fi
        count=count+1
done < /tmp/tempfile
if test $l -gt $count
then
        echo "http-gw: $perm $netw" >> /tmp/tempfile.$$
fi
mv /tmp/tempfile.$$ /tmp/tempfile
grep -v "http-gw" /usr/local/etc/netperm-table | grep -v "HTTP-Gateway" >> /tmp/tempfile.2
echo "#               HTTP-Gateway rules" >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
rm /tmp/tempfile
mv /tmp/tempfile.2 /tmp/test
mainmenu
```

โปรแกรมทำการเพิ่มเมนู

```
#!/bin/sh
cat /usr/local/etc/netperm-table | grep $1 | grep -v "#" > /tmp/tempfile
case $1 in
tn-gw)
```

```
        listtngw
        addtngw;;
ftp-gw)
        listftpgw
        addftpgw;;
rlogin-gw)
        listrlogin
        addrlogin;;
http-gw)
        listhttp
        addhttp;;
netacl-in.telnetd)
        listtelnet
        addtelnet;;
netacl-wu.ftpd)
        listftp
        addftp;;
esac
```

โปรแกรมทำการเพิ่มกฎบนบริการเทลเน็ตเกตเวย์ (TELNET gateway)

```
#!/bin/sh
ok=y
until [ "$ok" = "n" ]
do
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
read perm
        if [ "$perm" = "y" ];
        then perm="permit-hosts"
        else perm="deny-hosts"
        fi
printf "Authentication (y/n) : \c"
```

```
read authentication
        if [ "$authentication" = "y" ];
        then authentication="-auth"
        else authentication=""
        fi
declare -i count=1
declare -i j=$line
while read bg permit network auth
do
        if test $count -eq $j
        then
                echo "$bg $perm $netw $authentication" >> /tmp/tempfile.$$
                echo "$bg $permit $network $auth" >> /tmp/tempfile.$$
        else
                echo "$bg $permit $network $auth" >> /tmp/tempfile.$$
        fi
        count=count+1
done < /tmp/tempfile
if  test $j -eq $count
then
        echo "tn-gw: $perm $netw $authentication" >> /tmp/tempfile.$$
fi
mv /tmp/tempfile.$$ /tmp/tempfile
clear
listtngw
printf "Add another record (y/n) : \c"
read ok
done
grep -v "tn-gw" /usr/local/etc/netperm-table | grep -v "TELNET-Gateway" >> /tmp/tempfile.2
echo "#               TELNET-Gateway rules" >> /tmp/tempfile.2
echo "tn-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.2
echo "tn-gw: welcome-msg /usr/local/etc/welcome.txt" >> /tmp/tempfile.2
echo "tn-gw: timeout 3600 " >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
rm /tmp/tempfile
mv /tmp/tempfile.2 /usr/local/etc/netperm-table
```

โปรแกรมทำการเพิ่มกฎบนบริการเอฟทีพีเกตเวย์ (FTP gateway)

```
#!/bin/sh
ok=y
until [ "$ok" = "n" ]
do
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
read perm
        if [ "$perm" = "y" ];
        then perm="permit-hosts"
        else perm="deny-hosts"
        fi
printf "Authentication (y/n) : \c"
read authentication
        if [ "$authentication" = "y" ];
        then authentication="-authall"
        else authentication=""
        fi
printf "Log (y/n) : \c"
read logf
        if [ "$logf" = "y" ];
        then logf="-log { retr stor }"
        else logf=""
        fi
declare -i count=1
declare -i j=$line
while read bg permit network auth log
do
        if test $count -eq $j
        then
```

```
                echo "$bg $perm $netw $authentication $logf" >> /tmp/tempfile.$$
                echo "$bg $permit $network $auth $log" >> /tmp/tempfile.$$
        else
                echo "$bg $permit $network $auth $log" >> /tmp/tempfile.$$
        fi
        count=count+1
done < /tmp/tempfile
if test $j -eq $count
then
                echo "ftp-gw: $perm $netw $authentication $logf" >> /tmp/tempfile.$$
fi
mv /tmp/tempfile.$$ /tmp/tempfile
clear
listtngw
printf "Add another record (y/n) : \c"
read ok
done
grep -v "ftp-gw" /usr/local/etc/netperm-table | grep -v "FTP-Gateway" >> /tmp/tempfile.2
echo "#              FTP-Gateway rules" >> /tmp/tempfile.2
echo "ftp-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.2
echo "ftp-gw: welcome-msg /usr/local/etc/welcome.txt" >> /tmp/tempfile.2
echo "ftp-gw: timeout 3600 " >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
rm /tmp/tempfile
mv /tmp/tempfile.2 /usr/local/etc/netperm-table
```

โปรแกรมทำการเพิ่มกฎบนบริการอาร์ล็อกอิน (RLOGIN)

```
#!/bin/sh
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
```

```
read perm
        if [ "$perm" = "y" ];
        then perm="permit-hosts"
        else perm="deny-hosts"
        fi
printf "Authentication (y/n) : \c"
read authentication
        if [ "$authentication" = "y" ];
        then
                authentication="-auth"
                printf "Change Authentication Password (y/n) : \c"
                read passok
                if [ "$passok" = "y" ];
                        then passok="-passok"
                        else passok=""
                fi
        else authentication=""
        fi
declare -i count=1
declare -i j=$line
while read bg permit network auth pass
do
        if test $count -eq $j
        then
                echo "$bg $perm $netw $authentication $passok" >> /tmp/tempfile.$$
                echo "$bg $permit $network $auth $pass" >> /tmp/tempfile.$$
        else
                echo "$bg $permit $network $auth $pass" >> /tmp/tempfile.$$
        fi
        count=count+1
done < /tmp/tempfile
if test $j -gt $count
then
        echo "rlogin-gw: $perm $netw $authentication $passok" >> /tmp/tempfile.$$
```

```
fi
mv /tmp/tempfile.$$ /tmp/tempfile
grep -v "rlogin-gw" /usr/local/etc/netperm-table | grep -v "RLOGIN-Gateway" >> /tmp/tempfile.2
echo "#              RLOGIN-Gateway rules" >> /tmp/tempfile.2
echo "rlogin-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.2
echo "rlogin-gw: welcome-msg /usr/local/etc/welcome.txt" >> /tmp/tempfile.2
echo "rlogin-gw: timeout 3600 " >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
rm /tmp/tempfile
mv /tmp/tempfile.2 /usr/local/etc/netperm-table
mainmenu
```

โปรแกรมทำการเพิ่มกฎบนบริการเทลเน็ต (TELNET)

```
#!/bin/sh
printf "Add to line number : \c"
read line
printf "Network : \c"
read netw
printf "Permitted (y/n) : \c"
read perm
        if [ "$perm" = "y" ];
        then perm="permit-hosts"
        else perm="hosts"
        fi
declare -i count=1
declare -i l=$line
while read bg permit network execute
do
        if test $count -eq $l
        then
                if [ "$perm" = "permit-hosts" ];
                then
                        echo "$bg $perm $netw -exec /usr/bin/in.telnetd " >> /tmp/tempfile.$$
```

```
                    echo "$bg $permit $network -exec /usr/bin/in.telnetd" >> /tmp/tempfile.$$
                else
                    echo "$bg $perm $netw -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
                    echo "$bg $permit $network -exec /bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
                fi
        else
                if [ "$perm" = "permit-hosts" ];
                then
                    echo "$bg $permit $network -exec /usr/bin/in.telnetd" >> /tmp/tempfile.$$
                else
                    echo "$bg $permit $network -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
                fi
        fi
        count=count+1
done < /tmp/tempfile

if test $count -eq $l
then
        if [ "$perm" = "permit-hosts" ];
        then
                echo "netacl-in.telnetd: $perm $netw -exec /usr/bin/in.telnetd" >> /tmp/tempfile.$$
        else
                echo "netacl-in.telnetd: $perm $netw -exec /usr/bin/cat /usr/local/etc/deny.txt" >>
/tmp/tempfile.$$
        fi
fi

mv /tmp/tempfile.$$ /tmp/tempfile
grep -v "netacl-in.telnetd" /usr/local/etc/netperm-table | grep -v "TELNET rules" >> /tmp/tempfile.2
echo "#               TELNET rules" >> /tmp/tempfile.2
cat /tmp/tempfile >> /tmp/tempfile.2
```

```
rm /tmp/tempfile
mv /tmp/tempfile.2 /tmp/test
mainmenu
```

โปรแกรมทำการลบกฎบนบริการเอฟทีพี (FTP)

```
#!/bin/sh
ok="y"
until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice
declare -i counter=1
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
mv /tmp/tempfile.$$ /tmp/tempfile
listftp
printf "Delete another record (y/n) : \c"
read ok
done
#---------- Update netperm-table ---------------------------
grep -v "netacl-wu.ftpd" /usr/local/etc/netperm-table | grep -v "FTP rules"  > /tmp/tempfile.$$
echo "#       FTP rules" >> /tmp/tempfile.$$
```

```
echo "netacl-wu.ftpd: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.$$
echo "netacl-wu.ftpd: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile.$$
echo "netacl-wu.ftpd: timeout 3600" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการลบกฎบนบริการเอฟทีพีเกตเวย์ (FTP gateway)

```
#!/bin/sh
ok="y"
until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice

declare -i counter=1
echo "" >> /tmp/tempfile.$$
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
mv /tmp/tempfile.$$ /tmp/tempfile
listtngw
printf "Delete another record (y/n) : \c"
```

```
read ok
done
#---------- Update netperm-table ----------
grep -v "ftp-gw" /usr/local/etc/netperm-table | grep -v "FTP-Gateway"  > /tmp/tempfile.$$
echo "#       FTP-Gateway rules" >> /tmp/tempfile.$$
echo "ftp-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.$$
echo "ftp-gw: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile.$$
echo "ftp-gw: timeout 3600" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการลบกฎบนบริการเอชทีทีพี (HTTP)

```
#!/bin/sh
ok="y"
until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice
declare -i counter=1
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
```

```
mv /tmp/tempfile.$$ /tmp/tempfile
listhttp
printf "Delete another record (y/n) : \c"
read ok
done
#---------- Update netperm-table ----------
grep -v "http-gw" /usr/local/etc/netperm-table | grep -v "HTTP-Gateway"  > /tmp/tempfile.$$
echo "#       HTTP-Gateway rules" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการลบเมนู

```
#!/bin/sh
cat /usr/local/etc/netperm-table | grep $1 | grep -v "txt" | grep -v "#" | grep -v "timeout" > /tmp/tempfile
case $1 in
tn-gw)
        listtngw
        deltngw;;
ftp-gw)
        listftpgw
        delftpgw;;
rlogin-gw)
        listrlogin
        delrlogin;;
http-gw)
        listhttp
        delhttp;;
netacl-in.telnetd)
        listtelnet
        deltelnet;;
netacl-wu.ftpd)
        listftp
        delftp;;
```

```
esac
```

โปรแกรมทำการลบกฎบนบริการเทลเน็ตเกตเวย์ (TELNET gateway)

```
#!/bin/sh
ok="y"

until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice
declare -i counter=1
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
mv /tmp/tempfile.$$ /tmp/tempfile
listtngw
printf "Delete another record (y/n) : \c"
read ok
done
#---------- Update netperm-table ---------------------------
grep -v "tn-gw" /usr/local/etc/netperm-table | grep -v "TELNET-Gateway"  > /tmp/tempfile.$$
echo "#       TELNET-Gateway rules" >> /tmp/tempfile.$$
echo "tn-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.$$
echo "tn-gw: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile.$$
```

```
echo "tn-gw: timeout 3600" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการลบกฎบนบริการอาร์ล็อกอิน (RLOGIN)

```
#!/bin/sh
ok="y"
until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice
declare -i counter=1
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
mv /tmp/tempfile.$$ /tmp/tempfile
listrlogin
printf "Delete another record (y/n) : \c"
read ok
done
#---------- Update netperm-table ----------
grep -v "rlogin-gw" /usr/local/etc/netperm-table | grep -v "RLOGIN-Gateway"  > /tmp/tempfile.$$
```

```
echo "#       RLOGIN-Gateway rules" >> /tmp/tempfile.$$
echo "rlogin-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.$$
echo "rlogin-gw: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile.$$
echo "rlogin-gw: timeout 3600" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการลบกฎบนบริการเทลเน็ต (TELNET)

```
#!/bin/sh
ok="y"
until [ "$ok" = "n" ]
do
printf "Enter record number to delete or press e to exit to mainmenu : \c"
read choice
        if [ "$choice" = "e" ]
        then
                exit
        fi
declare -i j=$choice
declare -i counter=1
while read bg
do
        if test $j -ne $counter
        then
                echo "$bg" >> /tmp/tempfile.$$
        fi
        counter=counter+1
done < /tmp/tempfile
mv /tmp/tempfile.$$ /tmp/tempfile
listtelnet
printf "Delete another record (y/n) : \c"
read ok
```

```
done
#---------- Update netperm-table ----------
grep -v "netacl-in.telnetd" /usr/local/etc/netperm-table | grep -v "TELNET rules"  > /tmp/tempfile.$$
echo "#       TELNET rules" >> /tmp/tempfile.$$
echo "netacl-in.telnetd: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile.$$
echo "netacl-in.telnetd: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile.$$
echo "netacl-in.telnetd: timeout 3600" >> /tmp/tempfile.$$
cat /tmp/tempfile >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /usr/local/etc/netperm-table
rm /tmp/tempfile
```

โปรแกรมทำการแก้ไขกฎบนบริการเอฟทีพี (FTP)

```
#!/bin/sh
printf "Enter record which you want to edit e to exit : \c"
read record
declare -i j=$record
declare -i count=1
while read bg permit network execute thisfile
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
                chthisfile=$thisfile
        fi
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]. \c"
read network
if [ "$network" != "" ]
then
        chnetwork=$network
fi
```

```
if [ "$chpermit" = "permit-hosts" ]
then
        printf "Permit [y]: (y/n): \c"
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="permit-hosts"
        else
                chpermit="deny-hosts"
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
                chpermit="permit-hosts"
        fi
fi
if [ "$chpermit" = "permit-hosts" ]
then
        thisfile="$chthisfile"
else
        printf "File to execute [${chthisfile}]: \c"
        read thisfile
fi
declare -i count=1
while read bg permit network execute thisfil
do
        if test $j -eq $count
        then
                echo "$bg $chpermit $chnetwork $execute $thisfile"
        else
```

```
                echo "$bg $permit $network $execute $thisfil"
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "netacl-wu.ftpd" /usr/local/etc/netperm-table | grep -v "FTP rules" > /tmp/tempfile
echo "#           FTP rules" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแก้ไขกฎบนบริการเอชทีทีพี (HTTP)

```
#!/bin/sh
printf "Enter record which you want to edit or e to exit : \c"
read record
declare -i j=$record
declare -i count=1
while read bg permit network
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
        fi
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]: \c"
read network
if [ "$network" != "" ]
then
        chnetwork=$network
fi
if [ "$chpermit" = "hosts" ]
then
        printf "Permit [y]: (y/n): \c"
```

```
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="hosts"
        else
                chpermit="deny-hosts"
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
                chpermit="hosts"
        fi
fi
declare -i count=1
while read bg permit network authentication
do
        if test $j -eq $count
        then
                echo "$bg $chpermit $chnetwork $chauth"
        else
                echo "$bg $permit $network $authentication"
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "http-gw" /usr/local/etc/netperm-table | grep -v "HTTP-Gateway" > /tmp/tempfile
echo "#       HTTP-Gateway rules" >> /tmp/tempfile
echo "http-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile
echo "http-gw: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile
echo "http-gw: timeout 3600" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
```

```
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแก้ไขเมนู

```
#!/bin/sh
cat /usr/local/etc/netperm-table | grep $1 | grep -v "txt" | grep -v "time" > /tmp/tempfile
case $1 in
tn-gw)
        listtngw
        edittngw;;
ftp-gw)
        listftpgw
        editftpgw;;
rlogin-gw)
        listrlogin
        editrlogin;;
http-gw)
        listhttp
        edithttp;;
netacl-in.telnetd)
        listtelnet
        edittelnet;;
netacl-wu.ftpd)
        listftp
        editftp;;
esac
```

โปรแกรมทำการแก้ไขกฎบนบริการเทลเน็ตเกตเวย์ (TELNET gateway)

```
#!/bin/sh
printf "Enter record which you want to edit or e to exit : \c"
read record
declare -i j=$record
```

```
declare -i count=1
while read bg permit network authentication
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
                chauth=$authentication
        fi
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]: \c"
read network
if [ "$network" != "" ]
then
        chnetwork=$network
fi
if [ "$chpermit" = "permit-hosts" ]
then
        printf "Permit [y]: (y/n): \c"
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="permit-hosts"
        else
                chpermit="deny-hosts"
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
```

```
                chpermit="permit-hosts"
        fi
fi
if [ "$chpermit" = "permit-hosts" ];
then
        if [ "$chauth" = "-auth" ]
        then
                printf "Authentication [y]: (y/n):\c"
                read authentication
                if [ "$authentication" = "y" ]||[ "$authentication" = "" ]
                then
                        chauth="-auth"
                else
                        chauth=""
                fi
        else
                printf "Authentication [n]: (y/n):\c"
                read authentication
                if [ "$authentication" = "n" ]||[ "$authentication" = "" ]
                then
                        chauth=""
                else
                        chauth="-auth"
                fi
        fi
fi
declare -i count=1
while read bg permit network authentication
do
        if test $j -eq $count
        then
                echo "$bg $chpermit $chnetwork $chauth"
        else
                echo "$bg $permit $network $authentication"
```

```
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "tn-gw" /usr/local/etc/netperm-table | grep -v "TELNET-Gateway" > /tmp/tempfile
echo "#       TELNET-Gateway rules" >> /tmp/tempfile
echo "tn-gw: denial-msg /usr/local/etc/deny.txt" >> /tmp/tempfile
echo "tn-gw: welcome-msg /usr/local/etc/welcome.txt ">> /tmp/tempfile
echo "tn-gw: timeout 3600" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแก้ไขกฎบนบริการเอฟทีพีเกตเวย์ (FTP gateway)

```
#!/bin/sh
printf "Enter record whice you want to edit e to exit : \c"
read record
declare -i j=$record
declare -i count=1
while read bg permit network authentication log
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
                if [ "$authentication" = "-log" ]
                  then
                        chauth=""
                        chlog="-log { retr stor }"
                  else
                        chauth=$authentication
                        chlog=$log
                fi
        fi
```

```
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]: \c"
read network
if [ "$network" != "" ]
then
        chnetwork=$network
fi
if [ "$chpermit" = "permit-hosts" ]
then
        printf "Permit [y]: (y/n): \c"
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="permit-hosts"
        else
                chpermit="deny-hosts"
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
                chpermit="permit-hosts"
        fi
fi
if [ "$chpermit" = "permit-hosts" ]
then
        if [ "$chauth" = "-authall" ]
        then
                printf "Authentication [y]: (y/n):\c"
                read authentication
```

```
                if [ "$authentication" = "y" ]||[ "$authentication" = "" ]
                then
                        chauth="-authall"
                else
                        chauth=""
                fi
        else
                printf "Authentication [n]: (y/n):\c"
                read authentication
                if [ "$authentication" = "n" ]||[ "$authentication" = "" ]
                then
                        chauth=""
                else
                        chauth="-authall"
                fi
        fi
else
        chauth=""
fi
if  [ "$chlog" = "-log { retr stor }" ]
then
        printf "Log [y]: (y/n): \c"
        read log
        if [ "$log" = "y" ]||[ "$log" = ""]
        then
                chlog="-log { retr stor }"
        fi
        if [ "$log" = "n" ]
        then
                chlog=""
        fi
else
        printf "Log [n]: (y/n): \c"
        read log
```

```
        if [ "$log" = "y" ]
        then
                chlog="-log { retr stor }"
        fi
        if [ "$log" = "" ]||[ "$log" = "n" ]
        then
                chlog=""
        fi
fi
declare -i count=1
while read bg permit network authentication log
do
        if test $j -eq $count
        then
                echo "$bg $chpermit $chnetwork $chauth $chlog"
        else
                echo "$bg $permit $network $authentication"
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "ftp-gw" /usr/local/etc/netperm-table | grep -v "FTP-Gateway" > /tmp/tempfile
echo "#       FTP-Gateway rules" >> /tmp/tempfile
echo "ftp-gw: deny-msg /usr/local/etc/deny.txt" >> /tmp/tempfile
echo "ftp-gw: welcome-msg /usr/local/etc/welcome.txt" >> /tmp/tempfile
echo "ftp-gw: timeout 3600" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแก้ไขกฎบนบริการอาร์ล็อกอิน (RLOGIN)

```
#!/bin/sh
printf "Enter record whice you want to edit e to exit : \c"
read record
```

```
declare -i j=$record
declare -i count=1
while read bg permit network authentication log
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
                if [ "$authentication" = "-passok" ]
                  then
                        chauth=""
                        chlog="-passok"
                  else
                        chauth=$authentication
                        chlog=$log
                fi
        fi
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]: \c"
read network
if [ "$network" != "" ]
then
        chnetwork=$network
fi
if [ "$chpermit" = "permit-hosts" ]
then
        printf "Permit [y]: (y/n): \c"
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="permit-hosts"
        else
                chpermit="deny-hosts"
```

```
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
                chpermit="permit-hosts"
        fi
fi
if [ "$chpermit" = "permit-hosts" ]
then
        if [ "$chauth" = "-auth" ]
        then
                printf "Authentication [y]: (y/n):\c"
                read authentication
                if [ "$authentication" = "y" ]||[ "$authentication" = "" ]
                then
                        chauth="-auth"
                else
                        chauth=""
                fi
        else
                printf "Authentication [n]: (y/n):\c"
                read authentication
                if [ "$authentication" = "n" ]||[ "$authentication" = "" ]
                then
                        chauth=""
                else
                        chauth="-auth"
                fi
        fi
        if  [ "$chlog" = "-passok" ]
```

```
        then
                printf "Can change password [y]: (y/n): \c"
                read log
                if [ "$log" = "y" ]||[ "$log" = "" ]
                then
                        chlog="-passok"
                fi
                if [ "$log" = "n" ]
                then
                        chlog=""
                fi
        else
                printf "Can change password [n]: (y/n): \c"
                read log
                if [ "$log" = "y" ]
                then
                        chlog="-passok"
                fi
                if [ "$log" = "" ]||[ "$log" = "n" ]
                then
                        chlog=""
                fi
        fi
else
        chauth=""
        chlog=""
fi
declare -i count=1
while read bg permit network authentication log
do
        if test $j -eq $count
        then
                echo "$bg $chpermit $chnetwork $chauth $chlog"
        else
```

```
                echo "$bg $permit $network $authentication"
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "rlogin-gw" /usr/local/etc/netperm-table | grep -v "RLOGIN-Gateway" > /tmp/tempfile
echo "#           Rlogin-Gateway" >> /tmp/tempfile
echo "rlogin-gw: deny-msg /usr/local/etc/deny.txt" >> /tmp/tempfile
echo "rlogin-gw: welcome-msg /usr/local/etc/welcome.txt" >> /tmp/tempfile
echo "rlogin-gw: timeout 3600" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแก้ไขกฎบนบริการเทลเน็ต (TELNET)

```
#!/bin/sh
printf "Enter record which you want to edit e to exit : \c"
read record
declare -i j=$record
declare -i count=1
while read bg permit network execute thisfile
do
        if test $j -eq $count
        then
                chpermit=$permit
                chnetwork=$network
                chthisfile=$thisfile
        fi
        count=count+1
done < /tmp/tempfile
printf "Enter new value network number [${chnetwork}]: \c"
read network
if [ "$network" != "" ]
then
```

```
        chnetwork=$network
fi
if [ "$chpermit" = "permit-hosts" ]
then
        printf "Permit [y]: (y/n): \c"
        read permit
        if [ "$permit" = "y" ]||[ "$permit" = "" ]
        then
                chpermit="permit-hosts"
        else
                chpermit="deny-hosts"
        fi
else
        printf "Permit [n]: (y/n): \c"
        read permit
        if [ "$permit" = "n" ]||[ "$permit" = "" ]
        then
                chpermit="deny-hosts"
        else
                chpermit="permit-hosts"
        fi
fi
if [ "$chpermit" = "permit-hosts" ]
then
        thisfile="$chthisfile"
else
        printf "File to execute [${chthisfile}]: \c"
        read thisfile
fi
declare -i count=1
while read bg permit network execute thisfil
do
        if test $j -eq $count
        then
```

```
                echo "$bg $chpermit $chnetwork $execute $thisfile"
        else
                echo "$bg $permit $network $execute $thisfil"
        fi
        count=count+1
done < /tmp/tempfile > /tmp/tempfile.$$
grep -v "netacl-in.telnetd" /usr/local/etc/netperm-table | grep -v "TELNET rules" >> /tmp/tempfile
echo "#       TELNET rules" >> /tmp/tempfile
cat /tmp/tempfile.$$ >> /tmp/tempfile
mv /tmp/tempfile /usr/local/etc/netperm-table
rm /tmp/tempfile.$$
```

โปรแกรมทำการแสดงรายการบนบริการเอฟทีพีเกตเวย์ (FTP gateway)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "timeout" | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "   ------------------------------------------------------------------\n"
printf "   |        Network        | Permitted | Authentication | Log |\n"
printf "   ------------------------------------------------------------------\n"
declare -i num=0
while read bg permit network auth log
do
        num=num+1
        printf " $num."
        if [ $num -lt 10 ];
                then printf " "
        fi
        printf "| $network"
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
        do
                cnt=cnt-1
```

```
                printf " "
        done
        if [ "$permit" = "permit-hosts" ];
                then printf "|    yes    "
                else printf "|    no     "
        fi
        if [ "$auth" = "-authall" ];
                then printf "|     yes      "
                else
                if [ "$auth" = "-log" ];
                        then printf "|      no      | yes |\n"
                        else printf "|      no      "
                fi
        fi
        if [ "$log" = "-log { retr stor }" ];
                then printf "|  yes  |\n"
                else
                        if [ "$log" = "{ retr stor }" ];
                                then :
                                else
                                printf "|   no  |\n"
                        fi
        fi
done < /tmp/tempfile
printf "    ------------------------------------------------------------------------\n"
```

โปรแกรมทำการแสดงรายการบนบริการเอฟทีพี (FTP)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "    ----------------------------------------\n"
printf "    |        Network         | Permitted |\n"
printf "    ----------------------------------------\n"
```

```
declare -i num=0
while read bg permit network execute
do
        num=num+1
        printf " $num. "
        printf "| $network"
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
        do
                cnt=cnt-1
                printf " "
        done
        if [ "$permit" = "permit-hosts" ];
                then printf "|   yes   |\n"
                else printf "|   no    |\n"
        fi
done < /tmp/tempfile
printf "   ----------------------------------------\n"
```

โปรแกรมทำการแสดงรายการบนบริการเอชทีทีพี (HTTP)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "timeout" | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "   ----------------------------------------\n"
printf "   |        Network          | Permitted |\n"
printf "   ----------------------------------------\n"
declare -i num=0
while read bg permit network
do
        num=num+1
        printf " $num. "
        printf "| $network"
```

```
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
        do
                cnt=cnt-1
                printf " "
        done
        if [ "$permit" = "hosts" ];
                then printf "|    yes    |\n"
                else printf "|    no     |\n"
        fi
done < /tmp/tempfile
printf "    ----------------------------------------\n"
```

โปรแกรมทำการแสดงรายการบนบริการเทลเน็ตเกตเวย์ (TELNET gateway)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "timeout" | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "    ------------------------------------------------------------\n"
printf "    |        Network        | Permitted | Authentication | Log |\n"
printf "    ------------------------------------------------------------\n"
declare -i num=0
while read bg permit network auth log
do
        num=num+1
        printf " $num."
        if [ $num -lt 10 ];
                then printf " "
        fi
        printf "|  $network"
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
```

```
        do
                cnt=cnt-1
                printf " "
        done
        if [ "$permit" = "permit-hosts" ];
                then printf "|    yes    "
                else printf "|    no     "
        fi
        if [ "$auth" = "-auth" ];
                then printf "|     yes      "
                else
                if [ "$auth" = "-log" ];
                        then printf "|      no      | yes  |\n"
                        else printf "|      no      "
                fi
        fi
        if [ "$log" = "-log { retr stor }" ];
                then printf "|  yes  |\n"
                else
                        if [ "$log" = "{ retr stor }" ];
                                then :
                                else
                                printf "|  no  |\n"
                        fi
        fi
done < /tmp/tempfile
printf "    ------------------------------------------------------------------------\n"
```

โปรแกรมทำการแสดงรายการบนบริการอาร์ล็อกอิน (RLOGIN)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "timeout" | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "    ------------------------------------------------------------------------\n"
```

```
printf "   |        Network        | Permitted | Authentication |Passok |\n"
printf "   ------------------------------------------------------------------------\n"
declare -i num=0
while read bg permit network auth pass
do
        num=num+1
        printf " $num. "
        printf "|  $network"
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
        do
                cnt=cnt-1
                printf " "
        done
        if [ "$permit" = "permit-hosts" ];
                then printf "|    yes     "
                else printf "|    no      "
        fi
        if [ "$auth" = "-auth" ];
                then printf "|      yes       "
                if [ "$pass" = "-passok" ];
                        then printf "|  yes  |\n"
                        else
                        printf "|   no  |\n"
                fi
                else
                printf "|      no        |  no  |\n"
        fi
done < /tmp/tempfile
printf "   ------------------------------------------------------------------------\n"
```

โปรแกรมทำการแสดงรายการบนบริการเทลเน็ต (TELNET)

```
#!/bin/sh
cat /tmp/tempfile | grep -v "txt" | grep -v "#" >> /tmp/tempfile.$$
mv /tmp/tempfile.$$ /tmp/tempfile
printf "   ------------------------------------------\n"
printf "   |        Network        | Permitted |\n"
printf "   ------------------------------------------\n"
declare -i num=0
while read bg permit network execute
do
        num=num+1
        printf " $num. "
        printf "|  $network"
        declare -i cnt=`len "$network"`
        cnt=27-cnt
        while [ $cnt != 0 ]
        do
                cnt=cnt-1
                printf " "
        done
        if [ "$permit" = "permit-hosts" ];
                then printf "|    yes    |\n"
                else printf "|    no     |\n"
        fi
done < /tmp/tempfile
printf "   ------------------------------------------\n"
```

กิตติกรรมประกาศ

โครงงานนี้จะสำเร็จลงไม่ได้ หากไม่ได้รับความช่วยเหลืออย่างดีจากบุคคลหลาย ๆ ท่าน ตั้งแต่เริ่มทำโครงงานจนกระทั่งการทำปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์

ขอขอบพระคุณ

ผศ. บรรจง ปิยธำรง อาจารย์ที่ปรึกษา

ขอขอบคุณ

คุณ เอกวิทย์ สวัสดิ์พีระ ผู้บริหารเครือข่ายภาควิชาวิศวกรรมคอมพิวเตอร์

ที่ได้ให้คำปรึกษา แนะนำ และให้ความช่วยเหลือตลอดมา จนกระทั่งทำโครงงานนี้สำเร็จลุล่วงไปด้วยดี

คณะผู้จัดทำ

บรรณานุกรม

[1] D. Brent Chapman and Elizabeth D. Zwicky, "*Building Internet Firewalls*", O'Reilly & Associates Inc., 1995.

[2] Paul Albitz and Cricket Liu, *"DNS and BIND"*, O'Reilly & Associates Inc., 1992.

[3] Olaf Kirch, *"Linux Network Administrator's Guide"*, O'Reilly & Associates Inc., 1995.

[4] เอกชัย รัตนดิลกชัย, *"19 Firewall กับการปกป้องอินทราเน็ตของคุณ"*, ไบต์ ไทยแลนด์, ปีที่ 4 ฉบับที่ 42, ตุลาคม 2540, หน้า 95-110.