โปรแกรมกลั่นกรองการใช้งานเอ็มเอสเอ็น

MSN Firewall

โดย

ชาญ เชาว์ปฏิภาณ

รหัสประจำตัว 45061510

อาจารย์ที่ปรึกษา

ผศ.ดร. โชติพัชร์ ภรณวลัย

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2548
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	โปรแกรมกลั่นกรองการใช้งาน MSN
นักศึกษา	นายชาญ เชาว์ปฏิภาณ
อาจารย์ที่ปรึกษา	ผศ.ดร. โชติพัชร์ ภรณวลัย
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

โปรแกรมกลั่นกรองการใช้งาน เอ็มเอสเอ็น แมสเซ็นเจอ นี้สร้างโดยอาศัยความสามารถของการกลั่นกรองแพคเกจในระดับแอพพลิเคชันเลเยอร์ของ แอลเจ็ดฟิวเตอร์ กับ ไอพีเทเบิล ซึ่งเป็นไฟล์วอลที่มากับเคอเนลของลีนุกซ์ จุดประสงค์ของโครงงานเพื่อควบคุม เอ็มเอสเอ็น โปรโตคอล ภายในองค์กร ซึ่งโปรแกรมที่พัฒนานี้ทำงานบนระบบปฏิบัติการลีนุกซ์ ในลักษณะเว็บเบสแอพพลิเคชัน ซึ่งสามารถระบุหมายเลขไอพีที่จะให้ใช้เอ็มเอสเอ็น และ ระบุอีเมล์ที่จะให้ติดต่อได้

Title	MSN Firewall
Student	Mr. Chand Chaopatipharn
Advisor	Asst. Prof. Dr. Chotipat Pornavalai
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic year	2005

ABSTRACT

MSN Firewall is a firewall of MSN protocal base on linux that filter user to use MSN . It was develop on L7 application layer filter and IPtables .Purpose of Program for control MSN in company. This program is a web base application operate on Redhat linux 9 .It can filter IP to use msn or filter email in user contact list.

กิตติกรรมประกาศ

ขอบคุณบิดา-มารดาที่คอยสนับสนุนและให้โอกาสในการศึกษาจนถึงทุกวันนี้
ขอบคุณอากู๋และอาอี้ทั้งสองคนที่ให้ความช่วยเหลือในด้านต่างๆจนจบปริญญาโท
ขอขอบคุณ ผศ.ดร โชติพัชร์ ภรณวลัย ที่ให้หัวข้อนี้ทำให้ผมได้รับความรู้ทางด้าน Network มากขึ้น
ขอบคุณนายนพพล นาวสาววาสนาที่ให้ความช่วยเหลือในการแปลสัมมนา1
ขอบคุณนางสาวนฤนาท ศิลาคุปต์ ที่ช่วยพิมพ์และจัดรูปเล่มสัมมนา2 และโปรเจครวมถึงการตรวจสอบคำถูกผิด
ขอบคุณนางสาวธารทิพย์ ที่ให้ยืมหนังสือสอบอ่านcompre นางสาวมัณฑนา นายปพจน์ ที่เริ่มต้นให้ผมร่วมทำงานในกลุ่ม

ชาญ เชาว์ปฏิภาณ
1 กุมภาพันธ์ 2549

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูปภาพ

หน้า

รูปที่

บทที่1
บทนำ

การใช้งานอินเตอร์เนตในประเทศไทยได้มีการขยายตัวของผู้ใช้งานมากขึ้น ประกอบกับเทคโนโลยีบอร์ดแบรน์ในประเทศเริ่มมีราคาถูกลงทำให้ผู้ใช้ตามบ้าน องค์กรขนาดเล็กและขนาดกลางสามารถหามาติดตั้งได้แต่เดิมนั้นบอร์ดแบรน์จะมีใช้ตามองค์กรใหญ่ๆ มหาวิทยาลัย องค์กรขนาดกลางและขนาดเล็กไม่สามารถใช้ได้ และเทคโนโลยีทางด้านการสื่อสารทางอินเตอร์เนตก็พัฒนาไปอย่างรวดเร็วแต่เดิมนั้นรับส่งกันเพียงแต่ค่าต้า ปัจจุบันสามารถส่งได้ทั้งเสียง ภาพและค่าต้า อินเตอร์เนตได้เข้ามามีบทบาทในการดำเนินธุรกิจซึ่งช่วยให้การค้าระหว่างประเทศหรือภายในประเทศเป็นเรื่องง่ายขึ้น

1.1ความเป็นมาและความสำคัญของปัญหา

การสื่อสารทางอินเตอร์เนตได้เข้ามามีบทบาทสำคัญในการดำเนินธุรกิจขององค์กรซึ่งช่วยลดค่าใช้จ่ายในการสื่อสารระหว่างผู้ประกอบการกับผู้ประกอบการ ระหว่างผู้ประกอบการกับลูกค้าและการสื่อสารกันภายในองค์กร เอ็มเอสเอ็น เป็นโปรแกรมสื่อสารทางอินเตอร์เนตของบริษัทไมโครซอฟร์ที่มีความสามารถในการสื่อสาร โดยสามารถสื่อสารกันได้โดยวิธีพิมพ์ข้อความทางคีย์บอร์ด(chat) ทางไมโครโฟน และทางกล้องวีดีโอ (webcam) และยังมีความสามารถในการรับส่งไฟล์ระหว่างผู้สื่อสาร รวมถึงลูกเล่นต่างๆที่มีในเวอร์ชันใหม่ๆ จากความสามารถเหล่านี้ของ เอ็มเอสเอ็น จึงทำให้ผู้ที่ใช้ระบบปฏิบัติการวินโดส์มีโปรแกรมนี้ติดตั้งอยู่ในเครื่องคอมพิวเตอร์ของตน รวมถึงคอมพิวเตอร์ตามสถานที่ต่างๆที่ใช้ระบบปฏิบัติการวินโดส์เอ็กซ์พีก็มีโปรแกรม เอ็มเอสเอ็นติดมาด้วย การใช้ เอ็มเอสเอ็น ของพนักงานในเวลาทำงานโดยที่ไม่เกี่ยวข้องกับงานขององค์กร การที่พนักงานส่งไฟล์สำคัญขององค์กรออกไปให้ผู้อื่นผ่านทาง เอ็มเอสเอ็น ทำให้ข้อมูลขององค์การรั่วไหลได้ง่าย จากปัญหาดังกล่าวจึงเป็นที่มาของโปรแกรมกลั่นกรองการใช้งาน เอ็มเอสเอ็น (MSN Firewall)

1.2 วัตถุประสงค์ของการพัฒนาระบบ

- เพื่อศึกษา เอ็มเอสเอ็น โปรโตคอล
- เพื่อศึกษาการกลั่นกรอง แพ็คเกจ ในระดับชั้นแอพพลิเคชัน
- เพื่อวิเคราะห์และออกแบบรวมถึงการพัฒนาโปรแกรมที่ควบคุม เอ็มเอสเอ็น โปรโตคอล

1.3 ประโยชน์ที่คาดว่าจะได้รับ

ผู้ดูแลสารสนเทศขององค์กรสามารถควบคุมการใช้งาน เอ็มเอสเอ็น ให้เกิดประโยชน์แก่องค์กรและรักษาข้อมูลขององค์กรไม่ให้รั่วไหลออกไปได้ทาง เอ็มเอสเอ็น

1.4 ขอบเขตของการพัฒนาระบบงาน

- สร้างรูปแบบของการดักจับแพ็คเกจในระดับชั้นแอพพลิเคชัน บนโปรโตคอล เอ็มเอสเอ็น ที่สามารถทำงานร่วมกับ แอลเจ็ด ได้
- สร้าง เว็ปเบสแอพพลิเคชัน ที่ใช้ในการสร้างกฎสำหรับการใช้งาน เอ็มเอสเอ็น

1.5 ทฤษฎีที่ใช้ในการพัฒนาระบบ

- หลักการของโปรโตคอล เอ็มเอสเอ็น
- การใช้งาน ไอพีเทเบิล และ แอลเจ็ด
- ความรู้เรื่อง Regular expression

1.6 ขั้นตอนการพัฒนาระบบ

- ศึกษาโปรโตคอล เอ็มเอสเอ็น และการใช้งาน L7
- ศึกษาการเขียน Regular expression และสร้างรูปแบบการดักจับ
- พัฒนาโปรแกรมกลั่นกรองการใช้งาน เอ็มเอสเอ็น
- ทดสอบการทำงานของโปรแกรมที่พัฒนาแล้ว
- สรุปผลการทำงาน

บทที่ 2

MSN Messenger

เอ็มเอสเอ็น แมสเซนเจอร์ เป็นโปรแกรม Instant Message (IM) ที่ถูกพัฒนาขึ้นโดย Hotmail ผู้ให้บริการอีเมล์ฟรี ยักษ์ใหญ่ แต่เนื่องจากความง่ายในการใช้งาน บริษัทไมโครซอฟร์ จึงได้ซื้อโปรแกรมนี้มาเพื่อเป็นโปรแกรมหนึ่งที่มีมากับระบบปฏิบัติการของไมโครซอฟร์ แต่ผู้ใช้จะต้องมีบัญชีรายชื่อของฮอทเมล์ ก่อนจึงจะสามารถใช้งานได้ อาจรู้จักกันในชื่อของ ".NET Messenger" หรือ "Windows Messenger" (ซึ่งติดมากับ วินโดว์ เอ็กซ์พี)

เอ็มเอสเอ็น แมสเซนเจอร์ใช้โปรโตคอล เอ็มเอสเอ็นเอ็มเอส (MSN Messenger Service Protocol) ในการติดต่อสื่อสารกันระหว่างเครื่อง ไคเอ็น และ เซิฟเวอร์ โดย ไมโครซอฟร์ ได้ทำการเผยแพร่รายละเอียดของโปรโตคอลนี้ในเดือน กรกฎาคม ค.ศ. 1999 ในเอกสาร RFC2026 (สามารถศึกษาได้จาก http://www.hypothetic.org/docs/msn/ietf_draft.php) ในปัจจุบันนี้โปรโตคอล เอ็มเอสเอ็นเอ็มเอส ได้พัฒนาไปเร็วมากเพราะว่ามีผู้นิยมใช้งานโปรแกรม เอ็มเอสเอ็น แมสเซนเจอร์มากมายทั่วโลก โดยขณะนี้โปรโตคอลได้พัฒนามาถึง เอ็มเอสเอ็นพี เวอร์ชัน 12 และ ตัวโปรแกรมได้พัฒนามาถึง เวอร์ชัน 7.5 แล้ว

2.1 การเชื่อมต่อ

โปรโตคอล MSNMS ใช้ TCP Socket ในการสื่อสาร โดย เซิฟเวอร์ จะใช้ พอตร์ 1863 แต่เซิฟเวอร์ สามารถใช้ พอตร์ อื่นได้โดยจะมีการแจ้งมาบอก ไคลเอ็น ในส่วนของ ไคลเอ็น จะใช้พอตร์ ใดก็ได้ที่มีค่ามากกว่า 1024 โดยการสื่อสารเกือบทั้งหมดจะใช้ พอตร์ นี้ ยกเว้นการส่ง ไฟล์จะมีการใช้ พอตร์ อื่น แต่ว่าในบางสถานการณ์ที่ไม่สามารถสื่อสารด้วยโปรโตคอล เอ็มเอสเอ็นเอ็มเอส ได้นั้นสามารถที่จะห่อข้อมูลที่จะส่งลงใน HTTP Command ได้ โดยจะติดต่อไปยัง gateway.messenger.hotmail.com ที่ พอตร์ 80 ในการสื่อสารนั้นจะมี เซิฟเวอร์ อยู่ 3 ชนิดดังนี้ Dispatch Server จะเป็นเหมือนกับจุดเริ่มต้นในการสื่อสาร มีหน้าที่ในการตรวจสอบว่าโปรโตคอลที่ ไคลเอ็น จะใช้ เซิฟเวอร์ รองรับหรือไม่ และ โปรแกรม ที่ ไคลเอ็น ใช้เป็น เวอร์ชัน หรือระบบปฏิบัติการของ ไคลเอ็น เป็นอะไร นอกจากนี้อีกหน้าที่ที่สำคัญคือทำหน้าที่เลือก Notification Server ที่เหมาะสม

กับ ไคลเอ็น เพื่อให้ ไคลเอ็น ทำการเชื่อมต่อไปยัง Notification Server ต่อไป โดย URL ของ Dispatch Server คือ messenger.hotmail.com

Notification Server เป็น เซิฟเวอร์ ที่จะทำการเก็บ Session ทั้งหมด โดยที่เกือบทุกเหตุการณ์จะเกิดขึ้นที่ Notification Server เช่นการเปลี่ยนสถานะ,การร้องขอในการพูดคุย (Chat Request) เป็นต้น

Switchboard Server เป็นเสมือนกับประตูที่ใช้ในการพูดคุยสนทนากัน โดยทุก ๆ การสนทนาจะต้องทำการสร้าง Session ใหม่ทุกครั้ง นอกจากนี้ Switchboard Server ยังเป็น เซิฟเวอร์ ที่ใช้ในการเชิญและตกลงวิธีการในการส่ง File หรือ Voice Chat

เอ็มเอสเอ็นเอ็มเอส เป็นโปรโตคอลที่เป็น ASCII-Based และข้อมูลที่สื่อสารกันสามารถรวมอยู่ในแพ็คเกจ เดียวกันได้ โดยจะใช้การขึ้นบรรทัดใหม่ (\r\n) หรือ ขนาดของข้อความเป็นการบอกจุดสิ้นสุดของข้อมูล โดยประเภทของข้อมูลที่สื่อสารกันระหว่าง ไคลเอ็น และ เซิฟเวอร์ แบ่งเป็น 2 ประเภทดังนี้ คำสั่ง (Commands) ข้อมูลส่วนใหญ่ที่สื่อสารกันมักจะเป็นคำสั่ง โดยหนึ่งคำสั่งจะมีรหัสแทนที่เป็นตัวอักษร 3 ตัว เช่น VER ซึ่งเป็นคำสั่งในการตรวจสอบโปรโตคอล ต่อจากรหัสจะเป็น Parameter ต่าง ๆ ตามแต่ละคำสั่งจะกำหนดไว้ โดยหากมีหลาย Parameter จะแยกด้วยช่องว่าง และการสิ้นสุดคำสั่งจะแทนด้วยการขึ้นบรรทัดใหม่ (\r\n)

ข้อความ (Messages) เป็นคำสั่งหนึ่งแต่ว่ามีความแตกต่างจากคำสั่งทั่ว ๆ ไปเล็กน้อย โดยมีรหัสแทนคือ “MSG” ในคำสั่ง MSG นี้จะสามารถส่งข้อมูลที่เป็นการขึ้นบรรทัดใหม่ได้ ดังนั้นจึงใช้วิธีการอื่นในการบอกจุดสิ้นสุด นั่นคือการบอกขนาดของข้อมูลที่ส่ง ดังตัวอย่าง

MSG 3 A 157

MIME-Version: 1.0

Content-Type: text/plain; charset=UTF-8

X-MMS-IM-Format: FN=Microsoft%20Sans%20Serif; EF=I; CO=000000; CS=0; PF=22

Hello! How are you?

บรรทัดแรกจะมีการบอกขนาดของข้อมูลที่ส่งมานั่นคือ 157 Bytes นั่นเอง บรรทัดถัดมาจะเป็นการบอกถึง MIME Version และลักษณะของตัวอักษร และ บรรทัดสุดท้ายคือข้อความที่ส่ง โดยระหว่างรายละเอียดของ MIME และข้อความจะถูกขั้นด้วยการขึ้นบรรทัดใหม่ 2 ครั้ง โดยถ้าทำการส่งข้อมูลเป็นการขึ้นบรรทัดใหม่จะถูกนับเป็น 2 ไบต์

หากว่ามีข้อผิดพลาดขึ้นระหว่างการสื่อสาร เซิฟเวอร์ จะทำการส่งคำสั่ง Error Code มาแจ้งยังฝั่ง ไคลเอ็น โดยจะแทนด้วยตัวเลข 3 หลัก และตามด้วย Transaction ID ของคำสั่งที่เกิดความผิดพลาด โดย Error Code เป็นดังนี้

200 Syntax error

201 Invalid parameter

205 Invalid user

206 Domain name missing

207 Already logged in

208 Invalid username

209 Invalid username

210 User list full

215 User already there

216 User already on list

217 User not online

218 Already in mode

219 User is in the opposite list

280 Switchboard failed

281 Transfer to switchboard failed

300 Required field missing

302 Not logged in

500 Internal server error

501 Database server error

510 File operation failed

520 Memory allocation failed

600 Server is busy

601 Server is unavailable

602 Peer name server is down

603 Database connection failed

604 Server is going down

707 Could not create connection

711 Write is blocking

712 Session is overloaded

713 Too many active users

714 Too many sessions

715 Not expected

717 Bad friend file

911 Authentication failed

913 Not allowed when offline

920 Not accepting new users

924 Passport account not yet verified

ทุก ๆ คำสั่งที่ส่งจาก ไคลเอ็น ไปยัง เซิฟเวอร์ จะต้องมีการส่ง Transaction ID ไปด้วย เมื่อ เซิฟเวอร์ ทำการตอบกลับคำสั่งก็จะตอบกลับด้วย Transaction ID เดียวกับคำสั่งที่ส่งออกไป โดย Transaction ID จะมีค่าระหว่าง 0 ถึง 4294967295 (2^32 – 1) โดย Transaction ID จะอยู่หลังจากรหัสคำสั่ง แต่ว่าในบางครั้ง เซิฟเวอร์ จะส่งคำสั่งมายัง ไคลเอ็น ซึ่งเป็นคำสั่งที่ไม่ได้เกิดจากการร้องขอของ ไคลเอ็น ซึ่งอาจจะส่งมาด้วย Transaction ID ที่เป็น 0 ได้ ดังนั้นโดยทั่ว ๆ ไปแล้ว Transaction ID จะไม่เป็น 0 สำหรับการส่ง Password ไปยัง เซิฟเวอร์ จะต้องทำการเข้ารหัสเสียก่อนซึ่งวิธีการเข้ารหัสจะขึ้นอยู่กับ เวอร์ชัน ของโปรโตคอล โดยปัจจุบันโปรโตคอล Version MSNP10 นั้นจะใช้วิธีการ SSL ในการเข้ารหัสแบบ SSL ซึ่งเป็นการเข้ารหัสที่มีพื้นฐานจากเรื่อง Public Key และ Private Key ซึ่งเป็นวิธีการที่เว็บไซต์เกี่ยวกับ อีคอมเมอซ์นิยมใช้ ซึ่งสามารถศึกษาได้เพิ่มเติมจาก

http://home.netscape.com/security/techbriefs/ssl.html

2.2 การ Sign – In

ขั้นแรก ไคลเอ็น จะต้องทำการติดต่อไปยัง Dispatch Server เพื่อทำการตรวจสอบเวอร์ชัน โปรโตคอลที่ใช้และค้นหา Notification Server ที่เหมาะสม โดยวิธีการเลือก Notification Server นั้นทางไมโครซอฟร์ ไม่ได้เปิดเผย

ไคลเอ็น จะทำการสร้าง TCP Socket เชื่อมต่อไปยัง messenger.hotmail.com ที่ พอตร์ 1863 ซึ่งจะเป็นเช่นนี้เสมอ โดยหลังจากที่สามารถสร้างการเชื่อมต่อได้แล้วนั้น ไคลเอ็น จะส่งคำสั่งไปเพื่อตรวจสอบโปรโตคอลดังนี้

<<< VER 4 MSNP9 MSNP8 CVR0 \r\n

โดยค่า Parameters ที่ตามมาจากคำสั่ง VER คือ Transaction ID และ ลีส ของโปรโตคอลที่ไคลเอ็น จะใช้ได้ ส่วน CVR 0 หมายความว่าต่อจากคำสั่ง VER นี้จะมีการใช้คำสั่ง CVR ด้วยซึ่ง

คำสั่ง CVR นี้จะมีตั้งแต่ MSNMS Version MSNP7 ขึ้นไป การสิ้นสุดคำสั่งจะใช้การขึ้นบรรทัดใหม่แสดงถึงจุดสิ้นสุดคำสั่ง และเมื่อ เซิฟเวอร์ ได้รับคำสั่งนี้แล้วจะตอบกลับด้วยคำสั่ง VER ที่มี Transaction ID เดียวกับ Transaction ID ที่ได้รับ และตามด้วยโปรโตคอลที่ เซิฟเวอร์ รองรับได้ ซึ่งหากไม่มีโปรโตคอลที่ เซิฟเวอร์ รองรับเลยจะตอบกลับด้วยค่า Parameter เป็น 0 เช่น

<<<VER 4 MSNMSN CVR0 \r\n

>>>VER 4 0 \r\n

หลังจากนั้น ไคลเอ็น จะส่งคำสั่ง CVR ไปเพื่อบอกถึงรายละเอียดของโปรแกรมและระบบปฏิบัติการของ ไคลเอ็น ให้กับ เซิฟเวอร์ โดยตัวอย่างคำสั่งเป็นดังนี้

<<< CVR 5 0x0409 winnt 5.1 i386 MSNMSGR 6.0.0602 MSNMSGR naul220@hotmail.com \r\n

โดยค่า Parameter ที่ตามมาคือ Transaction ID ตามด้วยค่า Locale ID ของ ไคลเอ็น ซึ่งจะบอกถึงภาษาเริ่มต้น Software ไคลเอ็น ค่า Parameter ที่สี่และห้า คือ ระบบปฏิบัติการและรุ่นของระบบปฏิบัติการที่ ไคลเอ็น ใช้อยู่ ค่า Parameter ที่ 5 จะบอกถึงสถาปัตยกรรมของคอมพิวเตอร์ของไคลเอ็น ค่า Parameter ที่หกถึงแปดจะบอกถึง Software ไคลเอ็น และ เวอร์ชัน ของ Software นั้น และ Parameter ที่เก้าจะบอกถึง Account ของผู้ใช้ โดย เซิฟเวอร์ จะตอบกลับดังนี้

>>> CVR 5 6.0.0602 6.0.0602 6.0.0268 http://download.microsoft.com/download /8/a/4/8a42bcae-f533-4468-b871-d2bc8dd32e9e/SetupDl.exe http://messenger.msn.com \r\n

โดยค่า Parameter ที่สองและสามจะมีค่าเท่ากันคือ เวอร์ชัน ที่ เซิฟเวอร์ แนะนำ Parameter ที่สี่คือ เวอร์ชัน ที่ ไคลเอ็น ใช้อยู่ Parameter ที่ห้าคือ URL ที่สามารถจะไป Download ได้ และค่า Parameter ทีหกคือ URL ที่ผู้ใช้สามารถไปค้นหารายละเอียดเพิ่มเติมเกี่ยวกับ เอ็มเอสเอ็น แมสเซนเจอร์ ได้

หลังจากนั้นหากโปรโตคอลเป็น Version MSNP7 หรือ ต่ำกว่าจะต้องมีการส่งคำสั่งเพื่อสอบถามวิธีในการเข้ารหัสข้อมูลนั่นคือคำสั่ง INF ซึ่ง เซิฟเวอร์ จะตอบกลับด้วยวิธีการเข้ารหัสที่จะใช้ แต่ใน Version MSNP8 ขึ้นไปจะใช้วิธี SSL เป็นค่ามาตรฐาน ดังนั้นในการทดลองนี้หลังจากเสร็จสิ้นคำสั่ง CVR จึงไม่มีคำสั่ง INF เกิดขึ้น โดยหลังจากที่ตกลงวิธีการเข้ารหัสแล้ว ไคลเอ็น จะส่งคำสั่ง USR กลับไปเพื่อยืนยันอีกครั้งดังนี้

<<<USR 6 TWN I nual220@hotmail.com \r\n

โดยค่า Parameter แรกคือ Transaction ID ตามด้วยรหัสของวิธีการเข้ารหัสซึ่ง TWN หมายถึงการเข้ารหัสแบบ SSL ค่า Parameter ถัดมาจะเป็น I เสมอหมายความว่าเป็นการเริ่มต้นตรวจสอบสิทธิ์ผู้ใช้ และ Parameter สุดท้ายคือ User Account

Dispatch Server จะทำการตรวจสอบรายละเอียดของวิธีการเข้ารหัสและ User Account โดยไม่มีการตรวจสอบสิทธิ์แต่อย่างใด และจะตอบกลับด้วยคำสั่งดังนี้

>>>XFR 6 NS 207.146.106.147:1863 0 207.146.104.20:1863 \r\n

โดยคำสั่งนี้เป็นคำสั่งในการสั่งให้ ไคลเอ็น ไปทำการเชื่อมต่อกับ Notification Server โดย Parameter NS คือ Notification Server ซึ่งจะตามมาด้วย IP Address และ พอตร์ ของ Notification Server และ Parameter สุดท้ายจะเป็น IP Address และ พอตร์ ของ เซิฟเวอร์ ที่ ไคลเอ็น กำลังเชื่อมต่ออยู่ ส่วนค่า Parameter ที่เป็น 0 จะเป็นมาตรฐานหากโปรโตคอลที่ใช้เป็น เวอร์ชัน ที่สูงกว่า MSNP2 แต่ถ้าหากต่ำกว่าจะไม่มี Parameter นี้ หลังจากที่ ไคลเอ็น ได้รับคำสั่ง XFR แล้วจะยกเลิก TCP Socket ที่เชื่อมต่อกับ Dispatch Server เสีย แล้วทำการสร้าง TCP Socket เชื่อมต่อกับ Notification Server ตามที่ Dispatch Server ได้แจ้งมา โดยการเชื่อมต่อจะต้องใช้คำสั่ง VER และ CVR เช่นเดียวกับที่ได้กล่าวไปแล้ว หลังจากที่เสร็จสิ้นการตรวจสอบโปรโตคอลและข้อมูลของ ไคลเอ็น แล้วจะเป็นการตรวจสอบสิทธิ์ผู้ใช้โดยในการทดลองนี้ได้ใช้โปรโตคอล MSNP9 ดังนั้นจึงเป็นการตรวจสอบสิทธิ์โดยใช้การเข้ารหัสแบบ SSL โดยมีขั้นตอนดังนี้

ไคลเอ็น ส่งคำสั่ง USR พร้อม Parameter ต่าง ๆ ดังนี้

<<<USR 9 TWN I nual220@hotmail.com \r\n

ซึ่งค่า Parameter I หมายถึงการเริ่มต้นการตรวจสอบสิทธิ์ผู้ใช้งานดังที่ได้กล่าวไปแล้ว Notification Server จะทำการตอบกลับด้วยรายละเอียด และ เซิฟเวอร์ ที่จะต้องไปทำ SSL ดังนี้

>>> USR 9 TWN S lc=1033,id=507,tw=40,fs=1, ru=http%3A%2F
%2Fmessenger%2Emsn%2Ecom,ct=1062931397,kpp=1,
kv=5,ver=2.1.0173.1,tpf=4cd7ffaa26d10ef6f5b51f0a177ba302
\r\n

โดยจะแทนค่า Parameter I ด้วย Parameter S หมายความว่าเป็นการตรวจสอบสิทธิ์ที่ไม่ใช่ครั้งแรก ซึ่งต่อจากนี้จะเป็น S เสมอ

ต่อมา ไคลเอ็น จะทำการเชื่อมต่อไปยัง เซิฟเวอร์ ตามที่ Notification Server แจ้งมาเพื่อทำ SSL โดยค่ามาตรฐานจะเป็นที่ login.passport.com ที่ พอตร์ 443

หลังจากทำ SSL ได้แล้ว ไคลเอ็น จะส่งผลลัพธ์ที่ได้ไปยังเครื่อง Notification Server เพื่อใช้ในการยืนยันสิทธิ์ครั้งถัด ๆ ไป โดยค่า Parameter จะเป็นผลลัพธ์จากการทำ SSL

ซึ่งหากการทำ SSL ไม่สำเร็จจะทำให้การ Sign-In ไม่สามารถสำเร็จได้

Notification Server จะตอบกลับมาด้วยคำสั่ง USR OK เพื่อเป็นการสิ้นสุดการตรวจสอบสิทธิ์ของผู้ใช้ดังนี้

>>>USR 10 OK nual220@hotmail.com nual220@hotmail.com 1 0 \r\n

โดยค่า Parameter ที่ต่อจาก OK คือ User Account และ User Screen name

หลังจากที่ทำการตรวจสอบสิทธิ์เรียบร้อยแล้ว Notification Server จะส่งคำสั่ง MSG พร้อมด้วยข้อมูลของผู้ใช้งานกลับมาพร้อมด้วยรายละเอียดของ IP Address และ พอตร์ ของ ไคลเอ็น และผลลัพธ์จากการทำ SSL ข้างต้น ดังนี้

MSG Hotmail Hotmail 440\r\n

MIME-Version: 1.0\r\n

Content-Type: text/x-msmsgsprofile; charset=UTF-8\r\n

LoginTime: 1050223062\r\n

EmailEnabled: 0\r\n

MemberIdHigh: 85040\r\n

MemberIdLow: -517030579\r\n

lang_preference: 1033\r\n

preferredEmail: nual220@hotmail.com\r\n

country: TH\r\n

PostalCode: 90201\r\n

Gender: m\r\n

Kid: 0\r\n

Age: \r\n

BDayPre: 5\r\n

Birthday: 0\r\n

Wallet: 0\r\n

Flags: 1027\r\n

sid: 507\r\n

kv: 4\r\n

MSPAuth: 4sCuECZ4UsAaBIy0AIsk!c9bWcuATTmuQ$$\r\n

\r\n

Client IP: 161.246.53.141 \r\n

Client Port:36365 \r\n \r\n

โดยค่า Parameter ที่สำคัญคือ MSPAuth ที่จะใช้ในการ Log-In เข้าสู่ E-Mail โดยอัตโนมัติ และ ไคลเอ็น พอตร์ คือ พอตร์ ที่ ไคลเอ็น จะใช้ในการเชื่อมต่อ

2.3 การร้องขอ Contact List

ผู้ใช้งานทั่วไปมักคิดว่า Contact List ของ เอ็มเอสเอ็น แมสเซนเจอร์ คือ ลีส ของผู้ใช้งานที่เราจะคุยด้วยที่แสดงขึ้นมาทุกครั้งที่เรา Sign-In แต่ในความเป็นจริงแล้วนั้น Contact List มีทั้งหมด 4 ประเภทดังนี้

Forward List (FL) คือ ลีส ของ User Account ที่ผู้ใช้งานคนนั้น ๆ ได้เพิ่มเข้าไป FL นี้คือ ลีส รายชื่อที่แสดงขึ้นมาทุก ๆ ครั้งที่ผู้ใช้งานคนนั้น ๆ Sign-In ในปัจจุบันนี้ FL ได้จำกัดจำนวน User Account อยู่ที่ 150 ชื่อเท่านั้น

Reverse List (RL) คือ ลีส ของผู้ใช้งานคนอื่น ๆ ที่ได้เพิ่ม User Account ของเราเข้าไปใน FL ของผู้ใช้งานคนนั้น ๆ ซึ่งหมายความว่าหากผู้ใช้งานคนอื่น ๆ ได้ลบ User Account ของเราออกจาก FL User Account ของผู้ใช้งานคนนั้นจะถูกลบไปจาก RL ของเราโดยอัตโนมัติ

Allow List (AL) คือ ลีส ของ User Account ที่เราอนุญาตให้เห็นสถานะ การ Online ของเราได้ ซึ่งหากเราลบ User Account ของผู้ใช้งานคนอื่น ๆ ออกจาก FL ของเรา แต่มันจะไม่ลบออกจาก AL ซึ่งหมายความว่าผู้ใช้งานคนนั้น ๆ จะยังมองเห็นถึงสถานะ การ Online ของเราได้เสมอ

Block List (BL) คือ ลีส ของ User Account ที่เราไม่อนุญาตให้เห็นสถานะ การ Online ของเราได้

นอกจากนี้ เอ็มเอสเอ็น แมสเซนเจอร์ ยังมี ลีส อีกประเภทนั่นคือ Group List คือ ลีส ของ Group ที่ผู้ใช้งานได้กำหนดไว้ใน Forward List ซึ่งแต่ละ Group จะมีรหัสเรียกว่า Group ID

ในการแก้ไขเปลี่ยนแปลง ลีส แต่ละครั้งนั้น เซิฟเวอร์ จะทำการเก็บ เวอร์ชัน ของ ลีส ไว้ด้วยโดย เวอร์ชัน ของ ลีส จะมีค่าตั้งแต่ 0-4264967295 (2^32-1) โดย ลีส เวอร์ชัน นี้จะมีประโยชน์ในกรณีที่เครื่อง ไคลเอ็น ได้ทำการเก็บรายละเอียด ลีส ไว้แล้วซึ่งหาก เวอร์ชัน ของ ลีส ตรงกันก็ไม่จำเป็นที่ เซิฟเวอร์ ต้องทำการส่ง ลีส มาให้ ไคลเอ็น อีกครั้ง

ในการร้องขอ ลีส ไคลเอ็น จะทำการส่งคำสั่ง SYN พร้อมด้วย ลีส เวอร์ชัน ที่เครื่อง ไคลเอ็น มีอยู่ ซึ่งหากในเครื่องนั้นไม่ได้ทำการเก็บ ลีส ของผู้ใช้งานคนนั้น ๆ ไว้ค่า ลีส เวอร์ชัน จะเป็น 0 ดังนี้

<<<SYN 10 0 \r\n (กรณีไม่ได้ทำการเก็บ ลีส ไว้)

<<<SYN 10 1234 \r\n (กรณีที่ได้ทำการเก็บ ลีส เอาไว้ โดย 1234 คือตัวอย่าง ลีส เวอร์ชัน)

ซึ่ง เซิฟเวอร์ จะทำการตอบกลับมาซึ่งหาก ลีส เวอร์ชัน เท่ากันจะตอบกลับมาเพียงแค่สถานะการ Online ของ User Account ที่มีอยู่ใน FL ของเราเท่านั้น แต่ถ้า ลีส เวอร์ชัน เป็น 0 หรือต่ำกว่าที่ เซิฟเวอร์ มีอยู่ เซิฟเวอร์ จะทำการตอบกลับด้วยรายละเอียดทั้งหมด ดังตัวอย่าง

<<< SYN 8 0\r\n

>>> SYN 8 52 7 4\r\n

>>> GTC A\r\n

>>> BLP BL\r\n

>>>LSG 0 Other%20Contacts 0 \r\n

>>>LSG 1 Coworkers 0 \r\n

>>>LSG 2 Friends 0 \r\n

>>>LSG 3 Family 0 \r\n

>>>LST inuokashi@hotmail.com inuokashi@hotmail.com 1 2 \r\n

>>>LST kpkate@hotmail.com KATE 13 0 \r\n

>>>BPR MOB Y \r\n

>>>LST navy_srifa@hotmail.com navy_srifa@hotmail.com 11 2 \r\n

>>>LST nual220@hotmail.com nual 11 2 \r\n

>>>LST alpha3210@hotmail.com NON 11 0 \r\n

>>>BPR PHH 66%20023274613

>>>BPR PHM 66%20014851457

>>>LST alpha3211@hotmail.com alpha3211@hotmail.com 11 0 \r\n

>>>LST taotoon_online_net@hotmail.com taoton 8 \r\n

ในกรณีนี้ ไคลเอ็น ส่งคำสั่ง SYN ด้วย ลีส เวอร์ชัน เป็น 0 ดังนั้น Notification Server จะตอบกลับด้วยคำสั่ง SYN และ ลีส เวอร์ชัน ที่ เซิฟเวอร์ เก็บไว้อยู่ในกรณีนี้คือ 8 ค่า Parameter ที่สามคือจำนวนคำสั่ง LST ที่จะถูกส่งตามมา และ 4 คือจำนวนคำสั่ง LSG ที่จะถูกส่งตามมาเช่นกัน

หลังจากนั้น Notification Server จะส่งคำสั่ง LSG กลับมาพร้อมด้วยรายละเอียดของ Group ต่าง ๆ โดยค่า Parameter ที่ตามมาจะประกอบด้วย Group ID, Group Name และ 0 ตามลำดับ ซึ่งในกรณีที่ผู้ใช้งานไม่ได้เลือกให้มีการใช้ Group Notification Server จะตอบกลับด้วยค่า ลีส เวอร์ชัน เป็น 0~0

หลังจากนั้น Notification Server จะส่งรายละเอียดของ Forward List ด้วยคำสั่ง LST โดยค่า Parameter ที่ตามมาคือ User Account, User Screen name, List Operator Value และ Group ID ซึ่งค่า List Operator Value จะมีตั้งแต่ MSNP8 ขึ้นไป โดยค่านี้จะบอกว่า User Account นี้อยู่ใน ลิสประเภทใดบ้าง โดยค่า List Operator Value เป็นดังนี้

Forward List = 1

Reverse List = 8

Allow List = 2

Block List = 4

เช่นถ้า User Account อยู่ใน FL,AL,RL ค่า List Operator Value จะเท่ากับ 1+8+2 = 11

ส่วนคำสั่ง GTC และ BLP คือ Privacy Setting ของผู้ใช้งานซึ่งจะเก็บไว้ที่ เซิฟเวอร์ โดย GTC จะหมายถึงการที่หากมีผู้ใช้งานคนอื่น ๆ ทำการเพิ่ม User Account ของเราเข้าไปใน FL เราต้องการให้มีเมนูเพื่อให้เลือกว่าจะเพิ่มชื่อของผู้ใช้งานคนนั้น ๆ ลงไปใน AL หรือ BL หรือไม่ ซึ่งหากค่า Parameter ที่ตามมาเป็น A คือต้องการ แต่ถ้าเป็น N คือไม่ต้องการ ซึ่งค่าเริ่มต้นจะเป็น A เสมอ สำหรับคำสั่ง BLP จะมีค่า Parameter ตามมาที่เป็นไปได้ 2 ค่าคือ AL และ BL โดย AL จะหมายความว่าเราจะรับทุก ๆ message ที่มาจาก User Account ใด ๆ ยกเว้นที่อยู่ใน Block List ส่วน BL จะหมายความว่าจะรับ message เฉพาะ User ที่อยู่ใน Allow ลิส เท่านั้น

คำสั่ง BPR จะเป็นคำสั่งในการรับรายละเอียดข้อมูลเกี่ยวกับหมายเลขโทรศัพท์ โดยคำสั่งจะตามมาจาก LST นั้น โดยรูปแบบของคำสั่ง BPR จะมี 4 รูปแบบคือ

>>>BPR PHH 66%20023274613\r\n

>>>BPR PHW66%20027171524 \r\n

>>>BPR PHM 66%20014851457\r\n

>>>BPR MOB Y \r\n

ถ้า Parameter เป็น PHH จะหมายถึงหมายเลขโทรศัพท์บ้าน ซึ่งจะตามด้วยหมายเลขโทรศัพท์ ซึ่งในกรณีที่ไม่ได้กำหนดไว้คำสั่งนี้จะไม่ถูกส่งกลับมา ส่วน PHW คือหมายเลขโทรศัพท์ที่ทำงาน และ PHM คือหมายเลขโทรศัพท์เคลื่อนที่ สำหรับรูปแบบสุดท้ายคือ บอกว่าเราได้อนุญาตให้คนอื่นติดต่อมาทาง Mobile หรือไม่ถ้า Y คือใช่ ถ้า N คือไม่ใช่

2.4 การเตือนเมื่อมี E-Mail ที่ยังไม่ได้อ่าน หรือ มี E-Mail เข้ามาใหม่

ในการ Sign-In ทุกครั้งหาก E-Mail ของเรามี Mail ที่ยังไม่ได้อ่านอยู่จะมีการเตือนให้เราเข้าไปอ่านโดยใช้คำสั่ง MSG โดยมีคำสั่งดังนี้

```
⊟ MSN Messenger Service
    MSG Hotmail Hotmail 221\r\n
    MIME-Version: 1.0\r\n
    Content-Type: text/x-msmsgsinitialemailnotification; charset=UTF-8\r\n
    \r\n
    Inbox-Unread: 1\r\n
    Folders-Unread: 0\r\n
    Inbox-URL: /cgi-bin/HoTMaiL\r\n
    Folders-URL: /cgi-bin/folders\r\n
    Post-URL: http://www.hotmail.com\r\n
    \r\n
```

และในกรณีที่มี E-Mail ใหม่เข้ามา Notification Server จะส่งคำสั่ง MSG มาเตือนเช่นเดียวกันดังนี้

```
⊟ MSN Messenger Service
    MSG Hotmail Hotmail 355\r\n
    MIME-Version: 1.0\r\n
    Content-Type: text/x-msmsgsemailnotification; charset=UTF-8\r\n
    \r\n
    From: s6066201@kmitl.ac.th\r\n
    Message-URL: /cgi-bin/getmsg?msg=MSG1063177885.134&start=178&len=1218&curmbox=ACTIVE\r\n
    Post-URL: https://loginnet.passport.com/ppsecure/md5auth.srf?lc=1033\r\n
    Subject: =?tis-620?Q?Hey?=\r\n
    Dest-Folder: ACTIVE\r\n
    From-Addr: s6066201@kmitl.ac.th\r\n
    id: 2\r\n
```

2.5 การ Sign-Out

หากผู้ใช้งานทำการ Sign-Out ก็จะทำการส่งคำสั่ง OUT ออกไปยัง Notification Server ดังตัวอย่าง

<<<OUT \r\n

Notification Server จะทำการส่งข้อมูลไปยังทุก ๆ เครื่องที่อยู่ใน Reverse List ของเราโดยจะส่งเป็นคำสั่งการเปลี่ยนสถานะ ดังตัวอย่าง

>>>FLN natapon_p@msn.com \r\n

2.6 การเปลี่ยนสถานะ การ Online และ การรับสถานะ การ Online ของ User Account อื่น

สถานะ การ Online นั้น หมายถึงสถานะของเราซึ่งผู้ที่จะเห็นได้นั้นจะต้องอยู่ใน Allow List ของเราเท่านั้น โดยสถานะ Online นั้นจะมีสถานะย่อย ๆ อยู่ด้วย โดยแต่ละสถานะจะแทนด้วยรหัสเป็นตัวอักษรภาษาอังกฤษ 3 ตัว ดังนี้

NLN = Online

FLN = Offline

HDN = Hidden or Appear Offline

BSY = Busy

IDL = Idle

BRB = Be Right Back

AWY = Away

PHN = On The Phone

LUN = Out To Lunch

ในการ Sign-In หลังจากที่รับรายละเอียดของ ลีส มาแล้วต่อไปเครื่อง ไคลเอ็น จะต้องทำการแจ้งไปยัง Notification Server เพื่อทำการเปลี่ยนสถานะจาก Offline เป็น Online โดยใช้คำสั่งดังนี้

<<<CHG 9 NLN 268435492 \r\n

>>>CHG 9 NLN 268435492 \r\n

โดยคำสั่ง CHG เป็นคำสั่งในการเปลี่ยนสถานะของผู้ใช้งาน โดย Parameter ที่ตามมาคือรหัสสถานะที่ต้องการจะเปลี่ยนเป็น

หลังจากที่ทำการเปลี่ยนสถานะเป็น Online แล้ว Notification Server จะทำการส่งสถานะของ User Account ที่อยู่ใน Forward List ของเราและมีสถานะ Online อยู่กลับมาให้โดยใช้คำสั่งดังนี้

>>>ILN 9 NLN navy_srifa@hotmail.com Return%20To%20Zero 268435492 \r\n

โดยคำสั่ง ILN คือคำสั่งที่บอกถึงสถานะเริ่มต้นของ User Account โดย Parameter ที่ตามมาคือ User Account และ User Screen name แต่ถ้าเวลาต่อมา User มีการเปลี่ยนสถานะคำสั่งที่จะได้รับมาคือ

>>>NLN AWY natapon_pan@hotmail.com NATHAN {Display Picture} \r\n

>>>FLN natapon_p@man.com \r\n

โดยถ้าการเปลี่ยนสถานะยังเป็น Online จะต้องส่งสถานะว่า NLN ตามด้วยสถานะย่อยที่เปลี่ยนไปด้วย และ Parameter ที่ตามมาคือ User Account ของผู้ที่ทำการเปลี่ยนสถานะ

2.7 การลบ User Account

ในกรณีที่เราต้องการลบ User Account ออกจาก Forward List นั้นจะใช้คำสั่ง REM ดังนี้

<<<REM 111 FL natapon_p@msn.com \r\n

>>REM 111 FL 150 natapon_p@msn.com \r\n

โดยค่า Parameter ที่ตามมาจะเป็นประเภทของ ลีส ที่ต้องการจะลบ User Account นั้น และตามมาด้วย User Account ที่ต้องการจะลบ โดย Notification Server จะตอบกลับมาด้วยรูปแบบเหมือนกัน

2.8 การส่ง Instant Message

หาก ไคลเอ็น ต้องการจะส่ง Instant Message ไปยังผู้ใช้งานคนอื่น ๆ ที่ต้องการใน ลีส นั้น จะต้องทำการเชื่อมต่อไปยัง Switchboard Server เสียก่อนโดยใช้คำสั่ง XFR ดังนี้

<<<XFR 120 SB \r\n

>>>XFR 120 SB 207.46.108.65:1863 CKI 449169.1063097503.17917 \r\n

โดย ไคลเอ็น ต้องทำการส่งคำสั่ง XFR พร้อมด้วย Parameter SB เพื่อเป็นการบอกว่าต้องการสร้างการเชื่อมต่อไปยัง Switchboard โดยส่งคำสั่งไปยัง Notification Server ที่ ไคลเอ็น ทำการเชื่อมต่ออยู่ Notification Server จะตอบกลับมาด้วยคำสั่ง XFR พร้อมด้วย IP Address และพอตร์ ของ Switchboard Server ที่จะให้ไปเชื่อมต่อ พร้อมด้วย Hash Value ซึ่ง ไคลเอ็น จำเป็นต้องจำค่านี้ไว้ เพื่อใช้ในการยืนยันสิทธิ์ และ เรียกผู้ใช้อื่น ๆ ให้มาร่วมทำการสนทนา โดยการยืนยันสิทธิ์จะใช้คำสั่ง USR ดังนี้

<<<USR 18 nual220@hotmail.com 449169.1063097503.17917 \r\n

>>>USR 18 OK nual220@hotmail.com nual \r\n

โดย ไคลเอ็น จะทำการส่งคำสั่ง USR ไปพร้อมกับ User Account และ Hash Value ไปยัง Switchboard Server ที่ต้องการจะเชื่อมต่อ ซึ่ง Switchboard Server จะทำการตรวจสอบ Hash Value ซึ่งถ้าไม่มีปัญหาจะตอบกลับมาด้วยคำสั่ง USR OK

2.9 การเชิญผู้ใช้คนอื่นเข้ามาทำการสนทนา

หลังจากที่ทำการเชื่อมต่อเข้าไปยัง Switchboard Server ได้แล้วก็จะสามารถทำการเชิญให้ผู้ใช้คนอื่นเข้ามาทำการสื่อสารกับเราได้ โดยการเชิญนั้นจะส่งคำสั่ง CAL ไปยัง Switchboard Server โดยมีรูปแบบคำสั่งดังนี้

<<<CAL 19 natapon_pan@hotmail.com \r\n

โดยคำสั่ง CAL จะมี Parameter เดียวคือ User Account ซึ่ง Switchboard Server จะตอบกลับมาด้วยคำสั่ง CAL พร้อมด้วย Session ID ดังนี้

>>>CAL 19 Ringing 449169 \r\n

สำหรับฝั่งที่ได้รับการเชิญจะได้รับคำสั่ง RNG จาก Notification Server ของตน ดังนี้

>>>RNG 449169 207.46.108.65:1863 CKI 1063097508.7824 nual220@hotmail.com naul

โดยคำสั่ง RNG จะประกอบไปด้วย Parameter Session ID ของการเชื่อมต่อ IP Address และ พอตร์ ของ Switchboard Server ตามด้วย Hash Value ที่ใช้ในการเชื่อมต่อ และ User Account และ User Screen Name ของผู้ที่เชิญเข้าสู่การสนทนา

ซึ่ง ไคลเอ็น จะตอบกลับด้วยคำสั่ง ANS ไปยัง Switchboard Server พร้อมด้วย User Account, Hash Value และ Session ID ของการเชื่อมต่อดังนี้

<<<ANS 19 natapon_pan@hotmail.com 1063097508.7824 449169 \r\n

ซึ่งหากการเชื่อมต่อสำเร็จ Switchboard Server จะตอบกลับมาด้วยคำสั่ง IRO โดยจะตอบกลับมาตามจำนวนของ User Account ที่อยู่ใน Session ID นั้นแล้ว ดังตัวอย่าง

>>>IRO 19 1 1 nual220@hotmail.com nual \r\n

โดย Parameter แรกคือลำดับของคำสั่ง IRO ของ Transaction ID นี้ Parameter ถัดมาคือจำนวนของคำสั่ง IRO ที่จะถูกส่งมา และตามมาด้วย User Account และ User Screen Name ของผู้ที่อยู่ใน Session ID นั้น

ในขณะเดียวกันหลังจากการที่เชื่อมต่อของฝั่งตรงข้ามสำเร็จ ด้านผู้ที่เชิญเข้ามาทำการสนทนาจะได้รับคำสั่ง JOI จาก Switchboard Server ดังนี้

>>>JOI natapon_pan@hotmail.com I%20AM%20NATHAN \r\n

2.10 การส่งข้อความ Instant Message

ในการส่ง Instant Message ไปยังผู้รับที่อยู่ใน Session เดียวกับเราโดยใช้คำสั่ง MSG โดยจะส่งไปด้วยรูปแบบดังนี้

```
⊟MSN Messenger Service
    MSG 22 N 129\r\n
    MIME-Version: 1.0\r\n
    Content-Type: text/plain; charset=UTF-8\r\n
    X-MMS-IM-Format: FN=MS%20Shell%20Dlg; EF=; CO=0; CS=de; PF=0\r\n
    \r\n
    hello
```

โดยคำสั่ง MSG จะมี Parameter ที่สำคัญคือ Parameter แรกในบรรทัดที่หนึ่งซึ่งสามารถเป็นไปได้ 4 รูปแบบคือ N,U,D และ A ซึ่งมีความหมายดังนี้

U คือ Switchboard จะไม่ทำการตรวจสอบว่า Message นี้ได้รับหรือไม่

N คือ Switchboard จะทำการตรวจสอบและแจ้งกลับมาบอกว่า Message นี้ไม่ได้รับ

D คือ MSG นี้เป็นการเชิญให้เข้าร่วม Application ต่าง ๆ

A คือ Switchboard จะทำการตรวจสอบและแจ้งกลับมาบอกว่า Message นี้ได้รับแล้ว บรรทัดที่สองและสามจะเป็นรายละเอียดของ MIME Header บรรทัดที่สี่เป็นบรรทัดที่สำคัญอีกบรรทัดหนึ่งเป็นบรรทัดที่บอกถึงรายละเอียดเกี่ยวกับรูปแบบของตัวอักษรโดยแต่ละ Field จะมีความหมายดังนี้

FN คือ URL Encode ของชื่อ Font

EF คือ รูปแบบของตัวอักษร (Bold,Italic,Strikeout,Underline) ซึ่งหากเป็นรูปแบบผสมก็จะทำการส่งค่าเป็น รหัสต่อกันเช่น ถ้าต้องการตัวหนาและเอียง ค่าใน Field นี้จึงเป็น BI

CO คือ รหัสเลขฐานสิบหกของสีของตัวอักษร

CS คือ Character Set

PF นั้นไม่ได้มีคำอธิบายอย่างแน่ชัด

หลังจากหมดส่วนของ MIME แล้วจะทำการขึ้นบรรทัดใหม่สองครั้งและจะตามด้วยข้อความที่ต้องการจะส่งในกรณีนี้คือ hello

>>>IRO 91 1 2 natapon_p@msn.com I%20AM%20NATHAN \r\n

>>>IRO 91 2 2 nual220@hotmail.com nual \r\n

ซึ่งจะทำให้ผู้ที่ได้รับคำสั่ง IRO นี้ทราบว่ามีผู้ใช้งานคนใดบ้างที่เข้ามาทำการสนทนาใน Session นี้แล้วบ้าง

สำหรับการส่งข้อความที่ต้องการสนทนาก็จะใช้คำสั่ง MSG เช่นเดียวกับการส่งข้อความธรรมดา โดย Switchboard Server จะทำหน้าที่ในการส่งข้อความไปยังทุก ๆ คนที่อยู่ใน Session เดียวกับผู้ที่ส่งข้อความนั้น ซึ่งจะเห็นว่าไม่ได้มีความยุ่งยากซับซ้อนเลย

ในการส่งข้อความทั้งที่เป็นแบบหนึ่งต่อหนึ่ง หรือ หลาย ๆ คนนั้น หากระหว่างการส่งข้อความ ทางฝั่งผู้รับมีปัญหาทางด้านครือข่ายไม่สามารถรับข้อความได้ Switchboard Server จะพยายามส่งข้อความไปเรื่อย ๆ จนถึง ณ เวลาหนึ่ง Switchboard Server อาจจะทำการ Challenge เพื่อตรวจสอบสถานะ การทำงานของ ไคลเอ็น หรือ อาจจะถือว่าเครื่อง ไคลเอ็น ไม่ทำงานอยู่แล้วเลยก็ได้ และจะลบข้อความที่พยายามจะส่งนั้นทิ้งไป

บทที่ 3

IPTABLES

ลีนุกซ์สามารถใช้งานเป็นไฟร์วอลล์ได้ตั้งแต่เคอร์เนล 1.1 ซึ่งเป็นเวอร์ชันแรก โดย AlanCox ใช้ชื่อว่า ไอพีเอฟดับเบิลยู ต่อมา ลีนุกซ์ 2.0 ได้ถูกพัฒนาและปรับปรุง โดยอนุญาตให้ผู้ใช้สามารถควบคุมฟิวเตอร์ ได้ และต่อมา ลีนุกซ์ 2.2 ก็ได้สร้างเครื่องมือตัวใหม่ชื่อ ไอพีเชน ซึ่งเผยแพร่ในปี 1998 โดย Rusty Russel และทีมงาน ทั้งนี้ ไอพีเชน นี้ถือได้ว่าเป็นพัฒนาการขั้นที่สามของ ลีนุกซ์ไฟร์วอล จวบจนกระทั่งในปัจจุบัน ก็มี เน็ตฟิวเตอร์ และไอพีเทเบิล ซึ่งถือว่าเป็นพัฒนาการขั้นที่สี่ของ ลีนุกซ์ไฟร์วอล ใน เคอร์เนล 2.4

3.1 การใช้งานไอพีเทเบิล เบื้องต้น

มีรูปแบบการใช้งานดังนี้คือ

iptables [table] <command> <match> <target/jump>

โดย กฎ ที่เขียนขึ้นจะเป็นเป็นตัวบอกเคอร์เนลว่าให้กระทำ action อย่างไร ในกรณีที่พบ แพ็คเกจตรงตามที่ระบุไว้

- **[table]** หมายถึง ตารางที่ต้องการระบุ เช่น iptables -t nat หมายถึงให้ทำงานกับ ตารางเน็ท ในกรณีที่ไม่ได้ระบุตาราง จะถือว่าคำสั่งดังกล่าวระบุถึง ตารางฟิวเตอร์ โดยอัตโนมัติ
- **<command>** จะเป็นตัวสั่งให้ไอพีเทเบิล ทำในสิ่งที่ต้องการ เช่น iptables -A INPUT ซึ่งหมายถึงให้สร้าง กฎ ต่อท้าย INPUT chain ในตารางฟิวเตอร์
- **<match>** เป็นส่วนที่ใช้ตรวจสอบว่า แพ็คเกจ มีข้อมูลตรง (match) กับที่ระบุไว้หรือไม่ เช่น มี source ip address เป็น 1.2.3.4
- **<target/jump>** เป็นตัวระบุว่าเมื่อเจอ แพ็คเกจ ที่ ตรงกับกฎ ก็จะกระทำ (action) ตามที่ระบุไว้ เช่น ถ้า แพ็คเกจ ใดมี source ip address เป็น 1.2.3.4 ให้ ทิ้ง แพ็คเกจ นั้นไป

Table

ไอพีเทเบิล สามารถทำงานได้กับตาราง(table) 3 ตารางหลัก สามารถระบุตารางได้โดยใช้ออปชัน -t ตามด้วยชื่อ ตาราง คือ

1. **Filter table** ใช้สำหรับกรอง แพ็คเกจ มี 3 built-in chain คือ INPUT, OUTPUT, FORWARD
2. **Nat table** ใช้สำหรับการแปลงแอดเดรส (Network Address Translation) มี 3 built-in chain คือ PREROUTING, POSTROUTING, OUTPUT
3. **Mangle table** เป็นตารางที่ใช้เปลี่ยนแปลงหรือแก้ไข แพ็คเกจ เช่น เปลี่ยนค่า TTL, MARK ซึ่งปกติจะใช้ในการทำ routing ที่มีความซับซ้อนสูง มี 2 built-in chain คือ PREROUTING chain (ใช้แก้ไข แพ็คเกจ ก่อนที่จะเข้าสู่ไฟร์วอลล์และก่อนเข้าสู่ routing decision) และ OUTPUT chain (ใช้แก้ไข แพ็คเกจ ที่ถูกสร้างโดยไฟร์วอลล์ก่อนที่มันจะถูกส่งไปยัง routing decision) ทั้งนี้ไม่สามารถทำ network address translation หรือ masquerading ที่ table นี้ได้

Command

- **-A** เพิ่ม กฎ ใหม่ต่อท้าย chain (Append กฎ) เช่น
  # iptables -A INPUT -p ALL -i eth0 -j ACCEPT
- **-D** ลบ กฎ (Delete กฎ) เช่น
  # iptables -D INPUT --dport 80 -j DROP
- **-I** เพิ่ม กฎ ใหม่ ใน chain (Insert กฎ) เช่น
  # iptables -I OUTPUT -p ALL -s 127.0.0.1/32 -j ACCEPT
- **-R** แทนที่ กฎ เดิม ด้วย กฎ ใหม่ (Replace กฎ)
- **-L** แสดง กฎ ทั้งหมดใน chain (ถ้าไม่ระบุ chain จะแสดง กฎ ทั้งหมดใน filter table ทั้งสาม built-in chain) เช่น
  # iptables -L
  # iptables -L -t nat
  # iptables -L INPUT

- **-F** ลบ กฎ ทั้งหมดใน chain ทิ้ง เช่น

  # iptables -F INPUT

  # iptables -F mychain

- **-Z** ใช้ reset byte counter สำหรับทุก กฎ ใน chain ที่กำหนด เช่น

  # iptables -Z INPUT

- **-N** ใช้สร้าง chain ใหม่ เช่น

  # iptables -N mychain

- **-X** ลบ chain ที่ไม่มี กฎ ซึ่งสามารถลบ user-defined chain ที่ไม่มี กฎ ได้ แต่ไม่สามารถลบ built-in chain ได้ เช่น

  # iptables -X emptychain

- **-P** เปลี่ยน default policy ของ chain ค่าที่ใช้ได้คือ ACCEPT, DROP ทั้งนี้ค่านี้มีความสำคัญอย่างมากเพราะหาก แพ็คเกจ ถูกส่งเข้ามาใน chain แล้ว และไม่ ตรง กับ กฎ ใดๆ เลย แพ็คเกจ นั้นก็ต้องถูกตัดสินใจโดย policy ของ chain นั้นๆ เช่น

  # iptables -P FORWARD DROP

  ซึ่งหาก แพ็คเกจ ถูกส่งเข้ามายัง FORWARD chain และไม่ ตรง กับ กฎ ใดๆ ใน FORWARD chain นี้เลย มันก็จะถูก DROP ทันที

- **-E** ใช้เปลี่ยนชื่อ chain ใหม่ เช่น

  # iptables -E myoldchain mynewchain

การใช้ command ด้านบนนั้นสามารถใช้ร่วมกับออปชันบางอย่างได้ คือ

- **-V, --verbose** ใช้ร่วมกับ -L, -A, -I, -D, -R เพื่อให้แสดงจำนวน byte ที่ ตรง กับ กฎ ออกมาด้วย (หน่วยเป็นได้ทั้ง K(x1,000),M(x1,000,000),G(x1,000,000,000)) เช่น

  # iptables -L -v

- **-x, --exact** ใช้ร่วมกับ -L และ -v เพื่อให้แสดงจำนวน แพ็คเกจ และจำนวนของ byte ข้อมูลที่ ตรงกับกฎ โดยไม่ให้แสดงผลในหน่วยของ K,M,G เช่น

  # iptables -L OUTPUT -v -x

- **-n, --numeric** ใช้ร่วมกับ -L เพื่อสั่งให้ iptables แสดงข้อมูลไอพีแอดเดรสและ port เป็นตัวเลขเท่านั้น เช่น

  # iptables -L OUTPUT -n

- **--line-numbers** ใช้ร่วมกับ -L เพื่อแสดงเลขบรรทัดของ กฎ ซึ่งตัวเลขที่แสดงนี้จะสามารถใช้ได้กับคำสั่ง insert กฎ ที่ระบุเป็นลำดับที่ของ กฎ เช่น

  # iptables -L --line-numbers

- **--modprobe=***command* เพื่อโหลด module ที่เกี่ยวข้อง

Match

การตั้งเงื่อนไขของการ ตรงกับกฎ นั้นจะต้องอาศัยความเข้าใจในเรื่อง IP, TCP, UDP, และ ICMP มาบ้างพอสมควร จึงจะสามารถตั้งเงื่อนไขที่เหมาะสมและตรงตามความต้องการได้ ซึ่งมีรายละเอียดดังนี้

- **การระบุ source, destination IP address**

  สามารถระบุ source ip address ของ แพ็คเกจ โดยใช้ **-s** หรือ **--source** หรือ **--src** และสำหรับ destination ip address ก็ใช้ **-d** หรือ **--destination** หรือ **--dst** การระบุไอพีแอดเดรสนั้นสามารถทำได้ 4 แบบด้วยกันคือ

  1.ใช้ชื่อเต็มแทน เช่น localhost หรือ www.nectec.or.th

  2. ระบุไอพีแอดเดรสโดยตรง เช่น 127.0.0.1 หรือ 202.44.204.33

  3. ระบุเป็น group ของไอพีแอดเดรส เช่น 202.44.204.0/24 ซึ่งหมายถึงไอพีแอดเดรสตั้งแต่ 202.44.204.0 - 202.44.204.255

  4. หรืออาจจะใช้ 202.44.204.0/255.255.255.0 แทน 202.44.204.0/24 ได้

- **การทำ Inversion**

  ในบางกรณีนั้นหากต้องการระบุเป็น inverse เช่น อนุญาตให้ทุกไอพียกเว้นไอพีที่ระบุไว้ ซึ่งการใช้คำสั่งดังกล่าวสามารถทำได้โดยใช้เครื่องหมาย ! นำหน้า argument ที่ต้องการ (เครื่องหมาย ! หมายถึง NOT) เช่น -p ! TCP ซึ่งจะ ตรง กับโพรโตคอลทุกๆ ตัวที่ไม่ใช่

TCP หรือ -s ! localhost ซึ่งหมายถึง แพ็คเกจ ที่มี source ip address อื่นๆ ยกเว้น localhost (127.0.0.1)

- **การระบุโพรโตคอล**

  สามารถระบุโพรโตคอลที่ต้องการได้ดังนี้คือ TCP, UDP, ICMP หรือสามารถใช้ตัวเลขแทนได้ (สำหรับ *NIX อ้างอิงได้จาก /etc/protocols) และยังสามารถใช้ได้ทั้งตัวอักษรเล็กหรือใหญ่ (ใช้ได้ทั้ง tcp และ TCP) เช่น **-p TCP** หรือ **-p ! tcp**

  - **การระบุ interface**

    -i หรือ **--in-interface** ตามด้วยชื่อ interface ใช้เพื่อระบุ incoming interface ซึ่งหมายถึงว่า แพ็คเกจ ที่จะ ตรง กับ กฎ นี้ต้องเข้ามาจาก interface ที่กำหนด เช่น -i eth0 หมายความว่า ทุก แพ็คเกจ ที่เข้ามาทาง eth0 จะ ตรง กับ กฎ นี้ ทั้งนี้ชื่อ interface ที่สามารถใช้ได้นั้น สามารถตรวจสอบได้โดยใช้คำสั่ง ifconfig และ **-o** หรือ **--out-interface** ตามด้วยชื่อของ interface ใช้เพื่อระบุ outgoing interface ซึ่งหมายถึงว่า แพ็คเกจ ที่จะ ตรง กับ กฎ นี้ กำลังจะเดินทางผ่าน interface ที่ระบุไว้ เช่น -o eth1 หรือ -o ! eth1

- **fragment packet**

  ในการส่งข้อมูลใน ip network นั้นเป็นเรื่องปกติที่จะเกิดการ fragment ของ แพ็คเกจ เนื่องจากขนาดของ แพ็คเกจ มีขนาดใหญ่เกินไปที่จะส่งไปในครั้งเดียว จำเป็นต้องมีการแบ่ง แพ็คเกจ ออกเป็นหลายๆ ชิ้นทยอยส่งไป ซึ่งเรียกกันว่าการทำ fragment โดยเครื่องปลายทางจะทำหน้าที่ประกอบ fragment packet รวมกันเป็น แพ็คเกจ ที่สมบูรณ์ดังเดิม ข้อมูลที่เป็น fragment packet นั้นจะมี header ที่สมบูรณ์แค่ แพ็คเกจ แรกเท่านั้น ตัว แพ็คเกจ ที่ตามมาจะมีแค่ header บางส่วนคือ ไอพีแอดเดรสเท่านั้น ไม่มีข้อมูลของโพรโตคอลแนบมาด้วย ดังนั้นการตรวจสอบข้อมูล header ของ TCP, UDP, ICMP จึงไม่สามารถทำได้ใน แพ็คเกจ ที่สองเป็นต้นมาหากใช้ NAT บรรดา fragment packet จะถูกประกอบเข้าด้วยกันจนสมบูรณ์ก่อนที่ แพ็คเกจ จะเข้าไปถึง packet filtering ดังนั้นจึงไม่มีความจำเป็นที่จะต้องกังวลเกี่ยวกับ fragment packet ดังนั้นถ้าไม่ได้ใช้ NAT ก็ควรทำความเข้าใจไว้ว่า iptables มีกระบวนการในการทำงานกับ fragment packet อย่างไร หลังจากที่ fragment packet แรกผ่านเข้ามาแล้ว iptables สามารถตรวจสอบได้ว่าจะอนุญาตให้ผ่าน

หรือไม่ ในขณะที่ fragment packet ที่สองและหลังจากนั้นที่ตามมานั้น จะไม่สามารถ ตรงกับ กฎ ใดๆ เลย เช่น -p TCP --sport www หรือแม้แต่ -p TCP --sport ! www
อย่างไรก็ตาม สามารถเขียน กฎ ให้ตรวจสอบทั้ง fragment packet ตัวที่สองและหลังจากนั้นที่ตามมาได้ด้วยการใช้ **-f** หรือ **--fragment** ทั้งนี้อาจจะเขียนในทางตรงข้ามคือไม่ต้องตรวจสอบ fragment packet ที่สองและหลังจากนั้นโดยใช้ **! -f** ก็ได้
ทั้งนี้โดยปกติแล้วมักจะปล่อยให้ fragment packet ผ่านไป เนื่องจากถ้าสามารถ DROP ตัว fragment packet ตัวแรกได้แล้ว มันก็ไม่สามารถถูกประกอบที่เครื่องปลายทางได้ แต่ทั้งนี้ fragment packet ที่ถูกปล่อยไปดังกล่าวอาจจะทำให้เครื่องที่ได้รับ hang หรือ crash ได้ หรืออาจจะเกิดการโจมตีแบบ Denial of Service โดยใช้ fragment packet ได้

- **TCP extension**
  ถ้ามีการเรียกใช้ -p tcp ตัว TCP extension ก็จะถูกโหลดมาใช้งานโดยอัตโนมัติ โดยมีออปชันให้เลือกใช้งานดังนี้
  - **--tcp-flags** *mask flags* : mask นั้นหมายถึง flag ที่ต้องการตรวจสอบ และ flag เป็นตัวที่บ่งชี้ว่า flag ใดต้องถูก set บ้าง
    เช่น **# iptables -A INPUT -p tcp --tcp-flags ALL SYN,ACK -j DROP**
    โดย ALL นั้นหมายถึงทุกๆ flag (SYN,ACK,FIN,RST,URG,PSH) และถ้า flag SYN,ACK ถูก set พร้อมกันก็ให้ drop แพ็คเกจ นั้นทิ้งไป นอกจากนี้ยังสามารถใช้ NONE ซึ่งหมายถึงไม่มี flag ใดถูก set ได้
  - **--syn** เป็นสัญลักษณ์ย่อของ --tcp-flags SYN,RST,ACK SYN
  - **--source-port** หรือ **--sport** สามารถใช้ได้ทั้งตัวเลขและตัวอักษร (อ้างอิงจากไฟล์ /etc/services) และระบุเป็น port เดี่ยว หรือช่วงของ port ได้
    เช่น --sport 21:25 หมายถึง port 21 - 25 , --sport 25: หมายถึง port ที่มากกว่าหรือเท่ากับ 25 , --sport :25 หมายถึง port ที่น้อยกว่าหรือเท่ากับ 25
  - **--destination-port** หรือ **--dport** มีรูปแบบการใช้งานเช่นเดียวกันกับ --sport
  - **--tcp-option** ใช้ตรวจสอบ TCP option ว่าตรงกับเลขที่ระบุไว้หรือไม่
- **อธิบาย flag ของ TCP เพิ่มเติม**
  การเชื่อมต่อโดยใช้ TCP นั้น ผู้ที่เริ่มสร้าง connection จะเป็นผู้ส่ง SYN แพ็คเกจ มายังเครื่องปลายทาง ดังนั้นหากไม่ต้องการให้ให้เครื่องใดเป็นผู้เริ่มสร้างการติดต่อก็สามารถ block ไอพีดังกล่าวได้ โดยใช้ --syn เช่น -p TCP -s x.x.x.x --syn หากยังไม่เข้าใจรูปแบบ

การเชื่อมต่อแบบ TCP นี้แล้ว ก็เป็นการยากที่จะสร้าง กฎ สำหรับ iptables ดังนั้นจึงขอแนะนำให้ไปศึกษาหลักการทำงานเบื้องต้นของทั้ง TCP, UDP, ICMP มาก่อน

- **UDP extension**

  คล้ายกันกับ TCP ตัว UDP extension มีออปชันให้เลือกใช้เพียงแค่ 2 อย่างเท่านั้นคือ **--source-port (--sport)** และ **--destination-port (--dport)** โดยต้องระบุ **-p udp** ด้วย

- **ICMP extension**

  โดยการระบุ **-p icmp** ก็สามารถใช้งาน ICMP extension ได้ โดยมีออปชันให้เลือกคือ **--icmp-type** เช่น --icmp-type host-unreachable (หรือใช้เลข 3 แทนได้) นอกจากนี้ยังสามารถระบุ type/code ได้ เช่น 3/3 ซึ่งหมายถึง port unreachable

- **Match Extension**

  เป็น netfilter package ที่อยู่ในช่วงทดลองใช้ รูปแบบการใช้งานให้ใช้ -m แล้วตามด้วย กฎที่ต้องการ เช่น -m mac ทั้งนี้มีออปชันให้เลือกใช้งานดังต่อไปนี้

  - **mac**

    รูปแบบการใช้งาน: **-m mac** หรือ **--match mac**

    ใช้ตรวจสอบ source MAC address ว่าตรงกับค่าที่ระบุไว้หรือไม่ มีประโยชน์สำหรับ PREROUTING, INPUT chain โดยมีออปชันให้ใช้งานคือ

    - **--mac-source** เช่น --mac-source 00:55:81:CC:42:FF

  - **limit**

    รูปแบบการใช้งาน: **-m limit** หรือ **--match limit**

    ใช้เพื่อจำกัดจำนวนของการ ตรงกับกฎ ที่อาจจะมากเกินไป เป็นประโยชน์สำหรับกฎ ที่วางไว้ตอนท้ายสุดของ chain (ใช้ร่วมกันกับ DROP policy) ซึ่งส่วนใหญ่เป็นกฎ ที่ใช้เก็บข้อมูลลงล็อกไฟล์ ซึ่งถ้าผู้บุกรุกส่ง แพ็คเกจ ที่ไม่เข้าข่าย กฎ ใดๆ ใน chain จนกระทั่งมาถึง กฎ ที่ทำหน้าที่เก็บล็อกนี้ ถ้า แพ็คเกจ ที่เข้ามามีจำนวนมากก็อาจจะทำให้ฮาร์ดดิสก์เต็มได้ ดังนั้นจึงต้องใช้จำกัดจำนวนในการเก็บข้อมูลลงล็อก

  - **owner**

    รูปแบบการใช้งาน: **-m owner** หรือ **--match owner**

    ใช้ตรวจสอบลักษณะของ แพ็คเกจ ว่าใครเป็นผู้สร้าง ซึ่งสามารถใช้ได้กับ

OUTPUT chain เท่านั้น และใช้ได้กับบาง แพ็คเกจ ที่มีเจ้าของ เช่น ICMP แพ็คเกจ นั้นใช้ไม่ได้เพราะไม่มีเจ้าของ มีออปชันให้ใช้งานดังนี้คือ

- **--uid-owner** *userid*
  ใช้ตรวจสอบว่า แพ็คเกจ ถูกสร้างโดย user id ที่ระบุไว้หรือไม่ (ใช้ตัวเลขแทน userid เท่านั้น)
- **--gid-owner** *groupid*
  ใช้ตรวจสอบว่า แพ็คเกจ ถูกสร้างโดย user ที่อยู่ใน group id ที่ระบุไว้หรือไม่ (ใช้ตัวเลขแทน groupid เท่านั้น)
- **--pid-owner** *processid*
  ใช้ตรวจสอบว่า แพ็คเกจ ถูกสร้างขึ้นจาก process ที่มี process id ตรงกับที่ระบุไว้หรือไม่
- **--sid-owner** *sessionid*
  ใช้ตรวจสอบว่า แพ็คเกจ ถูกสร้างโดย process ที่อยู่ใน session group ที่กำหนดไว้หรือไม่

- **unclean**

รูปแบบการใช้งาน: **-m unclean** หรือ **--match unclean**
เป็นโมดูลที่อยู่ในระหว่างการทดลองใช้งาน นอกจากนี้ยังไม่มีออปชันสำหรับใช้งาน และโปรดระมัดระวังหากจะนำไปใช้กับเครื่องที่ต้องการความปลอดภัย เนื่องจากอาจจะยังมีข้อบกพร่องของโปรแกรมอยู่

โดย แพ็คเกจ ที่เข้าข่าย unclean คือ

- แพ็คเกจ ที่มี header ของ ICMP/TCP/UDP สั้นหรือไม่สมบูรณ์
- TCP, UDP แพ็คเกจ ที่มี source หรือ destination ip address เป็นศูนย์
- TCP แพ็คเกจ ที่ใช้ flag ผสมกันแบบผิดปกติ
- แพ็คเกจ ที่ใช้ TCP option, IP option เกินความยาวที่กำหนดไว้ หรือมีความยาวของออปชันเป็นศูนย์
- fragment packet ที่ไม่สมบูรณ์ ทั้งด้านความยาวและค่า offset ที่เหลื่อมซ้อนกัน เช่น Ping of Death

- **multiport** ใช้ร่วมกับ --sport หรือ --dport ในกรณีที่ต้องการระบุ port จำนวนมากกว่าหนึ่ง เช่น

  -m multiport -p tcp --sport 25,80,53

- **The State Match**

  รูปแบบการใช้งาน: **-m state** หรือ **--match state**

  เป็นโมดูลที่ใช้ประโยชน์ได้เป็นอย่างดี มีออปชันให้ใช้งานดังนี้

  - **NEW**

    รูปแบบการใช้งาน: **-m state --state new** หรือ **--match state --state new**

    หมายถึง แพ็คเกจ ที่เป็นตัวสร้าง connection ใหม่

  - **ESTABLISHED**

    รูปแบบการใช้งาน: **-m state --state established** หรือ **--match state --state established**

    หมายถึง แพ็คเกจ ที่เกี่ยวข้องกันกับ connection ที่สร้างไว้แล้ว เช่น echo-reply packet หรือ แพ็คเกจ ที่ส่งข้อมูลออกไปจาก web server เมื่อมี request web service เข้ามา

  - **RELATED**

    รูปแบบการใช้งาน: **-m state --state related** หรือ **--match state --state related**

    เป็น แพ็คเกจ ที่เกี่ยวข้องกับ connection ที่สร้างไว้แล้ว แต่ไม่ใช่ส่วนหนึ่งส่วนใดของ connection นั้น เช่น FTP data แพ็คเกจ (port 20) ที่เกิดขึ้นจากการใช้คำสั่งใน FTP command (port 21)

  - **INVALID**

    รูปแบบการใช้งาน: **-m state --state invalid** หรือ **--match state --state invalid**

    เป็น แพ็คเกจ ที่ไม่เกี่ยวข้องกับส่วนอื่นเลย เช่น icmp echo-reply ที่เกิดขึ้น โดยที่ไม่มีเครื่องได้ในระบบส่ง echo-request ออกไปเลย (กรณีเช่นนี้เกิดขึ้นได้เนื่องจากอาจจะโดนโจมตีแบบ Smurf attack)

การระบุ target

เมื่อมี แพ็คเกจ ที่ ตรง กับ กฎ แล้ว ต้องกำหนด target สำหรับ แพ็คเกจ ไว้ด้วย โดยปกติจะใช้กัน 2 target คือ DROP และ ACCEPT นอกจากนี้ยังมี target แบบอื่นได้คือ

- **user-defined chain**

  เนื่องจาก iptables อนุญาตให้ผู้ใช้สามารถสร้าง chain ขึ้นมาได้ใหม่นอกเหนือจาก built-in chain ทั้งสามตัว (INPUT, OUTPUT, FORWARD) ทั้งนี้จะต้องใช้ตัวอักษรตัวเล็กทั้งหมดสำหรับ chain ที่ผู้ใช้สร้างขึ้นเอง

  เมื่อ packet ตรง กับ กฎ ที่เป็น user-defined chain ตัว แพ็คเกจ จะถูกนำไปตรวจสอบใหม่โดย user-defined chain นั้นๆ และถ้าใน chain นั้นๆ ไม่มีการตัดสินใจใดๆ ตัว แพ็คเกจ ก็สามารถย้อนกลับมายัง กฎ ถัดไปใน chain ที่เริ่มต้นเดินทางได้ (ศึกษารายละเอียดได้จากตัวอย่างด้านล่าง)

INPUT
Rule1: -p ICMP -j DROP
Rule2: -p TCP -j test
Rule3: -p UDP -j DROP

test
Rule1: -s 192.168.1.1
Rule2: -d 192.168.1.1

  เช่น ถ้า TCP แพ็คเกจ เดินทางจาก 192.168.1.1 ไปยัง 1.2.3.4 ดังนั้น แพ็คเกจ จะเข้าสู่ INPUT chain และไม่ ตรง กับ กฎ1 แต่ ตรง กับ กฎ2 ซึ่งมี target เป็น test ดังนั้น แพ็คเกจ จะเข้าสู่ test chain และ ตรง กับ กฎ1 แต่เนื่องจาก กฎ1 ของ test ไม่ได้ระบุ target ดังนั้น แพ็คเกจ จึงผ่านไปยัง กฎ2 ซึ่งไม่ ตรง จากนั้น แพ็คเกจ จึงจะเดินทางกลับไปยัง กฎ3 ของ INPUT chain อีกครั้ง ซึ่งก็ไม่ ตรง เช่นกัน ในกรณีที่ผ่าน กฎ ทั้งหมดแล้วแต่ไม่ ตรง หรือ ตรง แต่ไม่มี target นั้น แพ็คเกจ จะถูก DROP หรือ ACCEPT ก็ขึ้นอยู่กับ default policy ของ chain นั้นๆ ซึ่งสามารถตั้งค่าได้ง่ายๆ

  เช่น # iptables -P INPUT DROP หรือ # iptables -P FORWARD ACCEPT

- **new target**

  เป็น target ที่สร้างเพิ่มเติมขึ้นมาคือ

  - **LOG**

    เป็นโมดูลที่มีความสามารถในการเก็บข้อมูลลงล็อก (มี syslog facility เป็น kernel) สำหรับ แพ็คเกจ ที่ ตรง กับ กฎ ที่ระบุ target เป็น LOG มีออปชันให้เลือกใช้งานดังนี้คือ

- **--log-level**
  เป็นการระบุ priority level ของ log ซึ่งกำหนดได้ตั้งแต่ debug, info ,notice, warning, crit, alert,
- **--log-prefix**
  ตามด้วยชุดของตัวอักษรยาวไม่เกิน 29 ตัว ซึ่งชุดของตัวอักษรดังกล่าวจะปรากฏอยู่บนล็อกไฟล์

- **REJECT**
  คล้ายกับ DROP เพียงแต่จะส่ง ICMP port unreachable กลับไปยังผู้ที่ส่ง แพ็คเกจมา (ข้อยกเว้นคือ ICMP error message ไม่ response กับ ICMP error message ด้วยกันเอง เพราะอาจจะทำให้เกิดลูปที่ไม่รู้จบ) ทั้งนี้สามารถใช้ร่วมกับ --reject-with ตามด้วย argument ที่ต้องการได้ รายละเอียดโปรดศึกษาจากคู่มือการใช้งาน iptables ที่มาพร้อมตัวโปรแกรม (#man iptables)

- **special built-in target**
  - **RETURN**
    กรณีที่ แพ็คเกจ ตรง กับ กฎ ที่มี target เป็น RETURN นั้นเสมือนกับเป็นคำสั่งให้ออกไปจาก chain ปัจจุบัน เช่น หาก ตรง กับ กฎ ที่อยู่ใน built-in chain (INPUT, FORWARD, OUTPUT) แพ็คเกจ ดังกล่าวจะถูกโยนไปยัง default policy ของ chain นั้นๆ และหาก แพ็คเกจ ตรง กับ กฎ ที่เป็น user-defined chain ตัว แพ็คเกจจะถูกโยนออกมา chain ก่อนหน้านั้น

  - **QUEUE**
    เป็น chain พิเศษ ใช้สำหรับส่งต่อ แพ็คเกจ ไปยัง application ที่เขียนขึ้นมารองรับโดยเฉพาะ โดยจะต้องมี queue handler และ application เป็นส่วนประกอบที่จะทำงานร่วมกัน

3.2 การเดินทางของ แพ็คเกจ

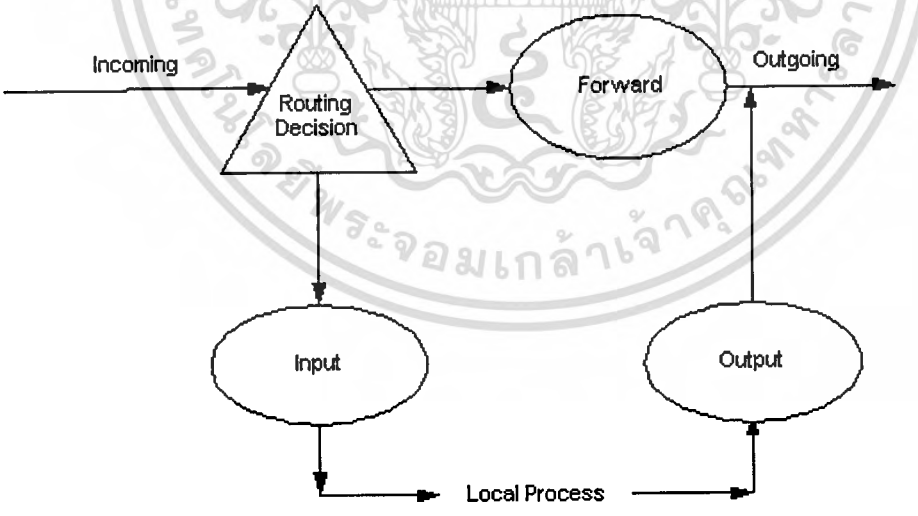
เมื่อ แพ็คเกจ เข้ามาถึงไฟร์วอลล์ มันจะผ่านฮาร์ดแวร์เข้ามายัง device ที่เหมาะสมในเคอร์เนล จากนั้น แพ็คเกจ จะเดินทางไปเป็นทอดๆ ก่อนที่จะถูกส่งไปยังปลายทางที่แท้จริง เช่น แอพลิเคชัน

ในเครื่องไฟร์วอลล์ หรือ forward ต่อไปยังเครื่องอื่น ซึ่งจะยกตัวอย่างเพื่อให้เห็นภาพอย่างชัดเจน ดังนี้

Filter Table

เป็นตารางที่ใช้งานมากที่สุด เป็นจุดที่ใช้ในการตรวจสอบและควบคุมการผ่านเข้าออกของแพ็คเกจ ถ้าหากจะพิจารณาการไหลเวียนของ แพ็คเกจ เฉพาะในส่วนของ filter table โดยไม่สนใจ table อื่นๆ นั้น ก็พอจะแสดงให้เห็นได้ดังภาพที่ 3.1 โดยเมื่อ แพ็คเกจ เข้ามาในระบบ มันจะเข้าไปยัง routing decision เพื่อตัดสินใจว่า แพ็คเกจ จะถูกส่งไปที่ใด

- ในกรณีที่ แพ็คเกจ ถูกส่งผ่านไปยังเครื่องอื่น แพ็คเกจ นั้นจะต้องถูกตรวจสอบโดย กฎ ใน FORWARD chain
- ถ้า แพ็คเกจ นั้น มีเป้าหมายเป็นเครื่องปัจจุบัน (เครื่องที่รัน iptables อยู่นี้ เรียกอีกอย่างว่า linux box) ตัว แพ็คเกจ จะถูกตรวจสอบโดย กฎ ใน INPUT chain
- และในกรณีที่ แพ็คเกจ ถูกสร้างจากเครื่องปัจจุบัน (linux box) ตัว แพ็คเกจ จะถูกตรวจสอบจาก กฎ ใน OUTPUT chain ก่อนที่จะถูกส่งออกไป



รูปที่ 3.1 แสดงให้เห็นว่า แพ็คเกจ มีเส้นทางการเดินทางอย่างไรเมื่อเข้ามาในระบบ (filter table)

ดังภาพ iptables ประกอบไปด้วย built-in chain จำนวน 3 chain ซึ่งไม่สามารถลบได้คือ INPUT, OUTPUT, FORWARD เมื่อเครื่องคอมพิวเตอร์เริ่มทำงานในครั้งแรก ทั้งสาม chain จะมี default policy เป็น ACCEPT ซึ่งหมายความว่าอนุญาตให้ทุกอย่างผ่านเข้าออกได้หมด และสำหรับ FORWARD chain นั้น ถึงแม้จะได้กำหนดให้ policy เป็น ACCEPT แล้ว แพ็คเกจ ก็จะยังไม่

สามารถถูก forward ไปยังจุดหมายที่ต้องการได้ ตราบใดที่ยังไม่ได้เซ็ตให้ enable IP forwarding ทั้งนี้โดย default แล้ว forward=0 สามารถกำหนดให้ enable IP forwarding (forward=1) ได้ โดย

ใช้คำสั่ง **echo "1" > /proc/sys/net/ip_forward** เพื่อกำหนดให้ IP forwarding เป็น enable เพื่อให้ Linux box สามารถ forward ip packet ได้ ในบางครั้งนั้นการใช้คำสั่งดังกล่าวทุกครั้งอาจจะไม่สะดวก สามารถแก้ไขไฟล์ configuration ที่ **/etc/sysctl.conf** แล้ว set ให้ **net.ipv4.ip_forward=1** เพื่อเป็นการแก้ไขแบบถาวร

ในกรณีที่ต้องการให้สนับสนุนการทำงานกับ dynamic IP ด้วย เช่น PPP, SLIP, DHCP ก็สามารถทำได้โดยใช้คำสั่ง **echo "1" > /proc/sys/net/ipv4/ip_dynaddr** ได้เช่นเดียวกัน

Mangle Table

เป็นตารางที่ใช้สำหรับแก้ไขข้อมูล TOS, TTL, MARK ของ แพ็คเกจ ซึ่งโดยปกติแล้วแทบจะไม่ได้ใช้งาน และไม่ควรทำ packet filtering หรือกรอง แพ็คเกจ ที่ตารางนี้ รวมทั้งไม่ควรทำ DNAT, SNAT หรือ Masquerading ที่ตารางนี้อย่างเด็ดขาดด้วย

Nat Table

เป็นตารางที่ใช้สำหรับทำ network address translation เช่น เปลี่ยนค่า source ip address, destination ip address จุดสำคัญอีกอย่างหนึ่งที่ต้องรู้ก็คือ มีเพียง แพ็คเกจ แรกเท่านั้นที่เข้ามาที่ chain นี้ ส่วน แพ็คเกจ ถัดไปนั้นจะถูกกระทำเหมือนที่ แพ็คเกจ แรกได้รับ ดังนั้นจึงไม่ควรทำ packet filtering ที่ chain เหล่านี้ การใช้งาน Nat table นั้นก็เพียงแต่ใช้ออปชัน -t nat เท่านั้น และ target ที่สามารถใช้งานได้คือ SNAT, DNAT, Masquerade, Redirect

บทที่ 4

L7-Filter

L7-filter คือตัวตรวจจับและจำแนกแพคเกจในชั้นแอพพลิเคชันเลเยอร์ (layer 7)สำหรับเคลเนลลีนุกซ์ซึ่งสามารถจำแนกแพ็คเกจของโปรโตคอลในชั้นแอพพลิเคชันได้แก่ HTTP, Bittorrent,edonkey, FTP ,Kazaa และอื่นๆอีกมากมาย โดยที่ไม่ต้องคำนึงถึงพอตที่โปรโตคอลนั้นๆใช้งาน

คุณลักษณะโดยรวมของL7-filter

- ใช้ได้กับเคอเนลเวอร์ชัน 2.4 และ 2.6
- สนับสนุน TCP,UDP,ICMP บน IPv4
- อาศัย Netfilter ในการตรวจสอบชนิดของโปรโตคอลในชั้นแอพพลิเคชันเลเยอร์
- ตรวจสอบข้อมูลแพคเกจจำนวนมากจากฝั่งหนึ่งไปอีกฝั่งหนึ่งได้
- ขนาดของแพ็คเกจที่ถูกตรวจสอบสามารถปรับแต่งได้
-มีคุณสมบัติของไฟล์วอลและQos
- สามารถกำหนดรูปแบบในการตรวจจับได้เอง
- ใช้ Regular Expression ในการจำแนกโปรโตคอล
- ถูกนำไปใช้ในการ บล็อกโปโตคอล, ควบคุมแบนวิธ, ตรวจจับ

4.1 การติดตั้งและการใช้งาน L7-filter

- ดาวน์โหลด l7-filter จาก http://l7-filter.sourcforce.net
- ดาวน์โหลด kernel จาก kernel.org ในที่นี้ใช้ kernel-2.4.28 จากนั้น rename เป็น linux-2.4.28 แล้วทำการแตก file ไปที่ /usr/src/
- แตกไฟล์ L7 และทำการ patch kernel โดยเลือกไฟล์ให้ตรงกับเคอเนลจากนั้นทำการ patch โดยใช้คำสั่ง patch –p1 < /path to kernel source/kernel-x.x.patch
- ทำการ patch ไอพีเทเบิล โดยใช้คำสั่ง patch –p1 < /path to iptable source/iptable.patch
- หลังจาก patch ไอพีเทเบิล แล้วให้พิมพ์คำสั่ง chmod +x extension/.layer7-test

- ปรับแต่ง config ตามนี้

1.Code maturity level options → Prompt for development and/or incomplete code/drivers (Y)

2.Networking Options → IP: Netfilter Configulation --> Packet Filtering (M)

3.Networking Options → IP: Netfilter Configulation -->Connection Tracking (M)

4.Networking Options → IP: Netfilter Configulation -->IP tables support (M)

5.Networking Options → IP: Netfilter Configulation --> Layer 7 match support (M)

- ส่วน option อื่นใน Netfilter Configulation ให้คงไว้

ทำการ complie kernel ในขั้นตอนนี้มีรายละเอียดดังนี้

1.ให้ copy config file จาก /boot มาที่ kernel sourcecode directory ที่ใช้ file config จาก/boot เพราะจะได้ลดปัญหาเรื่อง driver ของระบบเวลา complie kernel

cp /boot/config-2.4.20 /usr/src/kernel-2.4.28/.config

2. เรียกคำสั่ง make xconfig จาก directory kernel sourcecode

3. ปรับแต่ง kernel ตามนี้

4. เข้าไปแก้ไข makefile ใน /usr/src/linux-2.4.28/makefileแก้ไขข้อความเป็น

EXTRAVERSION=L7

5. เริ่ม complie kernel โดยใช้คำสั่ง make dep clean bzImage

6. จากนั้นจึง complie module พิมพ์ make modules

7. make modules_install เพื่อทำการติดตั้งโมดูล โมดูลที่ติดตั้งใหม่นี้จะไม่ติดตั้งทับที่เดิมแต่จะติดตั้งที่ใน /lib/modules/2.4.28L7

8. จากนั้นทำการสร้าง ram disk โดยใช้คำสั่ง Mkinitrd /boot/initrd-2.4.28L7.img 2.4.28L7

9. จากนั้น copy symbo table ของ kernel ใหม่ไปที่ boot โดยใช้คำสั่ง cp /usr/src/linux-2.4.28/System.map /boot/system.map-2.4.28L7

10. จากนั้นย้าย kernel ใหม่ไปที่ /boot โดยใช้คำสั่ง

cp /usr/src/linux2.4.28/arch/i386/boot/bzImage /boot/vmlinuz-2.4.28L7

11.copy .config ไปเก็บไว้ที่ /boot โดยใช้คำสั่ง cp /usr/src/linux-2.4.28 /boot/config-2.4.28L7

12.ทำให้ boot loader รู้จัก kernel ใหม่ในที่นี้ใช้ lilo ให้เพิ่มข้อความต่อไปนี้ต่อท้ายใน/etc/lilo.conf

```
image=/boot/vmlinuz-2.4.2
  lable=LinuxL7
     initrd =/boot/initrd-2.4.28L7.img
     read-only
     root=/dev/hda1
```

13. ทำการ update master boot record โดยใช้คำสั่ง lilo –v

- ทำการ complie ไอพีเทเบิล ใหม่เพื่อให้ ไอพีเทเบิล รู้จัก L7 ดังนี้

1.make KERNEL_DIR=<<where-you-built-your-kernel>>

2. make install KERNEL_DIR=<<where-you-built-your-kernel>>

L7-filter นั้นทำงานร่วมกับ ไอพีเทเบิล ดังนั้นคำสั่งที่ใช้งานจะเหมือนกับการเรียกใช้งานไอพีเทเบิลแต่จะมีส่วนเพิ่มเติมมาเล็กน้อย ตัวอย่างเช่น

```
Iptables –t filter –A FORWARD –m layer7 --l7proto edonkey –j DROP
```

จากตัวอย่างนี้หมายความว่าถ้ามีโปรโตคอล edonkey เข้ามาในเครือข่ายหรือออกไปจากเครือข่ายให้ทำการทิ้งแพคเกจ ในกรณีที่เรากำหนดรูปแบบการตรวจจับโปรโตคอลขึ้นเอง

```
Iptables –t filter –A FORWARD –m layer7 –l7dir /home/bob –l7proto mypat -j DROP
```

4.2 สิ่งที่ l7-filter ตรวจจับได้

-โปรโตคอล

- ประเภทของไฟล์

- โค้ดที่ไม่พึงประสงค์

บทที่5

การออกแบบระบบงาน

ในบทนี้จะกล่าวถึงขั้นตอนการทำงานของระบบและการออกแบบการทำงานขอโปรแกรมความต้องการของระบบโดยจะแสดงในรูปของการไหลของข้อมูล

5.1 ความต้องการของระบบ

ความต้องการของระบบ เอ็มเอสเอ็นไฟร์วอล ที่จะพัฒนาขึ้น มีดังต่อไปนี้

- ระบบดังกล่าวต้องทำงานในลักษณะ เว็ปแอพพลิเคชัน และทำงานบนระบบปฏิบัติการลีนุกซ์
- ผู้ขอใช้เอ็มเอสเอ็น จะต้องลงทะเบียนเพื่อขออนุญาตใช้งานเอ็มเอสเอ็น และ ต้องแจ้งอีเมล์และ รายละเอียดของผู้ที่ต้องการจะติดต่อลงในระบบ
- ผู้ดูแลระบบจะตรวจสอบข้อมูลผู้ขอใช้ เอ็มเอสเอ็น และพิจารณาตามความเหมาะสมในการขอใช้ เอ็มเอสเอ็น

5.2 การออกแบบการทำงานของระบบ

แบ่งออกเป็น 2 ส่วนคือ

5.2.1 การสร้างรูปแบบในการดักจับโปรโตคอลเอ็มเอสเอ็น

ในส่วนนี้ทำการศึกษาการทำงานของ การสื่อสารระหว่างเอ็มเอสเอ็น หรือการส่ง Instant message แบ่งออกเป็น 2 กรณี 1.ผู้ใช้เป็นผู้เชิญ 2.ผู้ใช้เป็นผู้ถูกเชิญ

- ผู้ใช้เป็นผู้เชิญ

การทำงานของโปรโตตอลจะเริ่มจากผู้ใช้ส่งคำสั่ง XFR พร้อมด้วยพารามิเตอร์ SBไปหา Notification Server (NS) NS จะตอบกลับด้วยคำสั่ง XFR พร้อมด้วย ไอพี และ พอตร์ ของ Switchboard Server ที่จะให้ไปเชื่อมต่อรวมถึงค่า hash จากนั้นผู้ใช้จะส่งคำสั่ง USR ตามด้วย อีเมล์ของผู้ใช้กับค่า hash ที่ได้มาจาก NS Switchboard Server จะตอบกลับด้วยคำสั่ง USR OK เมื่อได้รับคำสั่งนี้ผู้ใช้ถึงจะส่งคำสั่ง CAL ตามด้วย อีเมล์ ของผู้ที่จะติดต่อSwitchboard Server จะตอบกลับด้วยคำสั่ง CAL และพร้อมด้วย Session ID หากการเชื่อมต่อระหว่างผู้เชิญกับผู้ถูกเชิญสำเร็จ Switchboard Server ที่ผู้เชิญเชื่อมต่ออยู่จะส่งคำสั่ง JOI ตามด้วย อีเมล์ ของผู้ถูกเชิญมาให้

- ผู้ใช้เป็นผู้ถูกเชิญ

การทำงานของโปรโตคอลในขณะที่มีผู้เชิญเข้าสู่การสนทนา Notification Server จะส่งคำสั่ง RNG ตามด้วย Session ID, IP Addredd :PORT hash value User Account และ User Screen Name มาให้ผู้ถูกเชิญโดยที่ ค่า Session ID นั้นจะเป็นค่าเดียวกันกับค่าที่ผู้เชิญได้รับ ค่า IP Address:Port และ hash value นั้นจะเป็น ไอพี และ พอตร์ ของ Switchboard Server ที่จะให้ผู้ถูกเชิญเข้าไปเชื่อมต่อ ส่วน User Account และ User Screen Name จะเป็น อีเมล์ ของผู้เชิญ จากนั้นผู้ถูกเชิญจะตอบกลับด้วยคำสั่ง ANS ตามด้วย User Account ,hash value,Session ID ที่ได้มาจาก NS โดยที่ User Account คือ อีเมล์ ของผู้ถูกเชิญ ถ้าหากการเชื่อมต่อสำเร็จ Switchboard Server ทางผู้ถูกเชิญจะส่งคำสั่งIRO พร้อมด้วย อีเมล์ ของผู้เชิญ

ในการสร้างรูปแบบการดักจับแบ่งออกเป็น 2 รูปแบบ คือ รูปแบบของผู้ใช้เป็นผู้เชิญ กับรูปแบบของผู้ใช้เป็นผู้ถูกเชิญ

- รูปแบบของผู้ใช้เป็นผู้เชิญ

พิจารณาจากคำสั่งที่ใช้ในการเชื่อมต่อประกอบด้วย XFR , USR, CAL, JOI จากการวิเคราะห์พบว่าคำสั่ง JOI เหมาะที่จะนำมาสร้างรูปแบบการดักจับเพราะ คำสั่งนี้ไม่มีใช้ในส่วนอื่นๆและคำสั่งนี้ต้องส่ง อีเมล์ ของผู้ถูกเชิญมาด้วยทำให้สามารถระบุคนที่อนุญาตได้ง่าย

JOI email1|email2|email3

Email 1,email2,email3 คือ อีเมล์ ของผู้ที่อนุญาตให้ติดต่อ

- รูปแบบของผู้ใช้เป็นผู้ถูกเชิญ

พิจารณาคำสั่งที่ได้จาก Notification Server และ Switchboard Server ที่ส่งมาให้ผู้ถูกและคำสั่งที่ผู้ถูกเชิญส่งกลับไปยัง Switchboard Server ประกอบด้วยคำสั่ง RNG,ANS,IRO จากการวิเคราะห์พบว่าคำสั่ง IRO เหมาะสมที่สุดเนื่องจาก IRO เป็นคำสั่งที่มี อีเมล์ ของผู้เชิญตามมาด้วยทำให้สามารถระบุตัวได้

IRO [1-9][0-9][0-9]* [1-9]* [1-9]* email1| email2

Email 1,email2 คือ อีเมล์ ของผู้ที่อนุญาตให้ติดต่อ

5.2.2 ออกแบบเว็บเพจและการจัดเก็บลงฐานข้อมูล

ในส่วนนี้จะเป็นการออกแบบเว็บเพจเพื่อให้ผู้ใช้ใช้งานโดยระบบจะมีหน้าเว็บดังนี้

1.หน้าหลักของระบบจะเป็นหน้าเว็บสำหรับ ล็อกอิน

2.หน้าเว็บสำหรับกำหนดช่วง ไอพี ที่จะใช้ เอ็มเอสเอ็น

3.หน้าเว็บสำหรับสำหรับผู้พิเศษที่สามารถใช้งาน เอ็มเอสเอ็น โดยไม่ต้องขออนุญาต

4.หน้าเว็บสำหรับตรวจสอบข้อมูลผู้ขอใช้

5.หน้าเว็บสำหรับตรวจสอบข้อมูลของบุคคลที่ผู้ใช้ขอติดต่อ

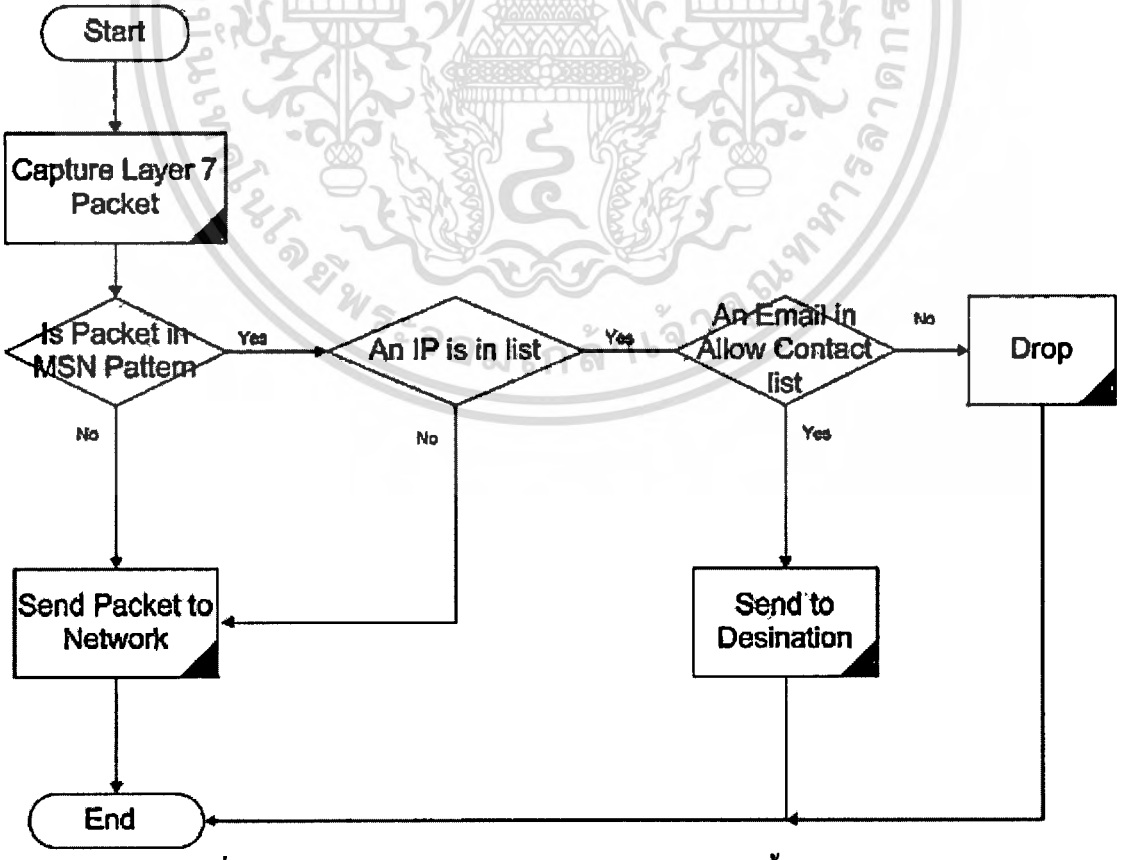
6.หน้าเว็บลงทะเบียนขอใช้ เอ็มเอสเอ็น

7.หน้าเว็บของผู้ใช้

8.หน้าเว็บสำหรับกรอกข้อมูลของผู้ที่จะติดต่อ

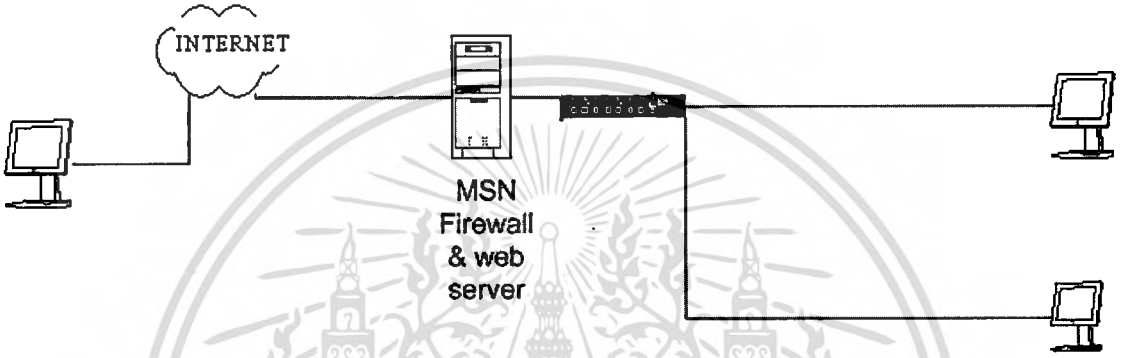
9.หน้าเว็บของผู้ใช้สำหรับดูรายชื่อผู้ที่จะติดต่อ

5.3 การทำงานของระบบในโครงงานนี้



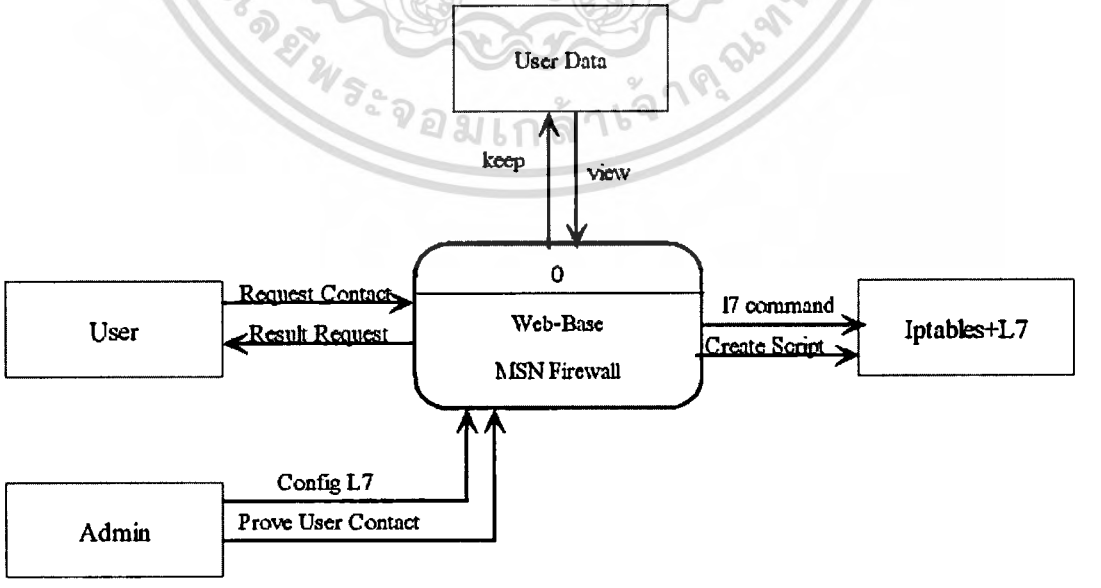
รูปที่ 5.1 แสดงการกรองแพ็คเกจของ เอ็มเอสเอ็นในชั้นแอพพลิเคชัน

จากรูป5.1เป็นการกรองแพ็คเกจของโปรโตคอล เอ็มเอสเอ็น ซึ่งจะทำการจับแพ็คเกจในชั้นที่7 มาวิเคราะห์ว่ามาจากโปรโตคอล เอ็มเอสเอ็น หรือไม่ถ้าไม่ใช่ก็ปล่อยผ่านไปถ้าใช่ก็จะส่งไปตรวจ ไอพี อยู่ในขอบเขตหรือไม่ถ้าไม่อยู่ในขอบเขตก็ใช้งาน เอ็มเอสเอ็น ได้ ถ้าอยู่ในขอบเขตก็จะดูเนื้อความในส่วนที่เป็นดาต้าว่ามี อีเมล์ ที่อนุญาตให้ผ่านหรือไม่ถ้า อีเมล์ ไม่ตรงก็ผ่านไม่ได้ ภาพรวมของโครงงานดังรูปที่ 5.2



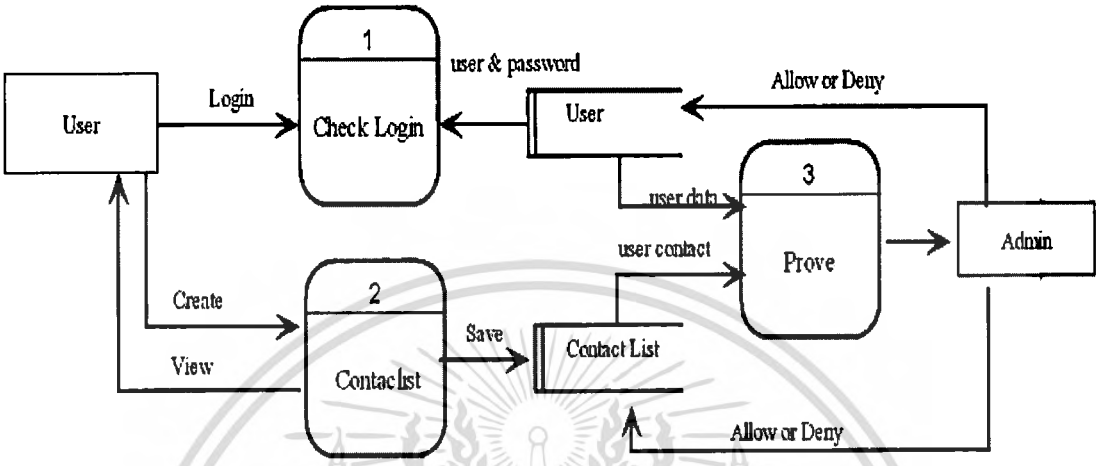
รูปที่ 5.2 ภาพรวมของโครงงาน

5.4 แผนผังการไหลของข้อมูลในโครงงาน



รูปที่ 5.3 แสดง Context Diagram ของโครงงาน

ผู้ดูแลระบบจะสั่งการ L7 ผ่านทางเว็บและนำข้อมูลของผู้ใช้มาแปลงเป็นคำสั่งของ L7



รูปที่ 5.4 แสดงผังการไหลของข้อมูลระดับที่ 0

- Check Login

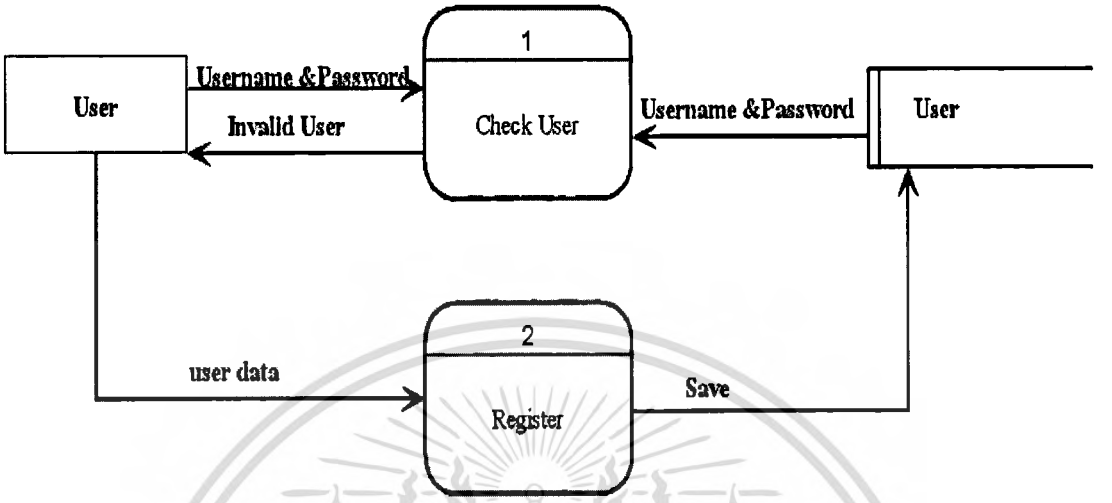
เป็นส่วนที่ผู้ใช้ต้อง ล็อกอิน เข้าสู่ระบบก่อนโดยที่ผู้ใช้ต้องมี ยูสเซอร์เนม และ พาสเวิดส์ แล้วระบบจะเอาไปตรวจสอบกับฐานข้อมูลผู้ใช้ ถ้าไม่มี ยูสเซอร์เนม ผู้ใช้สามารถลงทะเบียนกับระบบได้

- Contact list

เป็นที่สร้างรายชื่อผู้ที่ต้องการติดต่อของผู้ใช้ และผู้ใช้สามารถเรียกดูผลการตรวจสอบของผู้ดูแลระบบได้

- Prove

ผู้ดูแลระบบจากตรวจสอบข้อมูลใน contact list ของผู้ใช้แต่ละคนที่ร้องขอมาว่าสมควรจะอนุญาตหรือไม่



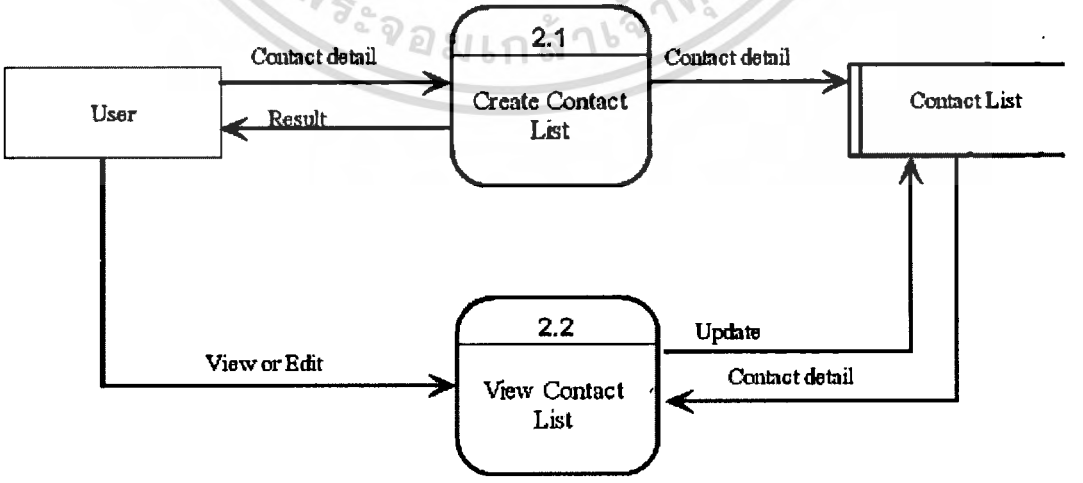
รูปที่ 5.5 แสดงผังการไหลของข้อมูลระดับที่ 1 กระบวนการที่ 1

- Check User

เป็นการตรวจสอบว่ามี user นี้อยู่ในระบบหรือไม่

- Register

เป็นการลงทะเบียนเพื่อเข้าสู่ระบบของผู้ใช้



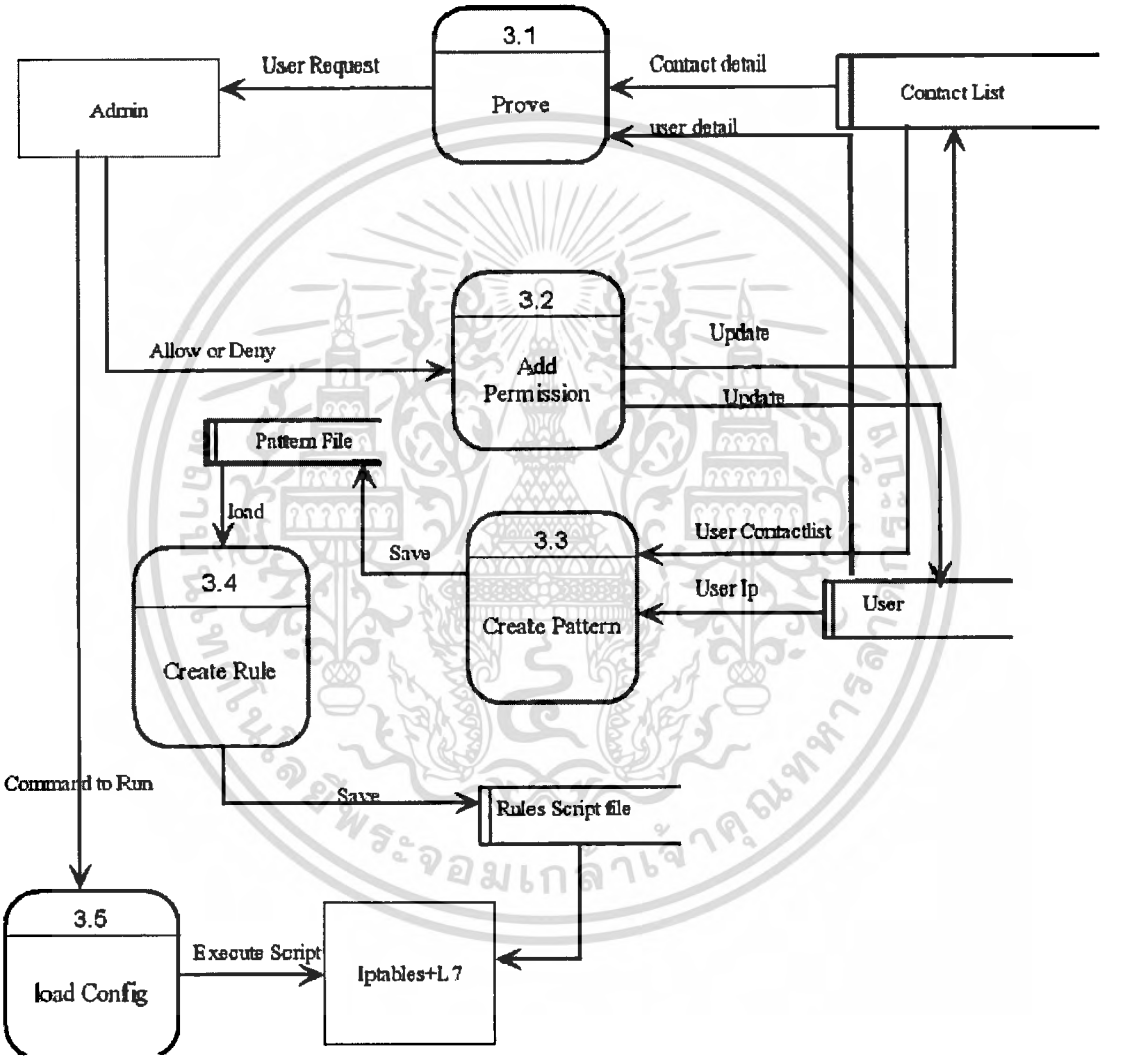
รูปที่ 5.6 แสดงผังการไหลของข้อมูลระดับที่1 กระบวนการที่ 2

- Create Contact list

เป็นการสร้างรายชื่อผู้ที่จะติดต่อให้ผู้ดูแลระบบตรวจสอบว่าจะอนุญาตหรือไม่

- View Contact list

เป็นการดูผลจากที่ผู้ดูแลระบบตรวจสอบ



รูปที่ 5.7 แสดงผังการไหลของข้อมูลระดับที่1 กระบวนการที่ 3

- Prove

เป็นการเรียกดูข้อมูลของผู้ขอใช้แต่ละคนที่ทำการร้องขอมา

- Add Permission

ผู้ดูแลระบบจะกำหนดสิทธิ์ให้กับรายชื่อที่ผู้ขอใช้ร้องขอ

- Create Pattern

เป็นการสร้างรูปแบบที่ผู้ดูแลระบบอนุญาตให้ติดต่อได้

- Create Rule

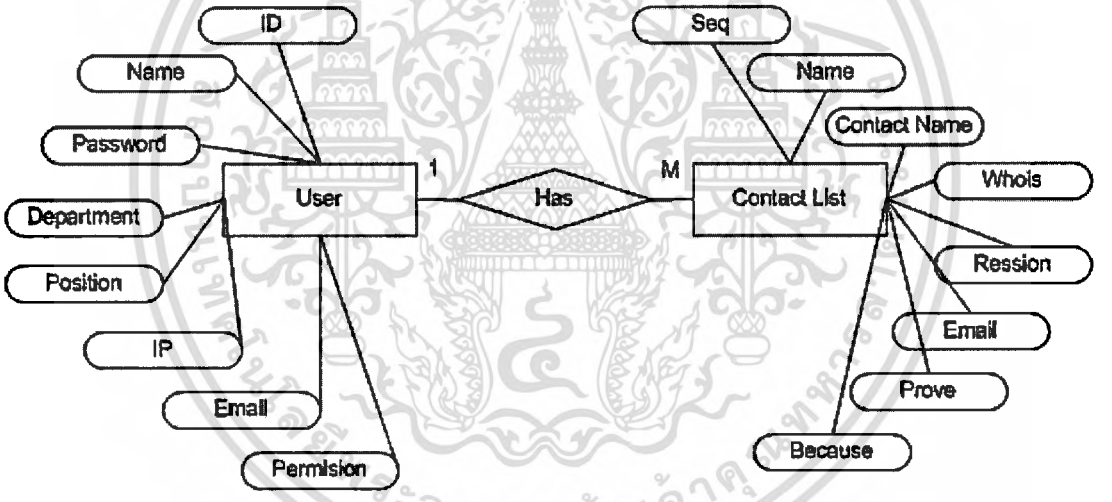
เป็นการประมวล ไฟล์แพทเทิน ของผู้ขอใช้เพื่อทุกคนเพื่อสร้างเป็น สคิป

- Load Config

เป็นการสั่งให้ L7 ทำงาน

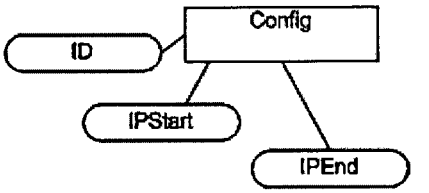
5.5 การออกแบบฐานข้อมูล

ในการพัฒนาระบบนี้จะเก็บข้อมูลของผู้ขอใช้และรายชื่อที่ผู้ขอใช้ต้องการจะติดต่อ รวมถึงขอบเขตของ ไอพี ที่ระบบจะต้องดูแล ลงในฐานข้อมูล โดยแบ่งออกเป็น 4 ตารางคือตารางผู้ขอใช้ ตารางรายชื่อที่ผู้ขอใช้ต้องการจะติดต่อ ตารางผู้มีสิทธิ์พิเศษ ตารางขอบเขตไอพี



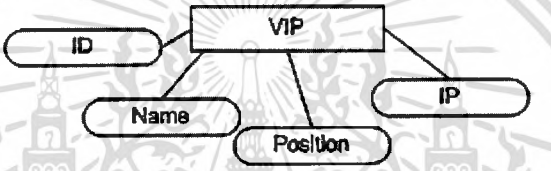
รูปที่ 5.8 แสดง E-R Diagram ของ MSN Firewall

จากรูปที่5.8 แสดงถึงความสัมพันธ์ของผู้ใช้กับรายชื่อที่ผู้ใช้ต้องการจะติดต่อ



รูปที่ 5.9 แสดง E-R Diagram ของ Config

จากรูปที่ 5.9เป็นส่วนที่เก็บขอบเขตไอพีที่จะควบคุมการใช้งาน เอ็มเอสเอ็น



รูปที่ 5.10 แสดง E-R Diagram ของ VIP

จากรูปที่ 5.10 เป็นส่วนที่เก็บข้อมูลผู้ใช้ที่ไม่ถูกจำกัดสิทธิ์การใช้ เอ็มเอสเอ็น

5.6 Data dictionary

ตารางที่ 5.1 User

Table Name : User Description : ข้อมูลผู้ใช้				
Name	Description	Type	Length	Key
ID	User ID	Integer	4	Pk
Name	User Name	Varchar	15	
Password	User Password	Varchar	10	
Department	Department	Varchar	10	
Position	User Position	Varchar	15	
IP	User IP Address	Varchar	15	
Email	User Email	Varchar	40	
Permision	Permision	Varchar	1	

ตารางที่ 5.2 Contact List

Table Name : Contact List Description : รายละเอียดของผู้ที่ติดต่อ				
Name	Description	Type	Length	Key
Seq	Sequence	Integer	4	Pk
Name	User Name	Varchar	15	
CName	Contact Name	Varchar	15	
Whois	This persion is	Varchar	15	
Ression	Why you need to contact	Varchar	100	
Email	Your contact Email	Varchar	40	
Prove	Allow or deny	Varchar	1	
Because	Why Admin set Deny	Varchar	100	

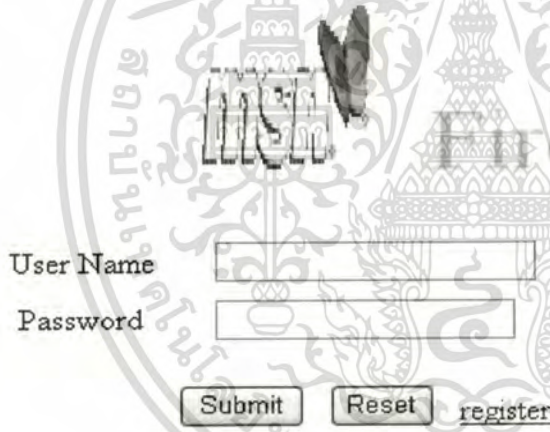
ตารางที่ 5.3 Config

Table Name : Config Description : ขอบเขตของ IP ที่จะใช้ เอ็มเอสเอ็น				
Name	Description	Type	Lenght	Key
ID	ID	Integer	4	Pk
IP Start	IP Address Start	Varchar	15	
IP End	IP Address End	Varchar	15	

ตารางที่ 5.4 VIP

Table Name : VIP Description : ผู้ใช้ที่ไม่ต้องขออนุญาต				
Name	Description	Type	Lenght	Key
ID	Sequence	Integer	4	Pk
Name	User Name	Varchar	15	
Position	User Position	Varchar	15	
IP	User IP Address	Varchar	15	

5.7 การออกแบบส่วนติดต่อผู้ใช้



รูปที่ 5.11 หน้าจอเข้าสู่ระบบ

จากรูปที่ 5.1 แสดงรูปหน้าจอหลักที่ผู้ใช้จะต้องใส่ ยูสเซอร์เนม และ พาสเวิดส์ เพื่อที่จะเข้าสู่ระบบ

รูปที่ 5.12 หน้าจอกำหนดขอบเขตของ ไอพี

จากรูปที่ 5.12 เป็นรูปแสดงหน้าจอที่ผู้ดูแลระบบกำหนดช่วง ไอพี ที่จะให้ใช้หรือไม่ให้ใช้ เอ็มเอสเอ็น

รูปที่ 5.13 หน้าจอกำหนดสิทธิ์การใช้งานของผู้ขอใช้

จากรูปที่ 5.13 แสดงหน้าจอที่ผู้ดูแลระบบจะอนุญาตให้ผู้ขอใช้ เอ็มเอสเอ็น สามารถใช้งานได้หรือไม่

Welcome : admin

Config MSN | Prove User Contact | Logout

Edit Config MSN

Name	Department	Position	IP	Permission
chand	sell	sell engineer	192.168.1.156	○Allow ○deny
narunat	sell	sell engineer	192.168.1.159	○Allow ○deny
nice	finance	ordit	192.168.1.154	○Allow ○deny
kapook	it	support	192.168.1.157	○Allow ○deny

Edit Config

รูปที่ 5.14 หน้าจอแสดงการแก้ไขสิทธิ์การใช้ เอ็มเอสเอ็น

จากรูปที่ 5.14 แสดงหน้าจอที่ผู้ดูแลระบบสามารถแก้ไขสิทธิ์ของผู้ขอใช้ เอ็มเอสเอ็น

Welcome : admin

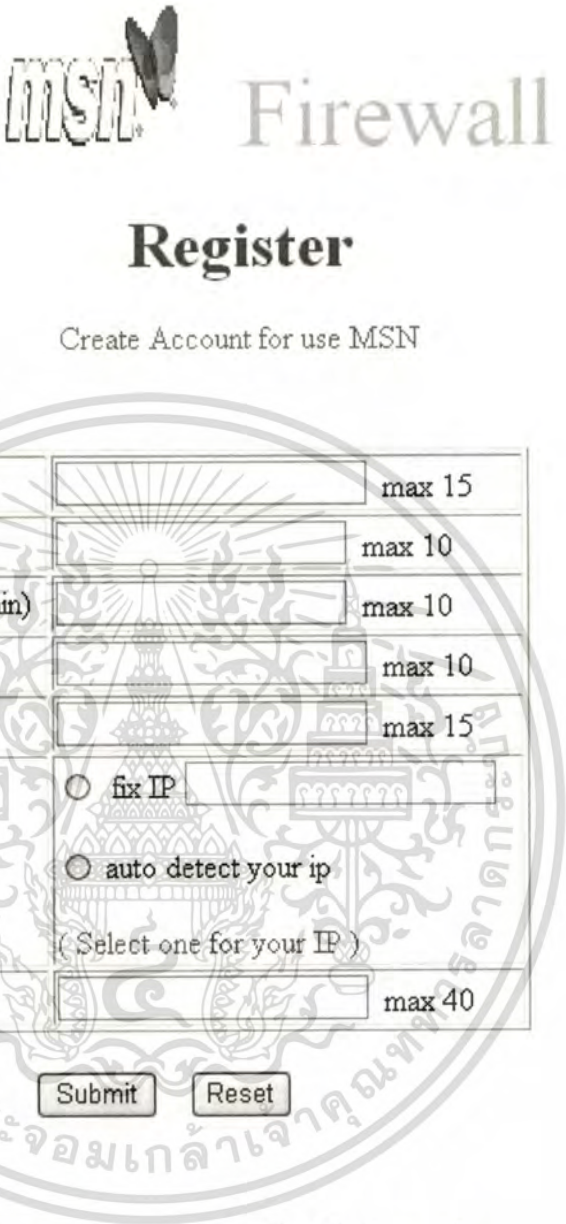
Config MSN | Prove User Contact | Logout

Name	Position	Contact Name	This Person is	Ression to contact	Permission
nice	ordit	chand	tech	ask program	○Allow ○deny Because:
nice	ordit	h2o	sister	talk	○Allow ○deny Because:
nice	ordit	linux	ajan	ask	○Allow ○deny Because:

Select

รูปที่ 5.15 แสดงหน้าจอตรวจสอบข้อมูล Contact list ของผู้ขอใช้ เอ็มเอสเอ็น

จากรูปที่ 5.15 แสดงหน้าจอที่ผู้ดูแลระบบสามารถจะตรวจสอบรายชื่อใน Contact List ที่ผู้ของใช้เอ็มเอสเอ็น ต้องการจะติดต่อ

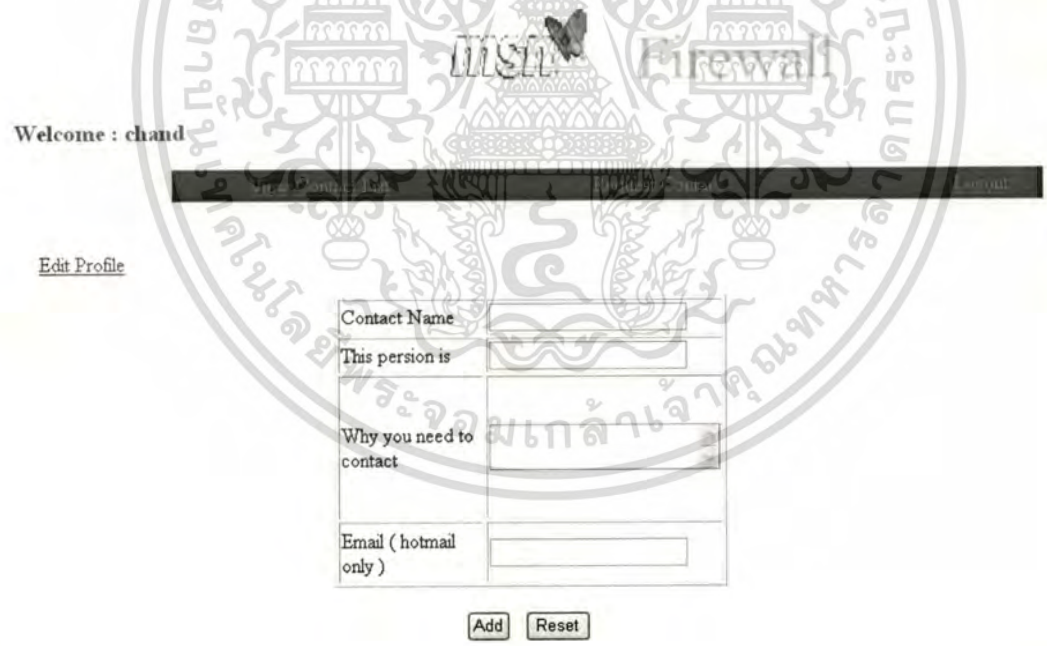


รูปที่ 5.16 หน้าจอการลงทะเบียนของผู้ขอใช้ เอ็มเอสเอ็น

จากรูปที่ 5.16 แสดงหน้าจอการลงทะเบียนการขอใช้ เอ็มเอสเอ็น

รูปที่ 5.17 หน้าจอของผู้ของใช้งานหลังจากลงทะเบียนกับระบบแล้ว

จากรูปที่ 5.17 แสดงหน้าจอหลักของผู้ขอใช้งาน เอ็มเอสเอ็น ที่ลงทะเบียนกับระบบแล้ว ประกอบด้วยเมนู
แก้ไขข้อมูลส่วนตัว ,เมนูลงทะเบียนรายชื่อผู้ที่จะติดต่อด้วย , เมนูรายชื่อผู้ที่จะติดต่อด้วย



รูปที่ 5.18 หน้าจอกรอกข้อมูลของผู้ที่จะติดต่อด้วย

จากรูปที่ 5.18 ถ้าผู้ใช้ต้องการติดต่อกับใครจะต้องบอกรายละเอียดของคนๆนั้นว่าเป็นใครทำไมถึงต้องการจะติดต่อด้วยให้ผู้ดูแลระบบทราบ

Welcome : chand

View Contact List | Request Contact | Logout

Edit Profile

Contact name	Email	Permission to Contact
h2o	bright_h2o@hotmail.com	Allow
h2o	bright_h2o@hotmail.com	Allow
h2o	bright_h2o@hotmail.com	Allow
linux	linuxl7@hotmail.com	Allow

Click Contact name to edit your contact detail

รูปที่ 5.19 หน้าจอ Contact list ของผู้ใช้

จากรูปที่ 5.19 แสดงถึง Contact list ของผู้ใช้โดยรายชื่อที่ปรากฏอยู่นี้จะมีทั้งที่อนุญาตและไม่อนุญาตโดยผู้ดูแลระบบ ซึ่งถ้าไม่อนุญาตผู้ดูแลระบบจะบอกเหตุผลให้ว่าทำไมไม่อนุญาตถ้าหากผู้ใช้ต้องการที่จะติดต่อกับคนที่ผู้ดูแลระบบไม่อนุญาตก็สามารถทำการแก้ไขเหตุผลโดย click ที่ชื่อของคนที่ผู้ดูแลระบบไม่อนุญาต

บทที่6

การพัฒนาระบบ

ระบบที่พัฒนาขึ้นนั้นมีจุดมุ่งหมายที่ทำงานบนระบบปฏิบัติการลีนุกซ์ ซึ่งใช้ เรดเฮ็ท 9 เป็นระบบปฏิบัติการ

6.1 เครื่องมือที่ใช้ในการพัฒนา

1. IPtables version 1.3.4
   เป็นไฟร์วอล์ บนลีนุกซ์
2. netfilter-layer7 version 2
   เป็นตัวกรองข้อมูลในระดับชั้นแอพพลิเคชัน
3. l7-protocols-2005-11-20
   เป็นรูปแบบของโปรโตคอลที่ทำงานในชั้น
4. kernel linux-2.4.28
5. MySql
   เป็นฐานข้อมูลของระบบที่ทำงานอยู่บนเครื่องเดียวกัน

6.2 ขั้นตอนในการพัฒนาระบบ

ภายในระบบประกอบด้วยฟังชันหลักๆดังนี้

- ฟังชันการเข้าสู่ระบบและตรวจสอบการเข้าใช้งานระบบ

เป็นฟังชันที่ทำการตรวจสอบผู้ใช้ที่จะเข้าสู่ระบบ โดยทำการสร้าง session ของผู้ใช้แต่ละคนไว้ซึ่งถ้ามีผู้ใช้คนใดมี session ไม่ถูกต้องก็จะถูกออกไปจากระบบ

- ฟังชันการทำงานกับฐานข้อมูล

เป็นฟังชันที่ทำหน้าที่ในการเชื่อมต่อ และจัดการกับฐานข้อมูลของระบบ

- ฟังชันสร้างคำสั่งของไอพีเทเบิล

เป็นฟังชันที่อ่านข้อมูลจากฐานข้อมูลของระบบและแปลงเป็นคำสั่งของไอพีเทเบิล

ตัวอย่างเช่น

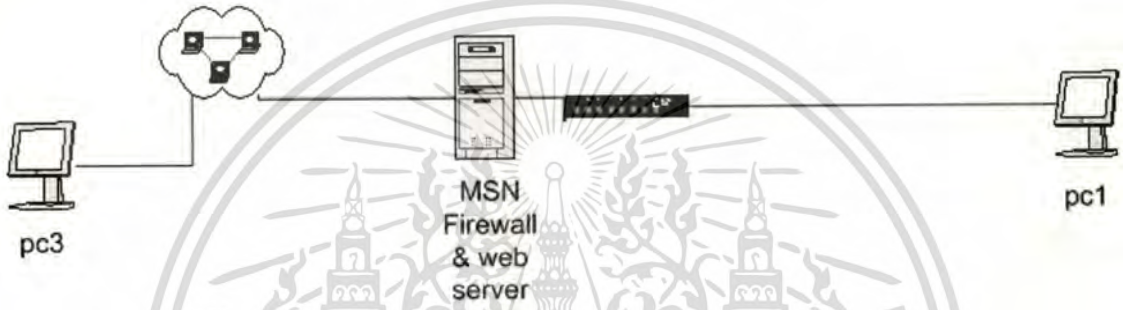
Iptables –t filter –I FORWARD 1 –m layer7 –l7dir /opt/msn/192.168.1.150 –l7proto ajoi

-j DROP

- ฟังชันในการสร้างรูปแบบของ เอ็มเอสเอ็น

6.3 การทดสอบการทำงานของระบบ

เนื่องจากการทดสอบต้องใช้เครื่อง อย่างน้อย 3 เครื่อง คือเครื่องที่เป็นไฟล์วอล เครื่องที่อยู่หลังไฟล์วอล และเครื่องที่อยู่นอกไฟล์วอลดังรูปที่



รูปที่ 6.1 แสดงโทโปโลยีที่ใช้ในการทดลอง

จากรูปที่ 6.1 พีซี1 ได้รับ ไอพี จาก ดีเฮ็ดซีพี เซิฟเวอร์ ซึ่งเป็นเครื่อง ไฟร์วอล 192.168.1.150 ส่วน พีซี3 มี ไอพี เป็น 161.246.46.139 สำหรับ ดีเฮ็ดซีพี เซิฟเวอร์ นั้นกำหนดให้จ่าย ไอพี ตั้งแต่ 192.168.1.150 – 192.168.1.200 แต่ในการทดลองจะกำหนดเงื่อนไขเพียง 5 ไอพี เท่านั้น มีรายละเอียดดังนี้

- 192.168.1.150 สามารถใช้ เอ็มเอสเอ็น ในการติดต่อกับ อีเมล์ ที่กำหนดให้เท่านั้น
- 192.168.1.151 ไม่สามารถใช้งาน เอ็มเอสเอ็น ได้
- 192.168.1.152 สามารถใช้ เอ็มเอสเอ็น ในการติดต่อกับ อีเมล์ ที่กำหนดให้เท่านั้น
- 192.168.1.153 สามารถใช้งาน เอ็มเอสเอ็น ได้ตามปกติ
- 192.168.1.154 ไม่สามารถใช้งาน เอ็มเอสเอ็น ได้

จากนั้นกำหนด อีเมล์ สำหรับใช้งาน เอ็มเอสเอ็น จำนวน 4 ชุดและกำหนดการสื่อสาร ดังนี้

- linuxl7@hotmail.com ล็อกอิน นี้ใช้ใน พีซี3
- ucraft@hotmail.com ล็อกอิน นี้ใช้ใน พีซี3

- lufy_d_monkey@hotmail.com ล็อกอิน นี้ใช้ใน พีซี1

- bright_h2o@hotmail.com ล็อกอิน นี้ใช้ใน พีซี3

กำหนดเงื่อนไขในการทดสอบดังนี้

- Lufy_d_monkey ไม่สามารถติดต่อกับ ucraft และ Bright_h2o ได้

- bright_h2o และ ucraft ไม่สามารถติดต่อกับ lufy_d_monkey ได้

- lufy_d_monkeyสามารถติดต่อกับ linuxl7 ในทางกลับกัน linuxl7 สามารถติดต่อ lufy_d_monkey ได้

จากนั้นนำเงื่อนไขกับ ip address ที่กำหนดมาสร้าง สคริป ของ l7 ได้ดังนี้

```
iptables -t filter -I FORWARD 1 -d 192.168.1.150 -m layer7 --l7dir /opt/msn/192.168.1.150 --
l7proto msn_joi -j ACCEPT
iptables -t filter -I FORWARD 2 -d 192.168.1.150 -m layer7 --l7dir /opt/msn/192.168.1.150 --
l7proto msn_deny3 -j DROP

iptables -t filter -I FORWARD 3 -d 192.168.1.150 -m layer7 --l7dir /opt/msn/192.168.1.150 --
l7proto msn_iro -j ACCEPT

iptables -t filter -I FORWARD 4 -d 192.168.1.150 -m layer7 --l7dir /opt/msn/192.168.1.150 --
l7proto msn_deny4 -j DROP

iptables -t filter -I FORWARD 5 -s 192.168.1.151 -m layer7 --l7proto msnmessenger -j DROP

iptables -t filter -I FORWARD 6 -d 192.168.1.152 -m layer7 --l7dir /opt/msn/192.168.1.152 --
l7proto msn_joi -j ACCEPT

iptables -t filter -I FORWARD 7 -d 192.168.1.152 -m layer7 --l7dir /opt/msn/192.168.1.152 --
l7proto msn_deny3 -j DROP

iptables -t filter -I FORWARD 8 -d 192.168.1.152 -m layer7 --l7dir /opt/msn/192.168.1.152 --
l7proto msn_iro -j ACCEPT
```

iptables -t filter -I FORWARD 9 -d 192.168.1.152 -m layer7 --l7dir /opt/msn/192.168.1.152 --l7proto msn_deny4 -j DROP

iptables -t filter -I FORWARD 10 -s 192.168.1.154 -m layer7 --l7proto msnmessenger -j DROP

6.4 สรุปผลการทดสอบ

- ระบบสามารถที่จะทำงานได้ตามที่ต้องการ
- การ บล็อก contact list ใน เอ็มเอสเอ็น โดยการระบุ อีเมล์ ทำงานได้

บทที่7
บทสรุปและข้อเสนอแนะ

7.1 บทสรุป

ในโครงงานนี้ได้ศึกษาโปรโตคอล เอ็มเอสเอ็น และ โปรแกรม แอลเจ็ด ซึ่งโปรแกรม แอลเจ็ด นี้ต้องทำงานร่วมกับ ไอพีเทเบิล ในการสร้างกฎขึ้นมานั้นค่อนข้างยุ่งยาก จึงนำเทคโนโลยีเว็บมาใช้เพื่อให้ง่ายต่อการใช้งานและสะดวกแก่ผู้ดูแลระบบ

ในช่วงการวิเคราะห์ออกแบบนั้นได้ศึกษาการใช้งาน ไอพีเทเบิล การสร้าง สคริป และการสร้างรูปแบบเฉพาะสำหรับโปรโตคอล ในระหว่างการพัฒนาระบบได้ ปรับปรุงการทำงานจนสามารถทำงานร่วมกับเว็บได้ ในการทดสอบนั้นเนื่องจากมีข้อจำกัดเรื่องเครื่องที่จะนำมาทดสอบจึงมีเครื่องใช้ในการทดลองเพียง 3 เครื่อง

7.2 ข้อดีและข้อเสียของระบบ

ข้อดีของระบบ

- ง่ายต่อการ บล็อก เอ็มเอสเอ็น ในระดับรายบุคคล
- สะดวกแก่การดูแลของผู้ดูแลระบบในการกำหนดสิทธิ์การใช้งาน เอ็มเอสเอ็น

ข้อเสียของระบบ

- หลังจาก สั่งให้ สคริป ทำงาน แล้ว ระบบยังไม่สามารถทำงานได้ทันที ต้องทิ้งช่วงเวลาของการทำงานประมาณ 1- 2 นาที ในระหว่างนี้ จะไม่สามารถบล็อกได้
- ยังไม่รองรับการ บล็อก การคุยผ่านไมโครโฟน

7.3 ข้อเสนอแนะ

ระบบงานนี้ทำได้เพียง บล็อก การสนทนาโดยการพิมพ์ เพื่อให้ระบบสมบูรณ์ยิ่งขึ้น ควรให้มีความสามารถ บล็อก การคุยผ่านไมโครโฟน ระหว่างบุคคลได้ และมีฟังชันแจ้งเตือนแก่ผู้ดูแลระบบเมื่อมีผู้ใช้มา request contact list

บรรณานุกรม

ณัฐพล พันธวงศ์ ,นนทพล สมราว,นวลพรรณ ศรีฟ้า,สน หาญวงศ์ . รายงานวิชา Computer Network Technology เรื่อง MSN Protocol. ปีการศึกษา 2546 [IS15.1] คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภูวดล ด่านระหาญ. 2544 .linux2.4 stateful firewall : iptables. [online].Available: http://www.thaicert.nectec.or.th/paper/firewall/iptables.php

สงกรานต์ ทองสว่าง. 2544. MySql ระบบฐานข้อมูลสำหรับอินเตอร์เน็ต. บริษัท ซีเอ็ดยูเคชัน จำกัด (มหาชน)

Matthew Strait .2003. Application Layer Packet Classifier for Linux . [online].Available http://l7-filter.sourceforge.net/

ประวัติผู้เขียน

ชื่อผู้เขียน	นายชาญ เชาว์ปฏิภาณ
วันเกิด	6 ธันวาคม 2522
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วท.บ (วิทยาศาสตร์ฟิสิกส์ประยุกต์)
	สาขาโซลิดสเตทอิเล็กทรอนิกส์
	คณะวิทยาศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
	ปีการศึกษา 2545