ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง

DIRECT FIREWALL POLICY CONTROLLER

โดย

นายทิวานนท์ จำพรต

นายเทพสรรค์ ปลอดอินทร์

นายธนกร ดอนนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง
Direct Firewall Policy Controller

โดย

นาย ทิวานนท์ จำพรต

นาย เทพสรรค์ ปลอดอินทร์

นาย ธนกร ดอนนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง
Direct Firewall Policy Controller

โดย

นาย ทิวานนท์	จำพรต	รหัสนักศึกษา	53010598
นาย เทพสรรค์	ปลอดอินทร์	รหัสนักศึกษา	53010600
นาย ธนกร	ดอนนา	รหัสนักศึกษา	53010611

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิพล สิทธิชีวภาค

รศ.เกรียงไกร วงศ์โรจนภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

10 / ม.ค. / 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

กรรมการผู้ตรวจชิ้นงาน

13 / ม.ค. / 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญานิพนธ์ปีการศึกษา 2556
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง
- Directory Firewall Policy Controller

ผู้จัดทำ

1.	นายทิวานนท์	จำพรต	53010598
2.	นายเทพสรรค์	ปลอดอินทร์	53010600
3.	นายธนกร	ดอนนา	53010611

..... อาจารย์ที่ปรึกษา
(รศ.ดร.สุวิพล สิทธิชีวภาค)

..... อาจารย์ที่ปรึกษา
(รศ.เกรียงไกร วงศ์โรจนภรณ์)

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยคำแนะนำที่มีค่าและให้คำปรึกษาที่ดียิ่งโดย รศ.ดร.สุวิพล สิทธิชีวภาค และรศ.ดร.เกรียงไกร วงศ์โรจนภรณ์ ที่ค่อยช่วยเหลือ สั่งสอน อนุเคราะห์เครื่องมือ และอุปกรณ์ต่างๆ ขอขอบคุณสำนักบริการคอมพิวเตอร์โดยเฉพาะยิ่ง คุณ กฤษฎิ์ธนิก ศรีธนสาร สำหรับคำแนะนำ คำปรึกษาในเรื่องต่างๆ จนสามารถแก้ไขปัญหาได้ลุล่วง ร่วมถึงเพื่อน และพี่ๆที่อยู่ในห้อง T304 ที่ค่อยให้คำปรึกษาอย่างเป็นกันเอง ทำให้โครงงานสำเร็จตามที่คาดหวัง

ทั้งนี้ขอขอบพระคุณคณาจารย์ให้ความรู้ และสั่งสอน รวมถึงคุณพ่อ และคุณแม่ ที่ให้การอบรมเลี้ยงมาเป็นอย่างดี จนทำให้คณะผู้จัดทำมีชีวิตที่ดีมาจนถึงทุกวันนี้

นาย ทิวานนท์ จำพรต
นาย เทพสรรค์ ปลอดอินทร์
นาย ธนกร ดอนนา
ผู้จัดทำ

ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง

Direct Firewall Policy Controller

โดย นาย ทิวานนท์ จำพรต 53010598
นาย เทพสรรค์ ปลอดอินทร์ 53010600
นาย ธนกร ดอนนา 53010611
อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิทธิชีวภาค
อาจารย์ที่ปรึกษาร่วม รศ.เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นโครงงานพัฒนาแอพพลิเคชั่นคำสั่งสำหรับควบคุมไฟร์วอลล์ในเครื่องเซิร์ฟเวอร์ โดยควบคุมจากไมโครคอนโทรลเลอร์ MCS-51 ส่งคำสั่งผ่านทางพอร์ต RS232 เพื่อใช้รักษาความปลอดภัยให้กับระบบเครือข่าย สามารถควบคุมคอนเนคชั่นตามนโยบายที่ได้ถูกกำหนดไว้อย่างเฉพาะเจาะจง ทำให้ระบบเครือข่ายที่อยู่ภายใต้ระบบควบคุมดังกล่าวมีความรัดกุมในการเข้าใช้บริการของเครือข่าย โดยไฟร์วอลล์แอพพลิเคชั่นการพัฒนาบนระบบปฏิบัติการลินุกซ์ ด้วยภาษาจาวาและโปรเซสบิวเดอไลบารีที่ทำงานร่วมกันกับไอพีเทเบิล ซึ่งเป็นเซอร์วิสที่ทำงานภายใต้ระบบปฏิบัติการลินุกซ์

ABSTRACT

This is a thesis which is developed an application for controlling the Firewall at server capabilities of the MCS-51. The microcontroller sends commands through the RS232 port to security to the network control connection to a policy. The policy has been specifically defined make network under such control systems are straining to hold in the access to the network by a firewall, an application development based on the Linux operating system. The operation system which the Java language and processes builder's library interoperability with IP Tables.

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีการนำเอาระบบเครือข่ายอินเทอร์เน็ตมาเป็นส่วนสำคัญในการเชื่อมโยงการสื่อสารอย่างแพร่หลาย การเชื่อมโยงข้อมูลดังกล่าวในระบบเครือข่ายจำเป็นต้องการป้องกันการโจมตีจากภายนอก โดยใช้แอพพลิเคชั่นไฟล์วอล์ (Firewall)

ไฟล์วอลล์ จะทำหน้าที่เป็นตัวกั้นระหว่างระบบเครือข่ายกับระบบเครือข่ายอื่นๆภายนอกที่เข้ามาเชื่อมต่อ โดยจะมีการกำหนดเงื่อนไขในการให้ร้องขอแพ็กเกต (Packet) ต่างๆของผู้ใช้งานระบบเครือข่ายอินเตอร์ให้เป็นไปตามที่ผู้ใช้งานแอพพลิเคชั่นไฟลวอลล์กำหนดขึ้นผ่านหน้าจอของแอพพลิเคชั่น

เมื่อพิจารณาในอีกด้านหนึ่งหนึ่งจะเห็นว่าการใช้ไฟล์วอลล์ในแบบปกติจะคอนฟิก(Config) ผ่านเว็บเบส (Web base) เมื่อระบบอินเทอร์เน็ตขัดข้อง จะด้วยตัวอุปกรณ์ หรือจะเป็นปัญหาที่ตัวไฟล์วอลล์เอง การตรวจสอบแก้ไขจะมีขั้นตอนยุ่งยากมาก โดยเริ่มจากการตรวจหาจุดผิดพลาดของเงื่อนไขที่กำหนดไปเสียก่อน ถ้าพบว่าเงื่อนไขใดทำให้เกิดความซ้ำซ้อนก็ทำการเปลี่ยนใหม่ ด้วยเหตุนี้จึงนำไมโครคอนโทรเลอร์มาใช้งานในการเป็นตัวควบคุมเงื่อนไขของแอพพลิเคชั่นไฟล์วอลล์

1.2 วัตถุประสงค์

1.เพื่อศึกษาหลักการทำงานของระบบรักษาความปลอดภัยให้กับเครือข่าย โดยใช้ไอพีเทเบิล (IP Tables) บนเซอร์วิส (Service) ระบบปฏิบัติการลินุกส์ อูบุนตู (Linux Ubuntu)

2.เพื่อพัฒนาแอพพลิเคชั่นไฟล์วอลล์ให้สามารถคอนฟิกโพลิซีโดยไม่จำเป็นต้องรู้ซินเท็ก (Syntax)

3.ศึกษาการทำงานของไมโครคอนโทรเลอร์ (Microcontroller) เพื่อควบคุมหน้าจอแอลซีดี (LCD Display) ในส่วนของการแสดงผล และ รับค่าอินพุตส่งเข้าไปในตัวเครื่องแม่ข่าย

1.3 ขอบเขตของโครงงาน

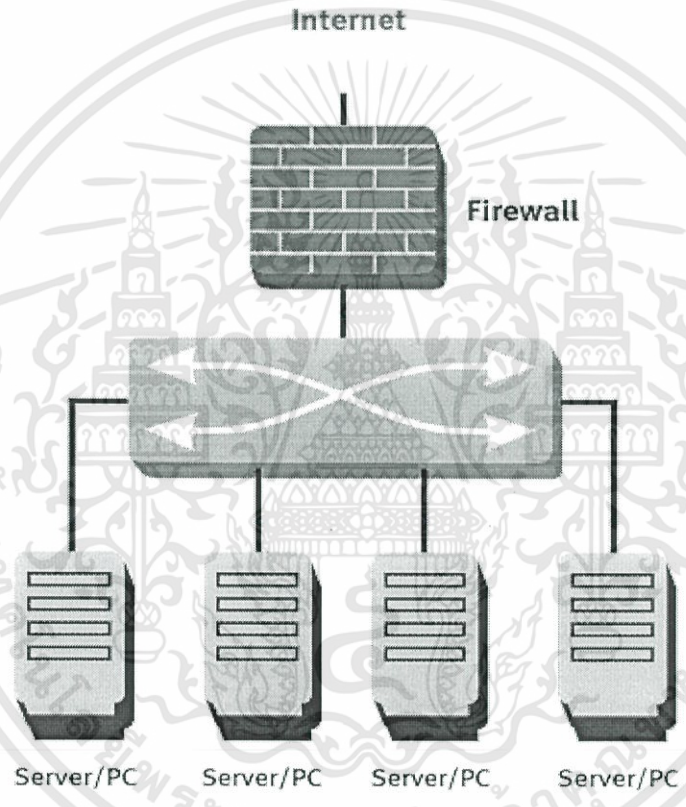
โครงงานนี้พัฒนาระบบไฟล์วอลล์ที่ป้อนคำสั่งโดยผ่านหน้าจอแอลซีดี (LCD Display) โดยพัฒนาด้วยภาษาจาวาและโปรเซสบิวเดอร์ไลบารี่ที่ทำงานร่วมกันกับไอพีเทเบิล ซึ่งหน้าจอแสดงผลสามารถแสดงสถานะของระบบไฟล์วอลล์ได้ ใช้ RS232 เป็นพอร์ตส่งผ่านข้อมูลระหว่างส่วนสั่งการกับส่วนประมวลผล

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไฟร์วอลล์ (Firewall)

ไฟร์วอลล์คือซอฟต์แวร์หรือฮาร์ดแวร์ที่ตรวจสอบข้อมูลที่มาจากอินเทอร์เน็ตหรือเครือข่าย แล้วบล็อคข้อมูลนั้นหรืออนุญาตให้ข้อมูลนั้นผ่านเข้ามายังคอมพิวเตอร์ ก็คือ เป็นระบบที่เอาไว้ป้องกันอันตรายจากอินเทอร์เน็ตหรือเน็ตเวิร์คภายนอก ดังแสดงในบล็อกไดอะแกรมดังรูปที่ 2.1

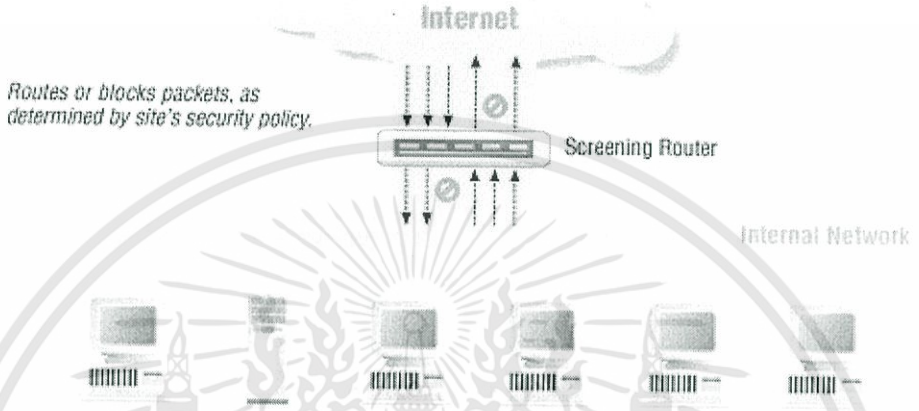


รูปที่ 2.1 บล็อกไดอะแกรมของไฟร์วอลล์

จากรูป ไฟร์วอลล์จะกั้นระหว่างระบบเครือข่ายที่เชื่อมต่อภายใต้เครื่องให้บริการหรือเครื่องแม่ข่ายกับระบบเครือข่ายภายนอก โดยแพ็กเกต (Packet) ที่ผ่านเข้า-ออกเครื่องแม่ข่ายจะถูกพิจารณาตามโพลิซี (Policy) ที่กำหนดไว้

2.1.1 ไฟร์วอลล์ชนิดแพ็กเกตฟิลเตอร์ริ่ง (Packet Filtering)

แพ็กเกตฟิลเตอร์ (Packet Filter) คือเราเตอร์ที่ทำการหาเส้นทางและส่งต่อ (route) อย่างมีเงื่อนไข โดยจะพิจารณาจากข้อมูลส่วนที่อยู่ในเฮดเดอร์ (header) ของแพ็กเกตที่ผ่านเข้ามาเทียบกับกฎ (rules) ที่กำหนดไว้และตัดสินว่าควรจะทิ้ง (drop) แพ็กเกตนั้นไปหรือว่าจะยอม (accept) ให้แพ็กเกตนั้นผ่านไปได้ดังแสดงในบล็อกไดอะแกรมดังรูปที่ 2.2



รูปที่ 2.2 บล็อกไดอะแกรมของไฟร์วอลล์ชนิด Packet Filtering

จากรูป ในการพิจารณาเฮดเดอร์แพ็กเกตฟิลเตอร์ จะตรวจสอบในระดับของอินเทอร์เน็ตเลเยอร์ (Internet Layer) และทรานสปอร์ตเลเยอร์ (Transport Layer) ในอินเทอร์เน็ตโมเดล ซึ่งในอินเทอร์เน็ตเลเยอร์จะมีแอตทริบิวต์ที่สำคัญต่อ Packet Filtering ดังนี้

- ไอพีต้นทาง (Source IP)
- ไอพีปลายทาง (Destination IP)
- ชนิดของโปรโตคอล (TCP UDP และ ICMP)

และในระดับของทรานสปอร์ตเลเยอร์ มีแอตทริบิวต์ที่สำคัญคือ

- พอร์ตต้นทาง
- พอร์ตปลายทาง
- แฟล็ก (Flag ซึ่งจะมีเฉพาะในเฮดเดอร์ของแพ็กเกตTCP)

ซึ่งพอร์ตของทรานสปอร์ตเลเยอร์ คือทั้ง TCP และ UDP นั้นจะเป็นสิ่งที่บอกถึงแอพพลิเคชั่นที่แพ็กเกตนั้นต้องการติดต่อด้วยเช่น พอร์ต 80 หมายถึง HTTP, พอร์ต 21 หมายถึง FTP เป็นต้น ดังนั้นเมื่อแพ็กเกตฟิลเตอร์พิจารณาเฮดเดอร์ จึงทำให้สามารถควบคุมแพ็กเกตที่มาจากที่ต่างๆ และมีลักษณะต่างๆ (ดูได้จากแฟล็กของแพ็กเกตหรือชนิดของ ICMP ในแพ็กเกตICMP) ได้ เช่น ห้ามแพ็กเกตทุกชนิดจาก crack.cracker.net เข้ามายังเน็ตเวิร์ค 203.154.207.0/24 , ห้ามแพ็กเกตที่มีไอพีต้นทางอยู่ในเน็ตเวิร์ค 203.154.207.0/24 ผ่านเราเตอร์เข้ามา (ในกรณีนี้เพื่อเป็นการป้องกันไอพีสปูฟิก (ip spoofing)) เป็นต้น

ข้อดีของแพ็กเกตฟิลเตอร์ริ่ง คือ ไม่ขึ้นกับแอพพลิเคชั่น มีความเร็วสูงรองรับการขยายตัวได้ดี ส่วนข้อเสีย คือ บางโปรโตคอลไม่เหมาะสมกับการใช้แพ็กเกตฟิลเตอร์ริ่ง เช่น FTP, ICQ เป็นต้น

2.2 ไอพีเทเบิล (IP Tables)

ไอพีเทเบิล เป็นเครื่องมือฟรีแวร์ตัวหนึ่งที่มีอยู่ในลินุกซ์ (Linux) ทำหน้าที่ตรวจแพ็กเกต ต่างๆที่เข้ามายังลินุกซ์ รวมทั้งทำการส่งต่อแพ็กเกตใดๆก็ตามที่เข้ามาได้ ไอพีเทเบิลสามารถใช้งานเป็นไฟร์วอลล์ได้ตั้งแต่เคอร์เนล 1.1 ซึ่งเป็นเวอร์ชันแรก โดย Alan Cox ใช้ชื่อว่า ipfw (จาก BSD) ต่อมาลินุกซ์ 2.0 ได้ถูกพัฒนาและปรับปรุงเป็นเครื่องมือที่มีชื่อว่า ipfwadm โดยเครื่องมือชิ้นนี้อนุญาตให้ผู้ใช้สามารถควบคุมฟิลเตอร์ริ่งรูล (filtering rule) ได้ และต่อมาลินุกซ์ 2.2 ก็ได้สร้างเครื่องมือตัวใหม่ชื่อไอพีเชน (ipchains) ซึ่งเผยแพร่ในปี 1998 โดย Rusty Russel และทีมงาน ทั้งนี้ไอพีเชนนี้ถือได้ว่าเป็นพัฒนาการขั้นที่สามของลินุกซ์ไฟล์วอลล์ (Linux Firewall) จวบจนกระทั่งในปัจจุบันก็มีเน็ตฟิลเตอร์(netfilter) และไอพีเทเบิล ซึ่งถือว่าเป็นพัฒนาการขั้นที่สี่ของลินุกซ์ไฟล์วอลล์ เน็ตฟิลเตอร์นั้นเป็นชื่อใหม่ของโค้ดที่ทำหน้าที่เป็นแพ็กเกตแฮนด์เลอร์ (Packet Handler(Stateful inspection)) ใน Linux kernel 2.4 (จริงๆคือเวอร์ชัน 2.3.15 และเวอร์ชันต่อๆมา) ซึ่งได้ถูกออกแบบและปรับปรุงใหม่จากเวอร์ชันก่อนหน้านี้ เป็นเรื่องที่น่ายินดีคือ เน็ตฟิลเตอร์ นั้นสามารถทำงานย้อนหลังร่วมกับไอพีเชนและ ipfwadm ได้ และคำสั่งในการเรียกใช้งานคือ ไอพีเทเบิล ทว่าในเอกสารอ้างอิงบางตัวได้สร้างความสับสนขึ้น เนื่องจากการกล่าวอ้างถึงไอพีเทเบิลและเน็ตฟิลเตอร์ในเชิงว่าเป็นสิ่งเดียวกันซึ่งผิดไปจากความเป็นจริงที่ว่า ไอพีเทเบิล นั้นเป็นเพียงส่วนหนึ่งของเน็ตฟิลเตอร์โดยเน็ตฟิลเตอร์คือชุดของฮุคส์ (hooks) ในเน็ตเวิร์คโปรโตคอลสแตก (network protocol stacks) ซึ่งอนุญาตให้ไอพีเทเบิลโมดูล (iptables module) จัดการกับเน็ตเวิร์คแพ็กเกต (network packets) ที่เดินทางอยู่ภายในโปรโตคอลสแตก (protocol stack) ซึ่ง เน็ตฟิลเตอร์ เฟรมเวิร์ค อนุญาตให้ไอพีเทเบิลโมดูลทำการ forward หรือ drop แพ็กเกตที่กำลังทำการเดินทางอยู่ภายในโปรโตคอลสแตก ได้ ซึ่งโดยทั่วแล้วมีการนำไอพีเทเบิล เขามาประยุกต์ใช้งานเพื่อทำหน้าที่เป็นเน็ตเวิร์คแอดเดรสทรานสเลชั่น (Network Address Translation (NAT)) หรือเรียกว่าการทำ Masquerading โดยบรรจุอยู่ในเครื่องที่ทำหน้าที่เป็นอุปกรณ์จัดเส้นทางหรือไฟร์วอลล์ในเครือข่ายภายใน ดังนั้นจึงไม่มีความจำเป็นที่ต้องใช้ไอพีเทเบิล ในเครื่องที่เป็นเวิร์คสเตชัน (workstation) หรือเครื่องอื่นๆ ภายในเครือข่ายแต่อย่างไร

2.2.1 การใช้คำสั่งไอพีเทเบิล

โดยปกติแล้วรูปแบบการใช้งานไอพีเทเบิลเบื้องต้นจะมีรูปแบบการใช้งานที่ต้องจดจำซินเท็ก คือ Iptables<command><match><target/jump> โดยกฎที่เขียนขึ้นจะเป็นเป็นตัวบอกเคอร์เนลว่าให้กระทำอย่างไรในกรณีที่พบแพ็กเกตตรงตามที่ระบุไว้ หมายถึง ตาราง ที่ต้องการระบุ มี 3 ตาราง คือ

- ฟิลเตอร์เทเบิล (Filter table)
- แนตเทเบิล (Nat table)
- แมงเกิลเทเบิล (Mangle table)

เช่น iptables -t nat หมายถึงให้ทำงานกับแนตเทเบิล ในกรณีที่ไม่ได้ระบุตาราง ไอพีเทเบิล จะถือว่าคำสั่งดังกล่าวระบุถึงฟิลเตอร์เทเบิลโดยอัตโนมัติ

<Command>จะเป็นตัวสั่งให้ไอพีเทเบิลทำในสิ่งที่ต้องการ เช่น iptables -A INPUT ซึ่งหมายถึงให้สร้างกฎต่อท้ายอินพุตเชน (INPUT chain) ในฟิลเตอร์เทเบิล

<Match>เป็นส่วนที่ใช้ตรวจสอบว่าแพ็กเกต มีข้อมูลตรง (match) กับที่ระบุไว้หรือไม่ เช่น มีไอพีแอดเดรสต้นทาง (source ip address) เป็น 1.2.3.4

<Target/jump>เป็นตัวระบุว่าเมื่อเจอแพ็กเกตที่ตรงก็จะกระทำ (action) ตามที่ระบุไว้ เช่น ถ้าแพ็กเกตใดมีไอพีแอดเดรสต้นทาง (source ip address) เป็น 1.2.3.4 ให้ทิ้งแพ็กเกตนั้น (DROP packet)

2.2.2 การส่งผ่านแพ็กเกตในฟิลเตอร์

โดยปกติเคอร์เนล (Kernel) จะมาพร้อมกับกฎพื้นฐาน 3 รายการ ซึ่งบรรจุอยู่ในตารางของฟิลเตอร์ ซึ่งเรียกรายการเหล่านี้ว่าไฟล์วอลล์เชน (firewall chains) หรือเชน (chains) โดยจะเรียกรายการต่างๆ ในตารางดังกล่าวนี้ว่า INPUT, OUTPUT และ FORWARD เมื่อมีแพ็กเกตผ่านเข้ามาทางอีเทอร์เน็ต (Ethernet) อื่นๆ เคอร์เนลจะใช้ INPUT chain ในการตัดสินใจเพื่อกำหนดเส้นทางของแพ็กเกตดังกล่าว ถ้า แพ็กเกตมีสิทธิที่จะผ่านไปได้ เคอร์เนลจะตัดสินใจว่าจะต้องส่งแพ็กเกตต่อไปยังที่ใดซึ่งเรียกว่าเราติ้ง (routing) และเมื่อแพ็กเกต มีจุดหมายปลายทางไปยังเครื่องอื่น เคอร์เนลจะตัดสินใจในการส่งต่อแพ็กเกต โดยพิจารณาจาก Forward chain และท้ายที่สุดก่อนที่แพ็กเกตจะถูกส่งออกไปเคอร์เนลจะพิจารณาจาก Output chain ในการเลือกเส้นทางหรืออินเตอร์เฟซ (interface) ที่ต้องทำการส่งต่อไปยังเชน คือ รายการตรวจสอบของกฎต่างๆ ซึ่งในทุกๆกฎมีการทำงานคือ ถ้ามีแพ็กเกตที่มีเฮดเดอร์เหมือนกับกฎที่ตั้งไว้แล้วจะต้องทำอะไรต่อไปกับแพ็กเกตดังกล่าวและถ้าแพ็กเกตไม่ตรงกับกฎที่ได้ตั้งไว้ ก็จะทำการพิจารณาจากกฎลำดับถัดไปในเชน และในท้ายที่สุดถ้าแพ็กเกตที่เข้ามาไม่ตรงตามกฎ ใด ๆ ในเชนเลยตัวเคอร์เนลจะทำการพิจารณาจากเชนโพลีซี(chain policy) เพื่อระบุว่าต้องจัดการกับแพ็กเกตดังกล่าวอย่างไรต่อไป ซึ่งในระบบความปลอดภัยโดยปกติแล้ว จะทำการกำหนดให้รีเจคท์ (reject) หรือดีไนแพ็กเกต (deny packet) ดังกล่าว หรืออีกนัยหนึ่งก็คือแพ็กเกตดังกล่าวจะไม่ถูกส่งต่อไปยังจุดหมายปลายทาง นั่นคือไม่สามารถที่จะผ่านตัวไฟร์วอลล์เพื่อเข้าหรือออกจากเครือข่ายได้

2.3 หมายเลขพอร์ต (Port Number)

หมายเลขพอร์ต คือเลขฐานสิบ 16 บิตตั้งแต่ 0 ถึง 65535 หมายเลขพอร์ตแต่ละหมายเลขจะถูกกำหนดโดยเฉพาะจากระบบปฏิบัติการ (Operating Systems) ทาง Internet Assigned Numbers Authority (IANA) เป็นหน่วยงานกลางในการประสานการเลือกใช้พอร์ต ว่าพอร์ตหมายเลขใดเหมาะสำหรับเซอร์วิสใด เช่น เลือกใช้พอร์ตหมายเลข 23 กับเซอร์วิสเทลเน็ต (Telnet) และหมายเลข 53 สำหรับโดเมนเนมเซอร์วิส (Domain Name Service (DNS)) หมายเลขพอร์ต ถูกจัดแบ่งเป็น 2 ประเภทคือเวลโนว์พอร์ต (Well know Ports) และรีจีสเตอร์พอร์ต (Registered Ports)

2.3.1 เวลโนว์พอร์ต (Well Know Port)

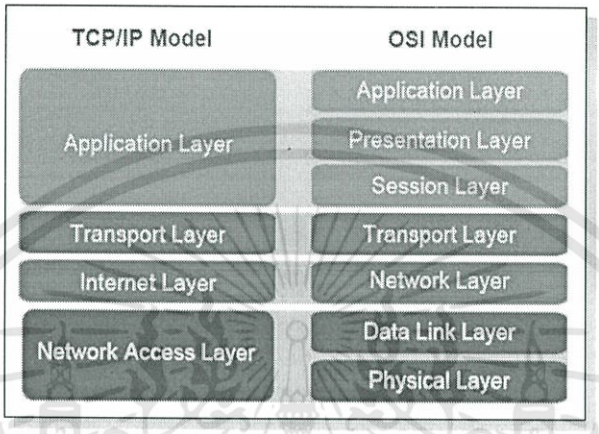
เวลโนว์พอร์ตเรียกว่าพอร์ตบริการ จะเป็นพอร์ตที่ระบบส่วนใหญ่กำหนดให้ใช้โดยผู้ใช้ที่มีสิทธิพิเศษ (Privileged User) โดยพอร์ตเหล่านี้ใช้สำหรับการติดต่อระหว่างเครื่องที่มีระบบเวลาที่ยาวนาน วัตถุประสงค์เพื่อให้บริการแก่ผู้ใช้ที่ไม่รู้จักหรือแปลกหน้า จึงจำเป็นต้องกำหนดพอร์ตติดต่อสำหรับบริการนั้นๆ เวลโนว์พอร์ตมีหมายเลขตั้งแต่ 0-1024 เรียกได้ว่าเป็นพอร์ตประเภทมาตรฐานก็ได้ เช่น HTTP, DNS, Telnet, FTP, SMTP

2.3.2 รีจีสเตอร์พอร์ต (Registered Ports)

รีจีสเตอร์พอร์ตเป็นพอร์ตที่มีหมายเลขตั้งแต่ 1024 ขึ้นไป เป็นพอร์ตที่ใช้ในการติดต่อใดๆที่นอกเหนือจากพอร์ตบริการ

2.4 ทีซีพี/ไอพี (TCP/IP)

TCP/IP (Transmission Control Protocol/Internet Protocal) เป็นโปรโตคอลพื้นฐานของระบบเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ ในเรื่องลำดับชั้นTCP/IP มีรูปแบบการจัดลำดับชั้นที่แตกต่างจาก OSI ดังรูป รูปที่ 2.3



รูปที่ 2.3 ความแตกต่างของ OSI model และ TCP/IP

โปรโตคอล TCP มีกลไกลควบคุมการรับส่งข้อมูลให้มีความถูกต้อง และมีการสื่อสารอย่างมีกระบวนการ (Connection-Oriented) คือต้องมีการตรวจสอบความพร้อมก่อนการรับส่งข้อมูลหรือที่เรียกว่าทรีเวย์แฮนด์เช็ค (Three way handcheck) ก่อนจะทำการแลกเปลี่ยนข้อมูลกัน

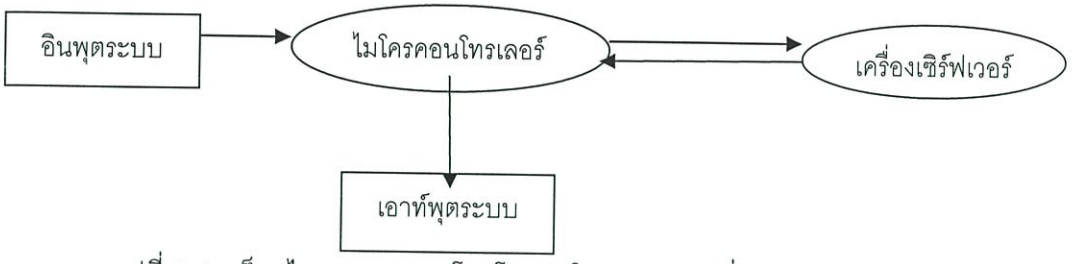
โปรโตคอลไอพี (IP Protocal) เป็นโปรโตคอลทำหน้าที่จัดการเกี่ยวกับแอดเดรสข้อมูลและควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเกต ซึ่งกลไกในการหาเส้นทางของไอพีจะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล

เมื่อโปรโตคอล TCP รับ Application DATA ถูกนำมารวมกับเฮดเดอร์ของโปรโตคอล TCP กลายเป็น TCP Segment จะถูกส่งต่อไปยังชั้นโปรโตคอล IP ถูกนำมารวมกับเฮดเอร์ของโปรโตคอลกลายเป็น TCP/IP Datagram

2.5 โปรโตคอล (Protocal)

โปรโตคอล คือ ข้อตกลงที่ใช้ในการสื่อสารระหว่างกันเพื่อให้อุปกรณ์นั้นมีความเข้าใจในทิศทางเดียวกันละสามารถสื่อสารกันได้ ในระบบเครือข่ายคอมพิวเตอร์อาจจะมีซอฟท์แวร์ (Software) ฮาร์ดแวร์ (Hardware) ที่แตกต่างกัน เมื่อมีการแลกเปลี่ยนข้อมูลจะทำให้เกิดการไม่เข้าใจกันของข้อมูล จึงจำเป็นต้องมีโปรโตคอลเป็นสื่อกลาง

ในที่นี้โปรโตคอลคือข้อตกในการสื่อสารแลกเปลี่ยนระหว่างโปรแกรมในบอร์ดไมโครคอนโทรเลอร์ กับโปรแกรมในเครื่องเซิร์ฟเวอร์ ที่มาจากการกำหนดเงื่อนไขที่ว่า การป้อนอินพุตเข้าไมโคคอนโทรลเลอร์ ให้ข้อมูลเข้าไปยังเครื่องเซิร์ฟเวอร์แล้วเครื่องเซิร์ฟเวอร์ส่งเอาท์พุตเพื่อให้ไมโครคอนโทรเลอร์มายังส่วนแสดงผลของการป้อนอินพุตเข้าไป ดังรูปที่ 2.4



รูปที่ 2.4 บล็อกไดอะแกรมของโปรโตคอลในการแลกเปลี่ยนข้อมูล

จากเงื่อนไขนี้ก็เกิดโปรโตคอลที่เป็นข้อตกลงในการส่งข้อมูลระหว่างไมโครคอนโทรเลอร์กับเครื่องเซิร์ฟเวอร์ ผ่านพอร์ตสื่อสารอนุกรม RS232

2.6 ลินุกซ์ (Linux) และ อูบุนตู (Ubuntu)

2.6.1 ลินุกซ์ (Linux)

เป็นระบบปฏิบัติการเช่นเดียวกับ ดอส ไมโครซอฟต์วินโดวส์ หรือยูนิกซ์ โดยลินุกซ์นั้นจัดว่าเป็นระบบปฏิบัติการยูนิกซ์ประเภทหนึ่ง การที่ลินุกซ์เป็นที่กล่าวขานกันมากขณะนี้ เนื่องจากความสามารถของตัวระบบปฏิบัติการและโปรแกรมประยุกต์ที่ทำงานบนระบบลินุกซ์ โดยเฉพาะอย่างยิ่งโปรแกรมในตระกูลของ GNU (GNU's Not UNIX) และสิ่งที่สำคัญที่สุดก็คือระบบลินุกซ์เป็นระบบปฏิบัติการประเภทฟรีแวร์ (Free Ware) คือไม่เสียค่าใช้จ่ายในการซื้อโปรแกรม

ระบบลินุกซ์ตั้งแต่เวอร์ชัน 4 นั้น สามารถทำงานได้บนซีพียูทั้ง 3 ตระกูล คือบนซีพียูของอิลเทล (PC Intel) ดิจิตอลอัลฟาคอมพิวเตอร์ (Digital Alpha Computer) และซันสปาร์ค (SUN SPARC) เนื่องจากใช้เทคโนโลยีที่เรียกว่า RPM (Red Hat Package Management) ถึงแม้ว่าในขณะนี้ลินุกซ์ยังไม่สามารถแทนที่ไมโครซอฟต์ วินโดวส์บนพีซีหรือแมคโอเอส (Mac OS) ได้ทั้งหมดก็ตาม แต่ผู้ใช้จำนวนไม่น้อยที่หันมาใช้และช่วยพัฒนาโปรแกรมประยุกต์บนลินุกซ์กัน และเรื่องของการดูแลระบบลินุกซ์นั้น ภายในระบบลินุกซ์เองมีเครื่องมือช่วยสำหรับดำเนินการให้สะดวกยิ่งขึ้น

2.6.1.1 ข้อดีของระบบปฏิบัติการลินุกซ์

- เป็นระบบปฏิบัติการที่ใช้งานได้ฟรี ไม่มีค่าลิขสิทธิ์
- ทำงานได้บนเครื่องพีซีทั่วไป ที่มีหน่วยประมวลผลกลางตั้งแต่ 80386 ขึ้นไป รวมถึง Motora 680x0, Compaq (Digital) Alpha, PowerPC, SPARC เป็นต้น จึงเป็นระบบปฏิบัติการที่มีความต้องการทรัพยากรของระบบในขั้นต่ำ
- สามารถทำงานได้รวดเร็ว เนื่องจากมีระบบการจัดการหน่วยความจำเสมือน (Virtual Memory) การจัดทำงานแบบมัลติทาสกิ้ง (Multitasking) และระบบป้องกันการรบกวนการทำงานระหว่างโปรเซส (Process) ต่างๆ
- มีความสามารถแบบ UNIX
- สามารถใช้งานร่วมกับดอส (DOS) และ Microsoft Windows โดยการแบ่งพาติชัน
- เป็นระบบปฏิบัติการแบบเปิด เนื่องจากทุกฟังก์ชันมี Source Code แนบมา ทำให้มีผู้พัฒนาจากทั่วโลกสามารถเข้ามาพัฒนาและแก้ไขข้อบกพร่องของระบบได้ตลอด ช่วยให้ระบบปฏิบัติการลินุกซ์ถูกพัฒนาอย่างต่อเนื่องและมีประสิทธิภาพ

- การติดตั้งระบบปฏิบัติการลินุกซ์จาก CD-Rom/DVD นั้น โปรแกรมแทบจะทุกโปรแกรมที่เราต้องการก็จะถูกติดตั้งลงไปพร้อมๆกัน ไม่เหมือนกับวินโดวส์ ที่จะต้องมานั่งลงทีละโปรแกรม ซึ่งอาจจะใช้เวลาเป็นวันๆ และยังต้องเสียค่าลิขสิทธิ์สำหรับบางโปรแกรมอีก
- รองรับการใช้งานของผู้ใช้หลายๆ คนได้พร้อมๆ กัน หมายความว่าผู้ใช้แต่ละคนสามารถที่จะรีโมทล็อกอิน(remote login) ผ่านโปรแกรมเทลเน็ต (telnet) หรือซีเคียวเชลล์ (secure shell) เพื่อเข้าไปใช้งานเครื่องเซิร์ฟเวอร์ ที่ใช้ระบบปฏิบัติการยูนิกส์ได้หลายๆ คนพร้อมๆ กัน
- ระบบปฏิบัติการลินุกซ์ นั้นมีโปรแกรมแทบจะทุกอย่างให้ใช้ฟรี ซึ่งสามารถทำงานได้ดีพอๆกับโปรแกรมในระบบปฏิบัติการวินโดวส์

โดยสรุป หากคอมพิวเตอร์เครื่องใดสามารถทำงานได้เป็นเวลานานๆ โดยไม่ต้องปิดเปิดเครื่องใหม่ นั่นหมายความว่า ระบบนั้นๆ มีเสถียรภาพลินุกซ์สามารถทำงานได้เป็นปีๆ โดยไม่จำเป็นต้องรีสตาร์ทเครื่องเลย ดังจะเห็นตัวอย่างในเว็บเซิร์ฟเวอร์ (Web Server) ต่างๆ ล้วนแล้วแต่ใช้ลินุกซ์ซึ่งบางเครื่องยังไม่เคยรีสตาร์ทเลยก็มี ยกเว้นในบางกรณีที่ต้องทำการอัพเดทระบบในระดับเชิงลึก ซึ่งนั่นก็เป็นเรื่องปกติที่ต้องรีสตาร์ทระบบ

2.6.2 อูบุนตู (Ubuntu)

อูบุนตู เป็นดิสทริบิวชัน (Distribution) ของลินุกซ์แต่อูบุนตูพัฒนามาจากเดเบียน (Debian) (ซึ่งก็เป็นดิสทริบิวชันของลินุกซ์เช่นกัน) เปรียบเทียบกับวินโดวส์ ก็เช่น WinVista WinXP WinME แต่ลินุกซ์แบ่งการพัฒนาเป็นดิสทริบิวชันออกมาเพื่อการใช้งานทั้งเดสก์ท็อป (Desktop) และเซิร์ฟเวอร์ (Server) เช่น Fedora (Red Hat), SUSE Linux (Novell), Ubuntu (Canonical Ltd.) Mandriva Linux Debian และ Gentoo แต่ละดิสทริบิวชันจะมาจากหลายบริษัทหรือจากที่เดียวกันอูบุนตูก็เป็นชื่อของ OS ที่เป็นลินุกซ์ชนิดหนึ่งก็คล้ายๆกับ Redhat, slackware, SUSE พวกนี้ก็เป็นลินุกซ์ แต่เป็นของแต่ละค่ายแต่ละบริษัท

2.6.2.1 ข้อดีของอูบุนตู

- ฟรี ไม่เสียค่าใช้จ่ายและสามารถใช้งานได้ทุกรูปแบบ โดยไม่มีเงื่อนไขกำหนดระบบการป้องกันอูบุนตู มีการป้องกันโดยให้สิทธิเฉพาะผู้ดูแลระบบเท่านั้นโดยต้องมีการกรอกรหัสผ่านเพื่อยืนยันตัวตนที่แท้จริงก่อนจึงอนุญาตให้ทำงานในระบบได้
- นอกจากการปรับปรุงต่างๆ จนกลายเป็น OS แล้ว ในเรื่องอื่น เช่นหน้าตา หรือสามารถปรับเปลี่ยนได้ ทั้งจะเปลี่ยนหน้าตา หรือการใช้งาน ให้เป็นอย่างที่ต้องการก็ได้
- การใช้ภาษาไทยในอูบุนตูทั้งด้านอ่าน การเขียน หรือเมนูโปรแกรมภาษาไทยใช้งานได้ดี เหมาะสำหรับเด็กๆ หรือผู้สูงอายุ รวมถึงผู้ที่เริ่มใช้งานใหม่ ๆ สามารถใช้งานได้ง่าย
- การใช้ทรัพยากรน้อย ความต้องการฮาร์ดแวร์ต่ำ เครื่องไม่แรงก็สามารถลงได้

2.6.2.2 ข้อเสียของอูบุนตู

- เกี่ยวกับงานด้านเอกสารทั่วไปโดยเฉพาะไทยมักจะใช้ไมโครซอฟท์ ออฟฟิศ (Microsoft Office) ดังนั้นการแลกเปลี่ยนเอกสารใช้งานซึ่งกันและกันอาจจะพบปัญหาความเข้ากันได้ และความผิดเพี้ยน
- ในการเรียกใช้อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) ต้องใช้เวอร์ชัน6.0 เท่านั้น

- การนำ Flash มาใช้ตกแต่งหรือทำเป็นอินเตอร์เฟซ ปัญหาคือ Flash Player จาก Adobe มีประสิทธิภาพเสียมาก ดังนั้นการใช้งานบางเว็บอาจใช้งานไม่ได้ หรือใช้ได้ไม่หรือไม่ก็กิน CPU จนน็อก
- Shortly Support เวอร์ชันที่มีระยะเวลาการใช้งานมากกว่า 18 เดือน จะไม่มีการซับพอร์ท (Support) และอัพเดท (Update)

2.7 การสื่อสารข้อมูลแบบอนุกรม

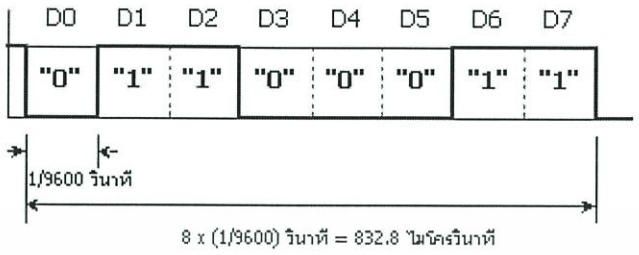
ข้อมูลในไมโครคอนโทรลเลอร์ที่เราใช้ศึกษาอยู่นี้ จะเป็นข้อมูลที่มีความยาวขนาด 1 ไบต์ หรือ 8 บิต ซึ่งโดยปกติถ้าเราจะให้ส่งข้อมูลพร้อมๆกันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดงได้ดังรูป 11ก จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนานจึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสาย และระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูล หรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง (TxD) 1 เส้น สายรับ (RxD) 1 เส้น และสายกราวด์ร่วม (Ground) 1 เส้น) นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ถ้าหากต้องการส่งข้อมูลขนาด 8 บิต ก็จะทำการส่งข้อมูลออกไปทีละบิตเป็นลำดับไป จนกว่าจะครบจำนวนทั้ง 8 บิต ดังในรูป 1ค จะแสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลจะถูกส่งไปตามสายสัญญาณที่ละบิตตามจังหวะเวลาที่กำหนด เป็นความกว้างของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐาน ของฝ่ายส่ง และฝ่ายรับด้วย ในการรับสัญญาณที่ส่งมาทีละบิต จะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "1" หรือ "0" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม

2.7.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม เพื่อรับหรือส่งข้อมูล จะเป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ 9,600 บิตต่อวินาที ดังแสดงในรูปที่ 2.5 ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ 8 x 104.1 หรือ832.8 ไมโครวินาที

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0

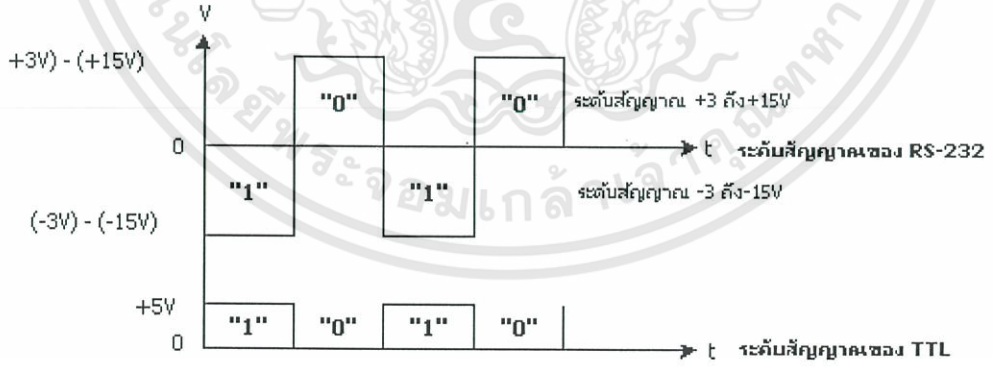


รูปที่ 2.5 การส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

2.7.2 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

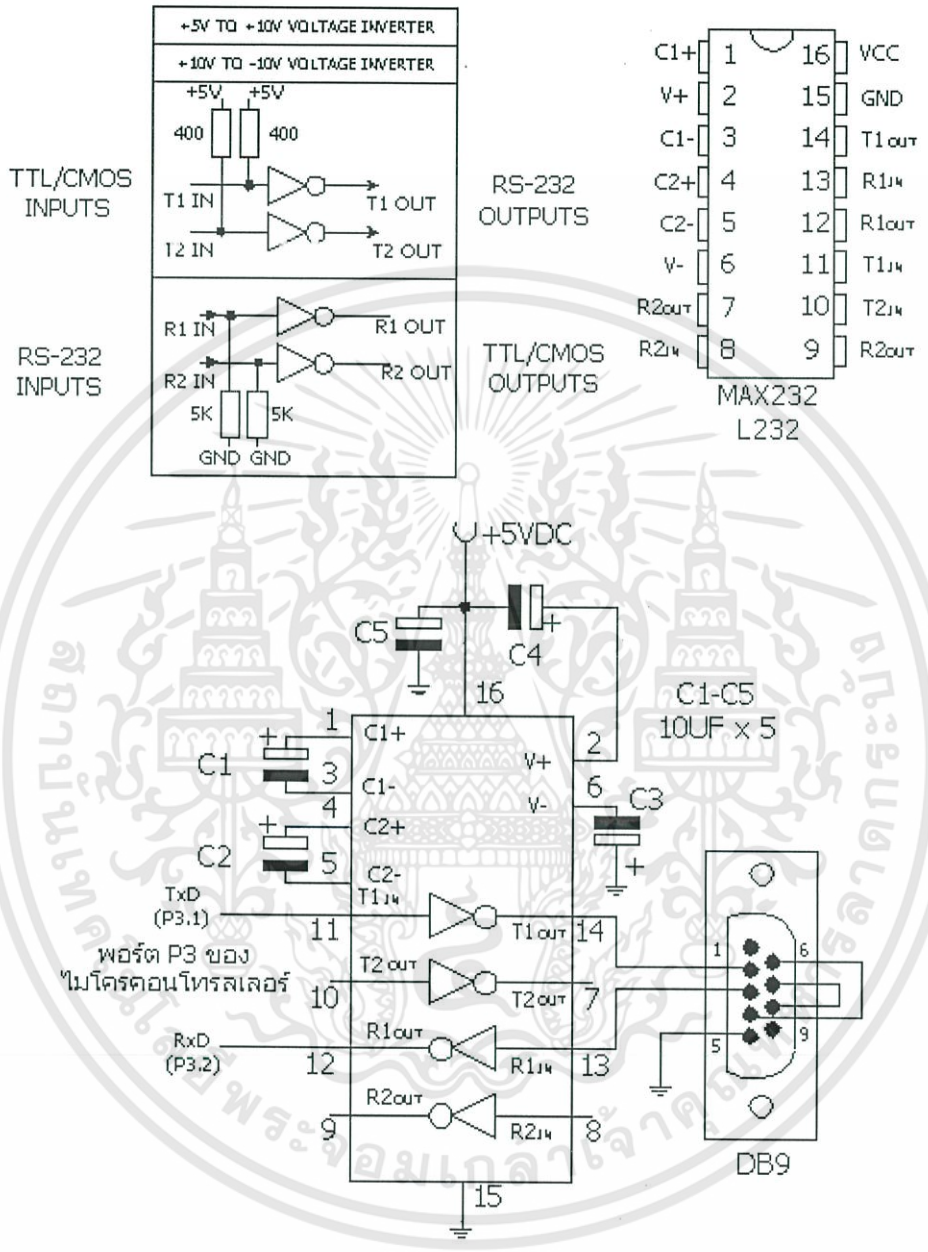
การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232 (x) เป็นมาตรฐานอุตสาหกรรม โดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งานในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุตโดยมีระดับสัญญาณตั้งแต่ 3 โวลต์ จนถึง 15 โวลต์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลต์ จนถึง -15 โวลต์ สำหรับลอจิก "1" ดังแสดงในรูป 2.6

ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิตอลทั่วๆไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 - 5 โวลต์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า - 3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งานเช่น MAX232, L232 เป็นต้น หรืออาจจะต่อวงจรจากทรานซิสเตอร์ได้



รูปที่ 2.6 ระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL

ไอซี MAX232, L232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้ ตำแหน่งขาของไอซี MAX232, L232 และการต่อใช้งานแสดงไว้ดังรูปที่ 2.7



รูปที่ 2.7 ตำแหน่งขาของไอซี MAX232, L232 และการต่อใช้งาน

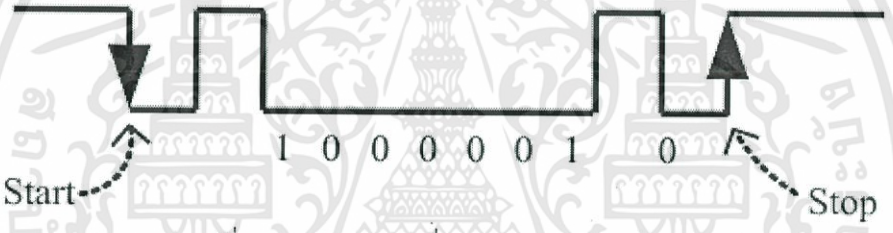
2.7.3 รูปแบบการสื่อสารผ่านพอร์ตอนุกรม

การสื่อสารในเครื่องคอมพิวเตอร์ เป็นการส่งถ่ายข้อมูลระหว่างเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปจะ มีรูปแบบการส่งถ่าย 2 แบบคือแบบขนาน และแบบอนุกรม

2.7.3.1 การสื่อการแบบขนาน เป็นการส่งถ่ายข้อมูลขนาด 8บิต พร้อมๆกัน ดังนั้นจึงต้องมีสายสัญญาณ 8 เส้น ใช้ความเร็วในการส่งถ่ายสูง แต่ไม่เหมาะกับการส่งถ่ายในระยะไกลๆ เพราะจะสิ้นเปลืองสายสัญญาณ

2.7.3.2 การสื่อสารแบบอนุกรม เป็นการสื่อสารครั้งละกลุ่มบิต ดังนั้นจึงใช้สายสัญญาณเพียง 2-3 เส้นเท่านั้น สามารถส่งถ่ายข้อมูลได้ระยะทางไกลกว่าการสื่อสารแบบขนาน แต่ความเร็วจะช้ากว่ามาก การสื่อสารแบบอนุกรมยังมีลักษณะการส่ง 2 แบบคือ

- แบบซิงโครนัส (Synchronous) จะเป็นการส่งข้อมูลไปพร้อมๆกับสัญญาณนาฬิกา เช่น การส่งข้อมูลจากแป้นพิมพ์ของเครื่องคอมพิวเตอร์
- แบบอะซิงโครนัส (Asynchronous) เป็นการส่งข้อมูลแบบใช้เวลาเป็นตัวควบคุมการส่ง ทั้งนี้ต้องขึ้นอยู่กับความพร้อมของภาครับกับภาคส่ง ซึ่งในตัว MCS-51 เป็นการสื่อสารอนุกรมแบบอะซิงโครนัส มีรูปแบบดังรูปที่ 2.8



รูปที่ 2.8 รูปแบบการสื่อสารอนุกรม 1 กลุ่มบิต

จากรูปเป็นการส่งข้อมูลขนาด 8 บิต (1 กลุ่ม) ประกอบไปด้วย

- บิตเริ่มต้น (Start bit) เป็นบิตที่ใช้บอก MCS-51 ตัวรับให้รู้ว่ากาลังจะมีข้อมูลถูกส่งมาถึง
- บิตข้อมูล (Data bit) มีขนาด 7-8 บิต เป็นบิตข้อมูลที่ใช้สื่อสารกันระหว่าง MCS-51 กับซีพียูตัวอื่นๆ (ถ้าเป็นเครื่องคอมพิวเตอร์มักจะเป็นรหัสแอสกี)
- บิตตรวจสอบความผิดพลาด (Parity bit) เป็นบิตที่ใช้ตรวจสอบข้อมูลที่ส่งว่ามีความถูกต้องหรือไม่
- บิตหยุด (Stop bit) ใช้เป็นตัวบอกให้รู้ว่าข้อมูลที่ส่งมานั้นสิ้นสุดแล้ว

2.7.4 มาตรฐาน RS-232

การสื่อแบบอนุกรมสามารถสื่อสารข้อมูลได้ไกล 50 เมตร ทั้งนี้ขึ้นอยู่กับคุณภาพของสายสัญญาณด้วย แต่เพื่อให้เป็นมาตรฐานในการสื่อสารแบบอนุกรมจึงมีข้อกำหนดอยู่ 4 ส่วน

- อัตราความเร็วในการรับ และส่ง (Baud Rate) มีตั้งแต่ 110 จนถึง 76800 เช่น อัตราความเร็วในการับ และส่ง 9600 หมายถึง มีการรับและส่งข้อมูลภายใน 1 วินาทีด้วยข้อมูล 9600 บิต
- ความกว้างของข้อมูล (Data Width) เป็นกลุ่มของข้อมูลที่จะจัดให้มีการรับและส่ง มีขนาด 7-8 บิต

- ค่าพาริตี้ (Parity bit) เป็นบิตที่ใช้ตรวจสอบความถูกต้องของข้อมูลที่รับ หรือส่ง
- บิตจบการสื่อสาร (Stop bit) มีขนาด 1 หรือ 2 บิตใช้บอกว่าเป็นการสิ้นสุดการส่งข้อมูล

บทที่ 3

การออกแบบและการสร้าง

3.1 การออกแบบ

ปัจจุบันทั้งในระดับองค์กรขนาดเล็กหรือตามบ้านนั้นการมีเราเตอร์คงเป็นเรื่องปกติไปแล้ว ซึ่งโดยส่วนใหญ่มีไว้เพื่อการแชร์อินเทอร์เน็ตภายในเครือข่ายของตน เราเตอร์บางตัวมีไฟร์วอลล์มาให้ซึ่งอาจจะช่วยป้องกันการแพร่ไวรัสที่โจมตีผ่านพอร์ตอย่างแบลสเตอร์ (Blaster) หรือแซสเซอร์ (Sasser) ได้ กล่าวได้ว่าไฟร์วอลล์กลายเป็นเรื่องจำเป็นสำหรับยุคปัจจุบันไปแล้ว ผู้จัดทำเคยลองหาข้อมูลไฟร์วอลล์ที่สามารถมาใช้ภายในองค์กรปรากฏว่าไฟร์วอลล์ที่มีรูปแบบเป็นอุปกรณ์สำเร็จรูปมีราคาหลายระดับ เริ่มตั้งแต่หลักหมื่นไปจนถึงหลักล้าน แต่หากเรามีลินุกซ์ที่เป็นฟรีแวร์เป็นเกตเวย์เราสามารถที่จะเปลี่ยนลินุกซ์ให้เป็นไฟร์วอลล์ ได้ด้วยโปรแกรม เน็ตฟิลเตอร์/ไอพีเทเบิล (Netfilter/Iptables) หรือที่เรียกกันสั้นๆ ว่าไอพีเทเบิล

การทำไฟร์วอลล์ในลินุกซ์คือการนำคำสั่งของไอพีเทเบิล มาเรียงลำดับให้เป็นไปตามกฎที่ต้องการ แต่การที่จะเข้าใจถึงกฎการเขียนไอพีเทเบิลเพื่อให้ได้มาซึ่งไฟร์วอลล์นั้นเป็นเรื่องที่ต้องศึกษาและใช้เวลาพอสมควร ในสภาวะที่ต้องการใช้งาน ไอพีเทเบิล โดยเร็ว แต่ความพร้อมยังไม่เต็มร้อย น่าจะมี "ตัวช่วย" บ้างก็คงจะดี ตัวช่วยที่ว่านี้ ในโลกของโอเพนซอร์สมีผู้จัดทำส่วนต่อประสานกับผู้ใช้ (User Interface) ที่ใช้กับไอพีเทเบิลไว้ โดยที่ผู้ใช้ไม่จำเป็นต้องทราบซินเท็ก (Syntax) สำหรับไอพีเทเบิลก็สามารถใช้งานไอพีเทเบิลเป็นไฟร์วอลล์ได้ เพียงแค่กดปุ่มสั่งการตามหน้าจอแสดงผล จากนั้น ผู้ใช้ก็สามารถที่จะใช้เครื่องคอมพิวเตอร์ (หรือเครื่องเซิร์ฟเวอร์อื่นๆ) ที่ใช้ในการแชร์อินเทอร์เน็ตเป็นไฟร์วอลล์ภายในตัวได้ทันที

3.1.1 การออกแบบการทำงานของแอพพลิเคชั่นไฟร์วอลล์

ในหัวข้อนี้จะขอกล่าวถึงภาพรวมของการทำงานของระบบเครื่องข่ายที่ใช้ไอพีเทเบิลเป็นไฟร์วอลล์ว่ามีส่วนประกอบสำคัญใดบ้างให้ได้ก่อน มาซึ่งผลลัพธ์ของข้อมูลที่สามารถนำมาวิเคราะห์และศึกษาต่อไป โดยภาพรวมของระบบที่ยังไม่ได้ออกแบบได้เป็นดังรูปที่ 3.1 และระบบที่ได้รับการออกแบบแล้วเป็นดังรูปที่ 3.2

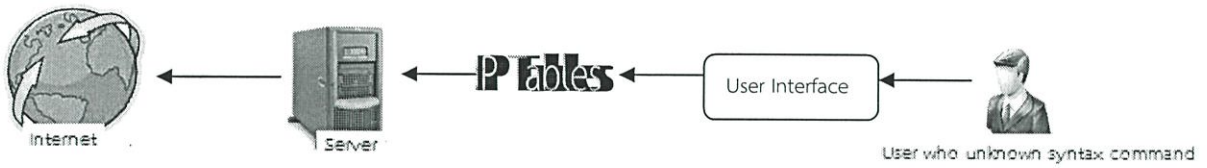
- โครงสร้างของระบบที่ยังไม่ได้ออกแบบ

รูปที่ 3.1 โครงสร้างของระบบที่ยังไม่ได้ออกแบบ

ผู้ใช้จะใช้งานไอพีเทเบิล โดยตรงผ่านหน้าเทอร์มินอล (Terminal) บนลินุกซ์เพื่อกำหนดโพลิซีไฟร์วอลล์สำหรับระบบเครือข่าย

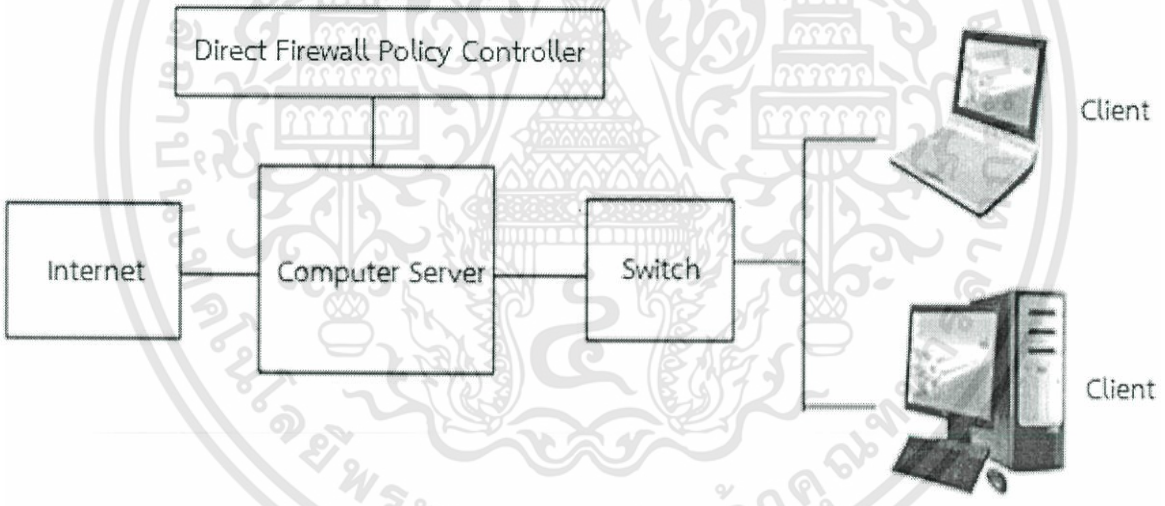
ปัญหาที่เกิดขึ้นผู้ใช้จำเป็นต้องมีความรู้และความเข้าใจในการเขียนคำสั่งสำหรับไอพีเทเบิลต้องเข้าใจในการเขียนซินเท็กคอมมานด์ (Syntax command)

- โครงสร้างของระบบที่ถูกออกแบบแล้ว



รูปที่ 3.2 โครงสร้างของระบบที่ถูกออกแบบ

ผู้ใช้จะสามารถใช้ไอพีเทเบิลเป็นไฟร์วอลล์สำหรับเครือข่ายได้สะดวกง่ายขึ้นโดยไม่ต้องรู้ซินเท็กคอมมานด์ การออกแบบโครงสร้างการทำงานของแอพพลิเคชั่น (Application) คือ การนำแอพพลิเคชั่นไฟร์วอลล์กั้นระหว่างเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ (Computer Server) กับอินเทอร์เน็ต (Internet) แอพพลิเคชั่นไฟร์วอลล์ที่ออกแบบขึ้นจะเป็นเสมือนประตูที่กั้นระหว่างระบบเครือข่ายของคอมพิวเตอร์กับระบบเครือข่ายอื่น ซึ่งจะยอมให้ผ่านประตูได้ก็ต่อเมื่อระบบเครือข่ายอื่นที่มาคอนเนคชั่นกับคอมพิวเตอร์มีความถูกต้องตามเงื่อนไขหรือโพลิซี (Policy) ที่ผู้ใช้ได้กำหนดไว้ ดังแสดงในบล็อกไดอะแกรมดังรูปที่ 3.3



รูปที่ 3.3 บล็อกไดอะแกรมการทำงานของระบบ

ภายในคอมพิวเตอร์จะทำงานภายใต้ระบบปฏิบัติการลินุกซ์ อูบุนตู ซึ่งมีไอพีเทเบิลเป็นเซอร์วิสที่ให้บริการบนลินุกซ์ อูบุนตู การทำงานของแอพพลิเคชั่นไฟร์วอลล์จะเริ่มต้นเมื่อชุดคำสั่งภาษาจาวารับข้อมูลที่ถูกส่งมาจากบอร์ดไมโครคอนโทรลเลอร์ โดยชุดคำสั่งดังกล่าวจะพิจารณาคำสั่งข้อมูลดังกล่าวและทำงานสองส่วน ส่วนที่หนึ่ง จะนำข้อมูลที่ได้รับไปเทียบกับเคสต่างๆที่ได้ประกาศไว้ในชุดคำสั่ง ดังรูปที่ 3.4

```
case "h":       menuPointer="h"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "AcI":     menuPointer="AcI"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "aIP":     menuPointer="aIP"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "aMK":     menuPointer="aMK"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "aPO":     menuPointer="aPO"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "aPT":     menuPointer="aPT"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "aAC":     menuPointer="aAC"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "1":       menuPointer="1"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "2":       menuPointer="2"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "21-":     menuPointer="21-"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "22-":     menuPointer="22-"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "221":     menuPointer="221"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "2-1":     menuPointer="2-1"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "2-2":     menuPointer="2-2"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "2-3":     menuPointer="2-3"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "3":       menuPointer="3"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "3-1":     menuPointer="3-1"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "3-2":     menuPointer="3-2"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "3-3":     menuPointer="3-3"; sendCMD("sudo echo -e '1!^@' >> /dev/ttyUSB0"); break;
case "4":       menuPointer="4";
```

รูปที่ 3.4 โค้ดของชุดคำสั่งภาษาจาวาบางส่วนของเคสต่างๆ

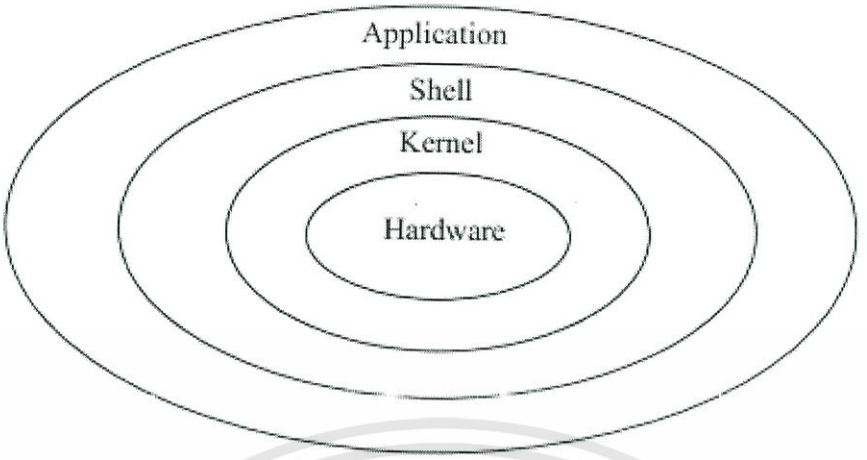
จากรูป โค้ดของชุดคำสั่งภาษาจาวาจะแบ่งเป็นเคสต่างๆ เช่น เมื่อได้รับเลข 1 มาก็จะนำไปพิจารณาเคสที่ 1 ภายในเคสที่ 1 จะมีข้อความต่างๆเก็บไว้ ข้อความดังกล่าวจะถูกส่งผ่าน RS232 กลับไปยังบอร์ดไมโครคอนโทรลเลอร์ โดยจะแบ่งข้อมูลที่จะส่งออกไปเป็นเพย์โหลด(Payload) โดยกำหนดไว้ว่าข้อมูลที่ได้ส่งออกไปใน 1 เพย์โหลด มีส่วนของเฮดเดอร์ (header) จะเป็นเลข “1-4” หรือ “a-d” ตามด้วยเครื่องหมาย “!” ถัดจากนั้นจะเป็นข้อมูลจำนวน 20 คาแรกเตอร์ (character) และปิดท้ายด้วยเครื่องหมาย “^@”เพื่อให้ทราบว่าจบข้อมูล ดังรูปที่ 3.5

(1-4, a-d)!	ข้อมูล 20 Character	^@

รูปที่ 3.5 ลักษณะของ เพย์โหลด

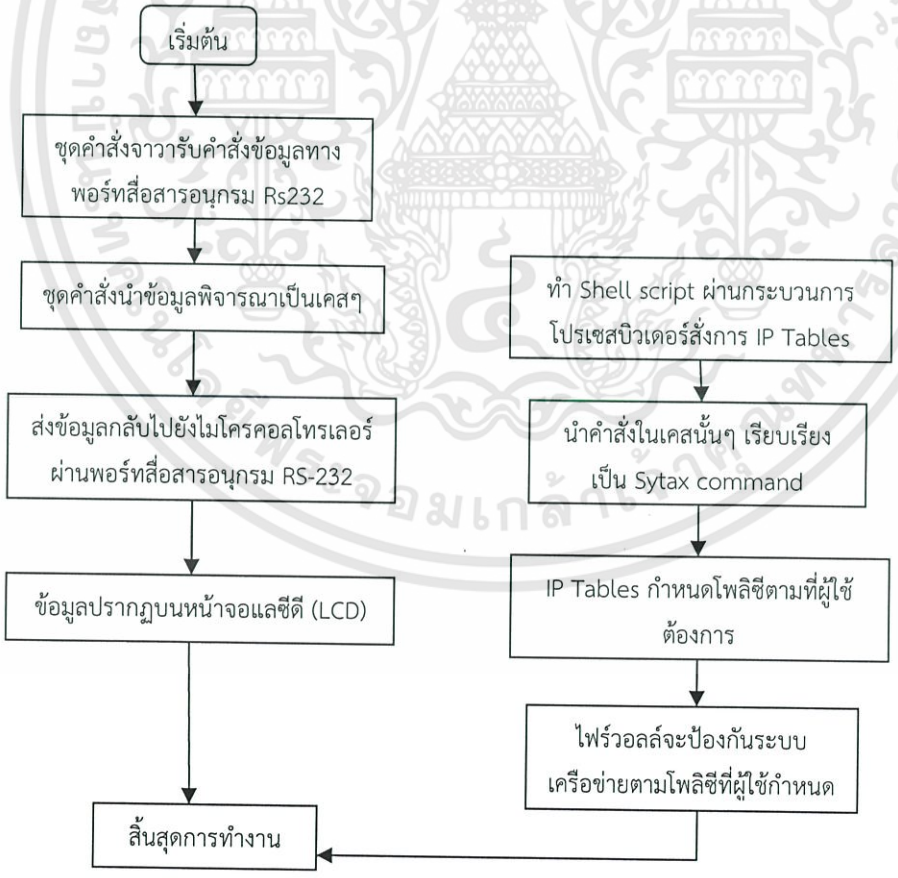
ข้อมูลของเพย์โหลดนี้ที่ตำแหน่งเฮดเดอร์จะบอกให้ไมโครคอนโทรลเลอร์ทราบว่าจะต้องนำข้อมูลนี้ไปแสดงที่บรรทัดใดและส่วนใดในหน้าจอแอลซีดี (LCD) เนื่องจากข้อมูลที่ส่งมาจากคอมพิวเตอร์เครื่องเซิร์ฟเวอร์นั้นมีมากกว่า 20 คาแรกเตอร์ ดังนั้นจึงกำหนดขึ้นว่า “<” แปลว่ายังมีข้อมูลอีกที่จะถูกส่งมา ทางฝั่งของไมโครคอนโทรลเลอร์จะตรวจสอบและรอรับข้อมูลดังกล่าวจนกว่าจะครบ จะมีการ“e!^@”เพื่อให้ทราบว่าหมดเพย์โหลดแล้ว และจะส่ง “end@” เพื่อจบการส่งข้อมูล

ส่วนที่สอง เนื่องจากชุดคำสั่งภาษาจาวาที่ได้พัฒนาขึ้นนั้นเป็นการพัฒนาในระดับแอพพลิเคชั่นตามโครงสร้างพื้นฐานการทำงานของระบบปฏิบัติการลินุกซ์ การที่จะสั่งลินุกซ์ทำงานได้ต้องใช้การทำงานในระดับชั้นของเชลล์ (Shell) ซึ่งทำหน้าที่เป็นอินเตอร์เฟซ (interface) ระหว่างผู้ใช้กับเคอร์เนลดังนั้นจึงมีการใช้โปรเซสบิวเดอไลบารี่ในการสร้างเชลล์สคริปต์ (shell script) เพื่อใช้สื่อกลางในการใช้งานไอพีเทเบิลในระดับชุดคำสั่งของแอพพลิเคชั่นที่จาวาเข้าใจ ดังรูปที่ 3.6



รูปที่ 3.6 โครงสร้างพื้นฐานการทำงานของระบบปฏิบัติการลินุกซ์

ดังนั้นในส่วนที่สองจะมีคำสั่งกำกับไว้ว่าให้นำคำสั่งข้อมูลที่ส่งมาเรียบเรียงเพื่อสร้างเป็นชินเท็กคอม-มานด์และนำชินเท็กคอมมานด์ที่ได้ไปสร้างเป็นเชลล์สคริปต์ส่งไปสั่งการเคอร์เนล โดยผ่านกระบวนการโปรเซสบิวเดอไปเป็นภาษาที่ไอพีเทเบิลเข้าใจเพื่อให้ไอพีเทเบิลเขียนคอมมานด์ (command) ในการกำหนดโพลิซีของไฟร์วอลล์ตามที่ผู้ใช้ต้องการให้กับระบบเครือข่าย การทำงานของแอพพลิเคชั่นไฟร์วอลล์ดังรูปที่ 3.7



รูปที่ 3.7 โฟลว์ชาร์ตการทำงานของแอพพลิเคชั่นไฟร์วอลล์

ผู้จัดทำได้นำคำสั่งของไอพีเทเบิลด้วยกันทั้งหมด ดังนี้

- A ใช้สำหรับเพิ่มโพลิซี
- D ใช้สำหรับลบโพลิซี
- R ใช้สำหรับแทนที่โพลิซีเก่าด้วยโพลิซีใหม่
- p ใช้สำหรับกำหนดโปรโตคอลในโพลิซีนั้น
- s ใช้สำหรับกำหนด Source ip/subnet mark
- sport ใช้สำหรับกำหนด Source port
- d ใช้สำหรับกำหนด Destination ip/subnet mark
- dport ใช้สำหรับกำหนด Destination port/subnet mark
- j ใช้สำหรับกำหนดให้ Policy นั้นๆ ทำการ Accert Reject หรือ drop Packet

3.1.2 การออกแบบโครงสร้างภายในของตัวโปรแกรม

แอพพลิเคชั่นไฟร์วอลล์ที่ผู้จัดทำได้พัฒนาขึ้นมานั้นได้หยิบเอาไอพีเทเบิลเป็นโครงของโปรแกรม แล้วใช้ภาษาจาวาโปรเซสบิวเดอค่าพารามิเตอร์ต่างๆที่ได้รับจากผู้ใช้ กำหนดไปเป็นภาษาที่ไอพีเทเบิลเข้าใจ ซึ่งการที่จะให้โปรแกรมทำงานได้ จึงต้องมีการกำหนดตัวแปรจาวาขึ้นมาเพื่อให้คำสั่งไอพีเทเบิล (เป็นตัวแปรประเภทสตริงซ์) นำตัวแปรที่กำหนดมาเชื่อมสตริงซ์ จากนั้นกำหนดเงื่อนไขวิเคราะห์เพื่อเขียนโปรแกรมให้ผลลัพธ์ตามที่ผู้ใช้ต้องการ โดยในการเขียนโปรแกรมจะเน้นในส่วนของอินพุตที่ป้อนเข้ามาว่า ถ้าอินพุตมีค่าเข้ามาเป็นแบบนี้แล้วโปรแกรมจะทำแบบไหน ซึ่งอินพุตจะถูกแบ่งเป็น 2 ส่วนคือ

1. เป็นแบบอินดิเพนเดนท์ (Independent) คือเป็นอินพุตที่ไม่สามารถเปลี่ยนค่าได้ ได้แก่

- ประเภทของเส้นทางของแพ็กเกต
- ประเภทของโปรโตคอล เช่น TCP, UDP, ALL
- ประเภทของแอ็คชั่น (Action) ได้แก่ Accept, Reject, Drop

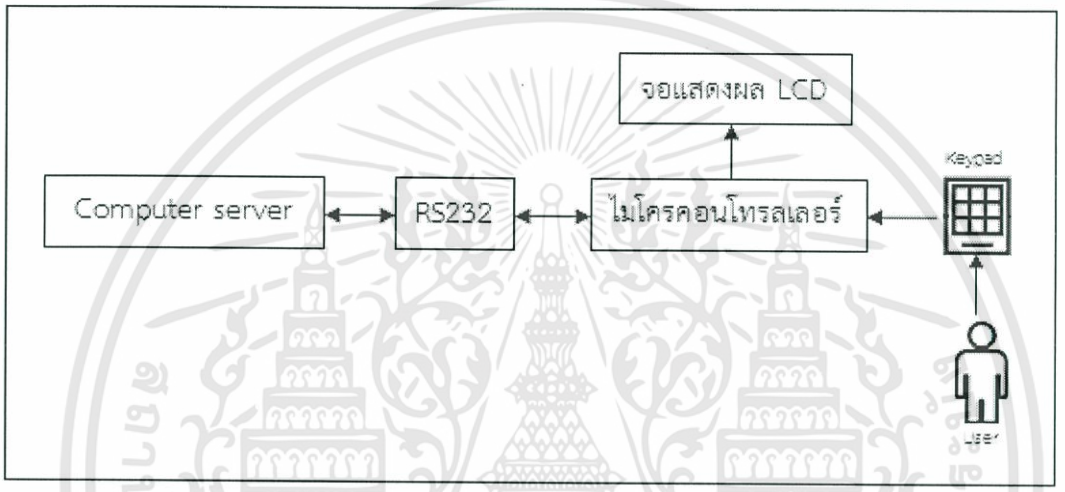
2. แบบดีเพนเดนท์ (Dependent) คือส่วนของอินพุตที่ขึ้นอยู่กับความต้องการใช้งานของยูสเซอร์ ได้แก่ Source IP, Destination IP, Port Number

- อินคัมมิ่งแพ็กเกต (Incomimg Packet) จากเน็ตเวิร์คอื่นๆ (Other Network) ถูกส่งมายังเครื่องเซิร์ฟเวอร์เป็นเมลออกไปยังเครือข่ายอื่น
- เอาท์คัมมิ่งแพ็กเกต (Outgoing Packet) จากเครื่องเซิร์ฟเวอร์ถูกส่งออกไปยังเครือข่ายอื่น เช่น เครื่องเซิร์ฟเวอร์ส่งเมลออกไปยังเครือข่ายอื่น
- ฟอร์เวิร์ดแพ็กเกต (Forward Packet) จะถูกส่งออกไปภายนอกจากไคลแอนท์ (Client) ที่อยู่ภายใต้ Main Network, Source IP, Destination IP
- ไบไดเรคชั่น (Bi-Direction) หมายถึงแพ็กเกตจากเน็ตเวิร์คอื่นๆเข้ามายังเครื่องเซิร์ฟเวอร์และแพ็กเกต จากเครื่องเซิร์ฟเวอร์ ถูกส่งไปยังเน็ตเวิร์คนั้น

ในเชิงโปรแกรมได้กำหนดไดเรคชัน (Direction) ไว้เป็น Forward เมื่อทำการป้อนอินพุตเข้าไปอินพุต เหล่านี้จะมีค่าพารามิเตอร์รองรับ เมื่อกดปุ่ม ENT ที่คีย์แพด (Keypad) โปรแกรมจะเริ่มทำงาน ในเชิงโปรแกรม เมื่อกดปุ่ม ENT บนคีย์แพดจะมีโค้ดที่ใช้ในการเชื่อมอินพุตเหล่านี้เป็นคำสั่งที่ใช้สั่งการได้ แต่การจะให้ไอพีเทเบิล และทำงานโดยรับคำสั่งเหล่านี้จะต้องมีคำสั่งภาษาจาวาที่อ้างสิทธิ์การเปิดไอพีเทเบิลไปสั่งเปิดไอพีเทเบิลขึ้นมา ซึ่งโค้ดทั้งสองจะถูกส่งออกไปพร้อมกันไอพีเทเบิลจะทำงานตามค่าที่ได้รับ ผลที่ได้คือแพ็กเกตจะถูกส่งออกไปตามเส้นทางที่ผู้ใช้กำหนด กล่าวคือ ค่าอินพุตที่ป้อนโดยผู้ใช้นั่นเอง

3.1.3 บล็อกไดอะแกรมการทำงานของอุปกรณ์โดยรวม

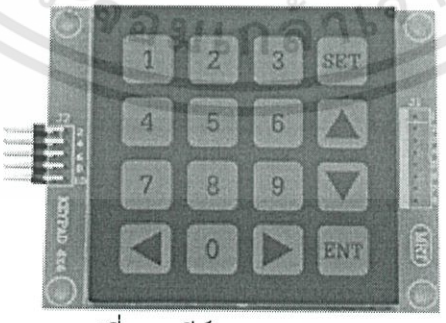
สำหรับปริญญานิพนธ์นี้ ตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง (Direct Firewall Policy Controller) ประกอบไปด้วยส่วนของไมโครคอนโทรลเลอร์ MCS-51 ซึ่งมีคีย์แพดและหน้าจอแอลซีดีเชื่อมต่อเอาไว้ ไมโครคอนโทรลเลอร์จะเชื่อมต่อเข้ากับเครื่องเซิร์ฟเวอร์ด้วย RS232 มีหลักการการงานดังนี้ ไมโครคอนโทรลเลอร์จะรับคำสั่งจากผู้ใช้ผ่านทางคีย์แพดซึ่งเลือกคำสั่งต่างๆ โดยดูจากหน้าจอแอลซีดี จากนั้นส่งคำสั่งไปยังเครื่องเซิร์ฟเวอร์ผ่านพอร์ตสื่อสารอนุกรม RS232 เครื่องเซิร์ฟเวอร์ จะพิจารณาคำสั่งที่ได้รับ เพื่อกระทำ หนึ่ง สั่งค่ากลับไปยังไมโครคอนโทรลเลอร์เพื่อให้หน้าจอแอลซีดีแสดงผล สอง นำคำสั่งที่ได้รับมาเรียงเป็นซินเท็กคอมมานด์และนำซินเท็กคอมมานด์ทั้งหมดไปโปรเซสบิวเดอให้ไอพีเทเบิลเข้าใจเพื่อกำหนดโพลิซีไฟร์วอลล์ให้กับระบบเครือข่าย ดังรูปที่ 3.8



รูปที่ 3.8 บล็อกไดอะแกรมการทำงานของตัวควบคุมโพลิซีไฟร์วอลล์โดยตรง

3.1.4 ออกแบบการทำงานของบอร์ดไมโครคอนโทรลเลอร์

บอร์ดไมโครคอนโทรลเลอร์ใช้ MSC-51 AT89C51RE2 ภายในจะบรรจุชุดคำสั่งภาษาซีที่ผ่านการบิวด์ (build) ให้เป็นเฮกไฟล์ (.Hex) ซึ่งประกอบด้วยส่วนของคำสั่งสแกนคีย์สำหรับให้คีย์แพดทำงานได้ตามที่กำหนด ใช้คีย์แพดแบบ 4*4 ดังรูปที่ 3.9



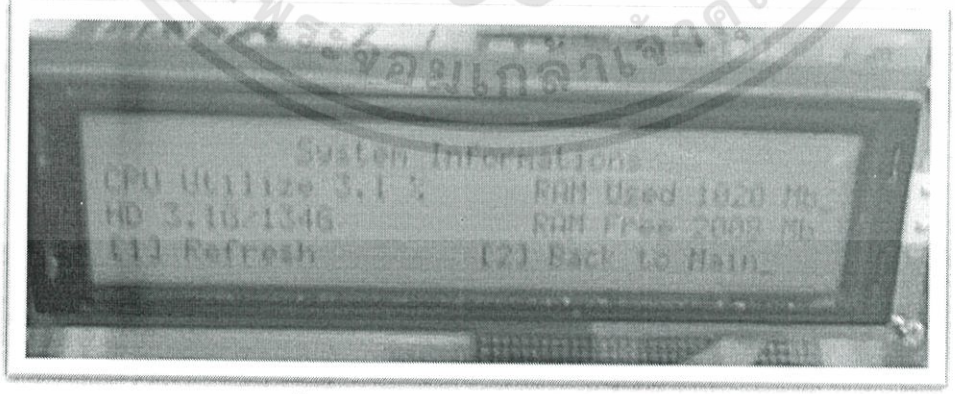
รูปที่ 3.9 คีย์แพดแบบ 4*4

จะใช้สัญญาณควบคุมเพียง 7 เส้น โดยที่การต่อสัญญาณคีย์สวิตช์เมตริกซ์นั้นจะใช้แบบพูลอัพ (Pull-up) สัญญาณ สถานะของสวิตช์จะมีลอจิก 1 หรือ High ทั้งหมด เชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ที่ Port 1 จากนั้นกำหนดค่าของหลัก (Column) ที่ต้องการอ่านค่าก่อนโดยกำหนดให้เป็นลอจิก 0 หรือ Low เพราะหลักจะเป็นขาสัญญาณควบคุม เมื่อเขียนโปรแกรมและอ่านค่าจากแถว (Row) ทั้งหมด โดยหากแถวใดมีการเปลี่ยนแปลง แสดงว่าแถวนั้นมีการกดคีย์สวิตช์ ทำให้ทราบว่าคีย์สวิตช์ตำแหน่งใดมีการกดเกิดขึ้น การสแกนคีย์บอร์ดได้กำหนดตำแหน่งที่จะสแกนในคอลัมน์และอ่านค่าคีย์จากแถว โดยจะรู้ตำแหน่งของคอลัมน์เนื่องจากได้กำหนดด้วยซอฟต์แวร์ จากนั้นก็อ่านค่าแถวกลับมา หากแถวใดมีการเปลี่ยนแปลงจะสามารถรู้ตำแหน่งของสวิตช์ที่ถูกกด เพราะค่าประจำตำแหน่งจะแตกต่างกัน จากนั้นจะเป็นส่วนของชุดคำสั่งสำหรับการส่งข้อมูลที่ได้รับจากการกดคีย์แพดไปยังเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ผ่านพอร์ตสื่อสารอนุกรม RS232 ในส่วนนี้เมื่อเริ่มต้นใช้งานผู้ใช้จะกด ENT หน้าจอแอลซีดี จะปรากฏฟังก์ชันให้ผู้ใช้เลือก ดังรูปที่ 3.10-3.16

รูปที่ 3.10 ข้อความที่ปรากฏให้ผู้ใช้ได้เลือกฟังก์ชัน

จากรูป จะมีฟังก์ชันปรากฏขึ้นมาให้ผู้ใช้ได้เลือกด้วยกัน 4 ฟังก์ชัน ประกอบด้วย

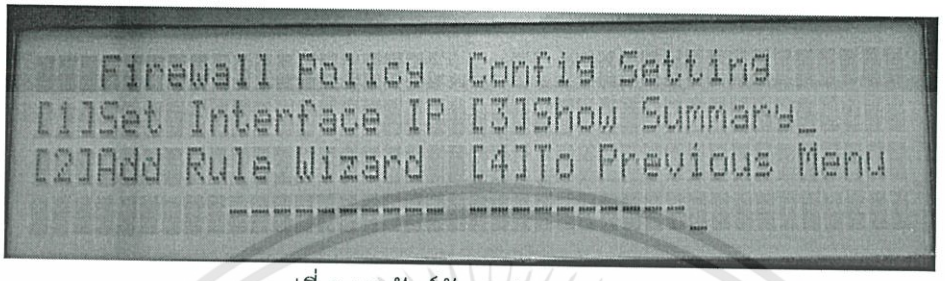
- ฟังก์ชัน Show system info เป็นฟังก์ชันออกแบบไว้สำหรับบอกถึงข้อมูลของระบบ
- ฟังก์ชัน Policies setting เป็นฟังก์ชันออกแบบไว้สำหรับเซ็ตโพลิซีให้กับไฟร์วอลล์
- ฟังก์ชัน Chassis control เป็นฟังก์ชันออกแบบไว้สำหรับการควบคุมพื้นฐานให้กับเครื่องเซิร์ฟเวอร์
- ฟังก์ชัน Show Policies เป็นฟังก์ชันออกแบบไว้สำหรับแสดงโพลิซี



รูปที่ 3.11 ฟังก์ชัน Show system Info

จากรูป เมื่อกดปุ่มบนคีย์แพดเพื่อเลือกฟังก์ชัน Show system Info ข้อมูลพื้นฐานของ เครื่องเซิร์ฟเวอร์จะแสดงผลขึ้นมา จะประกอบด้วย

- CPU Utilize จะบอกถึงการใช้ประสิทธิภาพของซีพียู (CPU) โดยจะบอกเป็นเปอร์เซ็นต์
- HD จะบอกถึงจำนวนฮาร์ดดิสก์ที่ใช้งานต่อจำนวนฮาร์ดดิสก์ที่เหลือให้ใช้งาน
- Ram Used และ Ram free จะบอกถึงจำนวนแรมที่ใช้งานและจำนวนแรมที่เหลือให้ใช้งาน

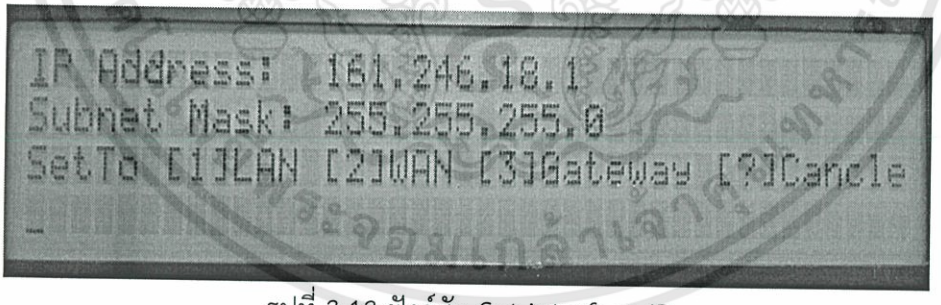


รูปที่ 3.12 ฟังก์ชัน Policies Setting

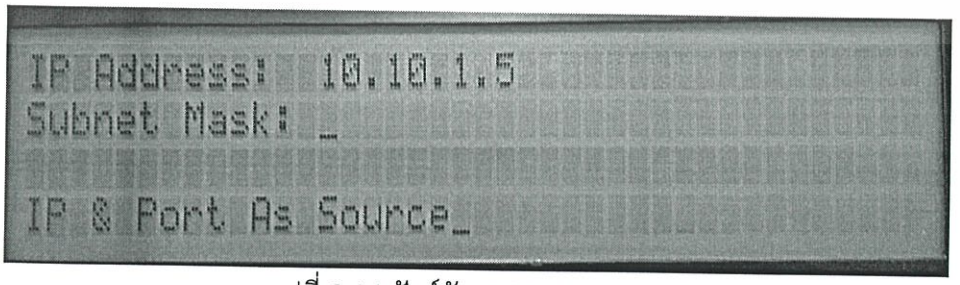
จากรูป ฟังก์ชัน Policies setting จะประกอบด้วย

- Set interface IP ใช้สำหรับการตั้งค่าการ์ดแลนของเครื่องเซิร์ฟเวอร์ (Eth0 Eth1 Gateway) โดยกำหนดไอพี (IP) ให้กับการ์ดแลนเพื่อให้เป็นทางเข้าและออกของระบบเครือข่าย
- Add rule wizard ใช้สำหรับการกำหนดโพลิซีให้กับไฟร์วอลล์ โดยไม่ต้องบังคับซินเท็กคอม-มานด์ของไอพีเทเบิลผู้ใช้เพียงแค่ป้อนค่าตามที่โปรแกรมร้องถาม ตัวชุดคำสั่ง จาวาจะรับค่าและนำไปเรียบเรียงซินเท็กให้เอง
- Show Summary ใช้สำหรับแสดงอินเตอร์เฟซที่เรากำหนดไว้ (Eth0 Ehh1 Gateway) และใช้ในการแก้ไขอินเตอร์เฟซต่างๆ

ฟังก์ชันย่อยของฟังก์ชัน Policies setting แสดงดังรูปที่ 3.13-3.15



รูปที่ 3.13 ฟังก์ชัน Set interface IP



รูปที่ 3.14 ฟังก์ชัน Add rule wizard

รูปที่ 3.15 ฟังก์ชัน Show Summary

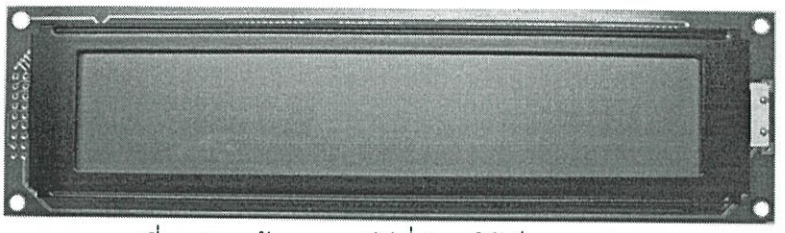
รูปที่ 3.16 ฟังก์ชัน Chassis Control

จากรูป ในฟังก์ชัน Chassis Control นั้นเป็นควบคุมคำสั่งพื้นฐานสำหรับควบคุมเครื่องเซิร์ฟเวอร์ ประกอบด้วย

- Reboot system ใช้สำหรับรีสตาร์ทเครื่องเซิร์ฟเวอร์
- Factory default ใช้สำหรับเคลียร์โพลิซีทั้งหมดทิ้ง
- Shutdown system ใช้สำหรับปิดเครื่องเซิร์ฟเวอร์

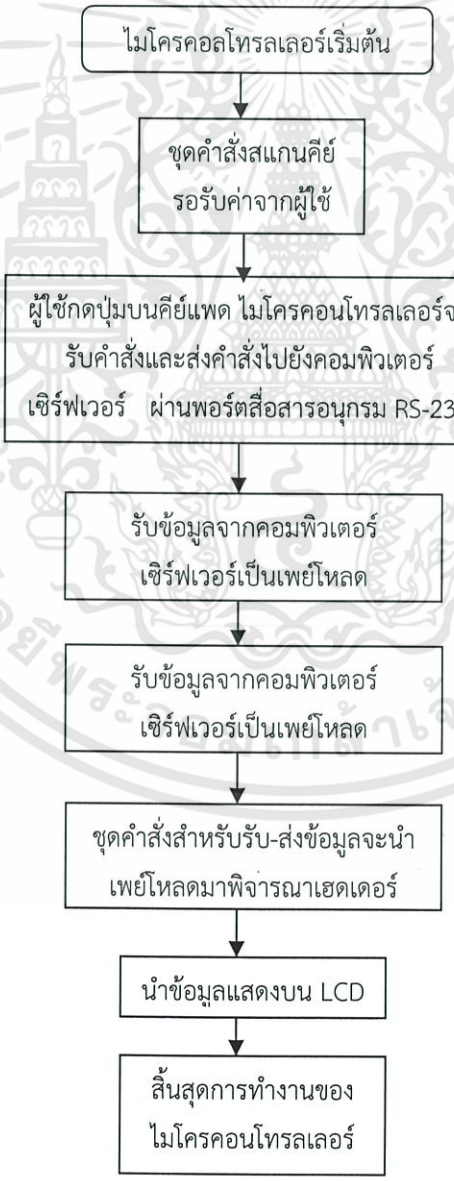
ผู้ใช้จะต้องกดปุ่ม 1-4 เพื่อเลือกฟังก์ชันต่างๆ ซึ่งไม่ว่าผู้ใช้จะกดปุ่มตัวเลขใดก็ตามชุดคำสั่งในส่วนนี้ก็จะนำค่านั้นแปลงเป็นเลขฐาน 2 ส่งผ่านพอร์ตสื่อสารอนุกรม RS232 ไปยัง เครื่องเซิร์ฟเวอร์

เครื่องเซิร์ฟเวอร์ จะรับเอาค่านั้นไปเปรียบกับชุดคำสั่งจาวา ซึ่งจะอธิบายต่อไปนี้ในหัวข้อถัดไปในส่วนของชุดคำสั่งจาวา ชุดคำสั่งจาวาจะส่งค่ากลับไปตามที่ได้กำหนดไว้ว่าจะต้องส่งข้อความใดกลับไปเมื่อได้รับเลขนี้ โดยส่งกลับมายังไมโครคอนโทรลเลอร์ผ่านทางพอร์ตสื่อสาร RS232 จากนั้นไมโครคอนโทรลเลอร์จะรับเอาข้อความที่ได้รับไปแสดงผลทางหน้าจอแอลซีดี ส่วนในกรณีที่ผู้ใช้ต้องกดปุ่มคีย์แพดเพื่อใส่ค่าไอพีแอดเดรส ตัวเลขที่ปรากฏบนหน้าจอแอลซีดี จะแสดงผลโดยไมโครคอนโทรลเลอร์และในขณะเดียวกันก็จะถูกส่งค่านั้นๆไปยังฝั่งของเครื่องเซิร์ฟเวอร์ ผ่านทางพอร์ตสื่อสารอนุกรม RS232 แบบเรียลไทม์ (Realtime) อีกด้วย และส่วนสุดท้ายจะเป็นชุดคำสั่งรับค่าจากฝั่งของเครื่องเซิร์ฟเวอร์โดยจะรับมาเป็นเพย์โหลดจะมีการกำหนดไว้แล้วเป็นข้อตกลงระหว่างเครื่องเซิร์ฟเวอร์กับไมโครคอนโทรลเลอร์ ถึงแม้ว่าทางฝั่งของเครื่องเซิร์ฟเวอร์จะเป็นจาวาโค้ดส่วนไมโครคอนโทรลเลอร์เป็นซีโค้ดซึ่งต่างกันทางภาษาคอมพิวเตอร์แต่ก็สามารถสื่อสารกันได้เพราะเป็นการสื่อสารในระดับข้อมูลนั้นเอง ซึ่งหน้าจอแอลซีดีที่นำมาใช้เป็นแบบ 40*4 ดังรูปที่ 3.17



รูปที่ 3.17 หน้าจอแอลซีดีที่นำมาใช้เป็นแบบ 40*4

ไมโครคอนโทรลเลอร์เมื่อได้รับเพย์โหลดมาแล้วจะพิจารณาคาแรคเตอร์ถัดมาจากเฮดเดอร์ซึ่งเป็นข้อมูลที่ต้องนำไปแสดง โดยเมื่อตรวจพบเครื่องหมาย “^@” ก็จะหยุดและตัดเอาเฉพาะคาแรคเตอร์ ก่อนหน้านั้น สิ่งที่ได้ก็คือ ข้อมูลจำนวน 20 คาแรคเตอร์นั้นเอง เนื่องจากข้อมูลที่ส่งมาจากเครื่องเซิร์ฟเวอร์นั้นมีมากกว่า 20 คาแรคเตอร์ดังนั้นจึงกำหนดขึ้นว่า “>” แปลว่า ยังมีข้อมูลอีกที่จะถูกส่งมาทางฝั่งไมโครคอนโทรลเลอร์จะตรวจสอบและรอรับข้อมูลดังกล่าวจนกว่าจะครบ เมื่อได้รับ “end@” เพื่อจบการส่งข้อมูล การทำงานในส่วนของไมโครคอนโทรลเลอร์ ดังรูปที่ 3.18



รูปที่ 3.18 โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์

3.2 อุปกรณ์ที่ใช้ในการทดลอง

อุปกรณ์ที่ใช้ในการทดลองประกอบไปด้วย

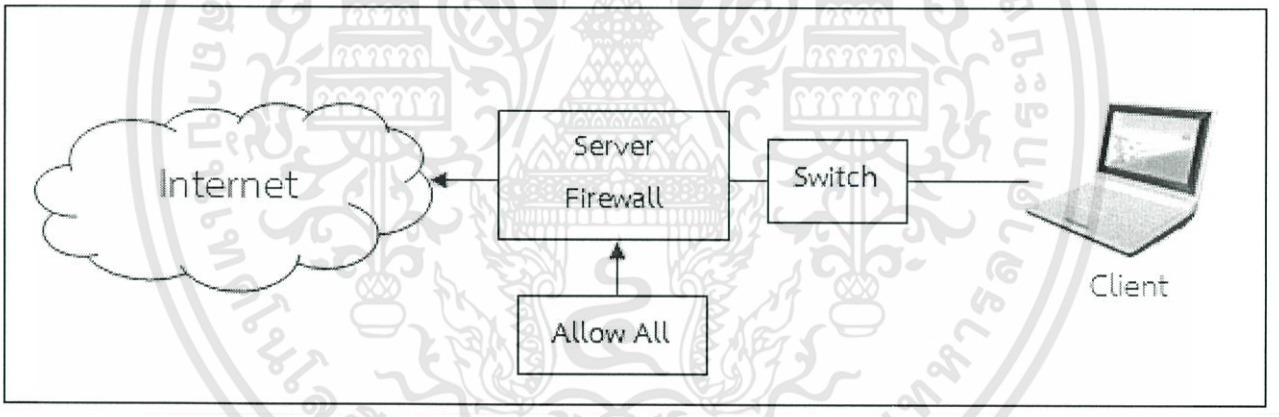
1. บอร์ดไมโครคอนโทรลเลอร์
2. เครื่องเซิร์ฟเวอร์
3. พาวเวอร์ซัพพลาย (Power supply)
4. ตัวแปลง USB to RS232
5. สาย RS232

3.3 การจัดเก็บผลการทดลอง

ในการจัดเก็บผลการทดลอง จะทดสอบการทำงานของแอพพลิเคชั่นไฟร์วอลล์ให้ทำงานตามโพลิซีที่กำหนดให้ผ่านตัวอุปกรณ์ และดูผลทดสอบผ่านการใช้งานเครือข่ายของไคลแอนต์และใช้โปรแกรมตรวจสอบควบคู่กัน โดยการทดสอบการทำงานของแอพพลิเคชั่นมีดังนี้

3.3.1 ทดสอบการทำงานของอุปกรณ์เพื่อเริ่มต้นใช้งานระบบเครือข่าย

ทดสอบระบบเครือข่ายสามารถใช้งานอินเทอร์เน็ตได้ โดยใช้ตัวอุปกรณ์เซ็ตอินเตอร์เฟซและป้อนค่าโพลิซีไฟร์วอลล์โดยกำหนดให้ไคลเอนต์สามารถใช้งานอินเทอร์เน็ตได้ (Allow all) บล็อกไดอะแกรมดังรูปที่ 3.19

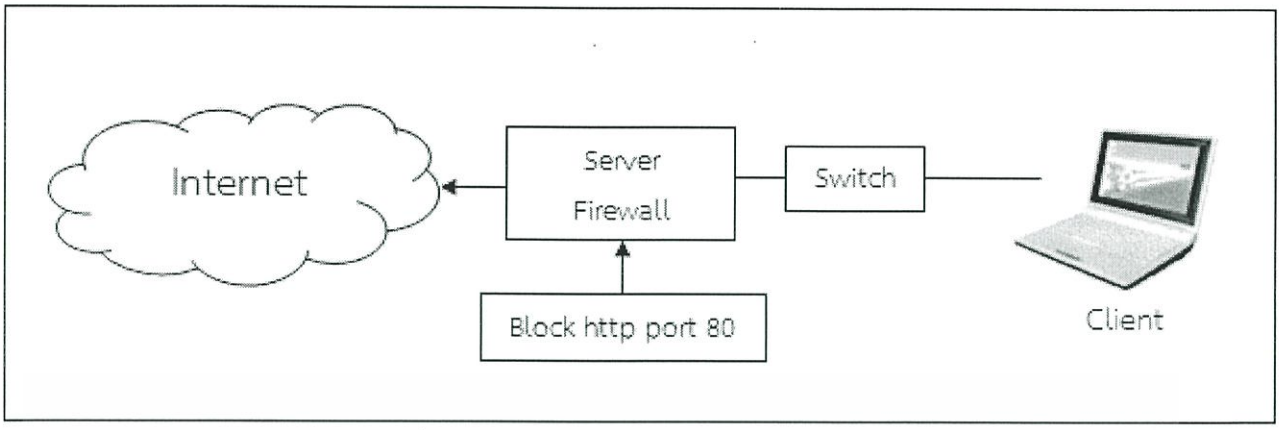


รูปที่ 3.19 บล็อกไดอะแกรมเมื่อเริ่มต้นใช้งานอินเทอร์เน็ต

3.3.2 ทดสอบการกำหนดโพลิซีเฉพาะอย่าง

3.3.2.1 ทดสอบบล็อค http

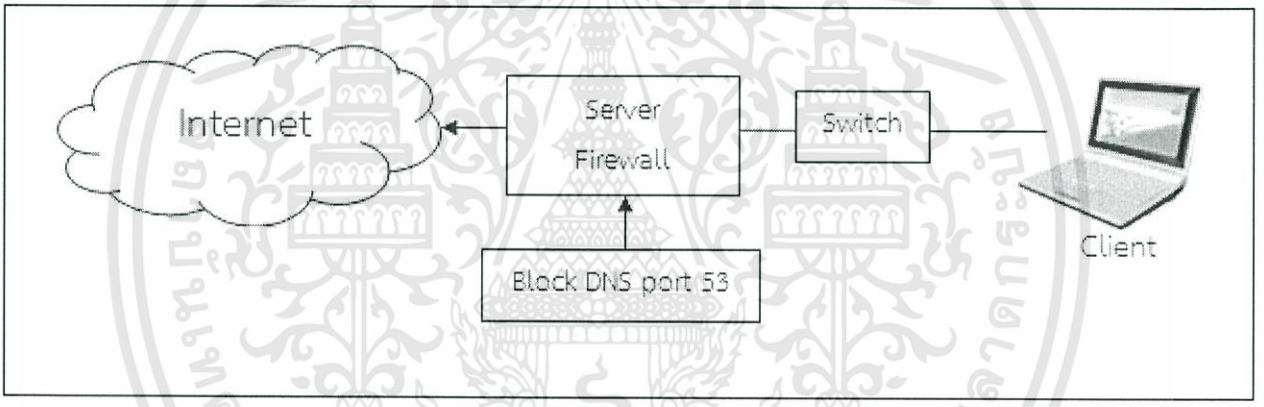
ทดสอบใช้ตัวอุปกรณ์กำหนดโพลิซีให้กับไฟร์วอลล์เพื่อบล็อค http ซึ่งเป็นพอร์ตทั่วไปของเว็บไซต์ต่างๆ (port 80) เพื่อไม่ให้ไคลเอนต์ใช้งานอินเทอร์เน็ต (เว็บไซต์ต่างๆที่เป็น http) บล็อกไดอะแกรมดังรูปที่ 3.20



รูปที่ 3.20 บล็อกไดอะแกรมบล็อค http พอร์ท 80

3.3.2.2 ทดสอบบล็อค DNS

ทดสอบใช้ตัวอุปกรณ์กำหนดโพลิซีให้กับไฟร์วอลล์เพื่อบล็อค DNS (port 53) เพื่อไม่ให้ไคลเอนต์เข้าใช้งานอินเทอร์เน็ตในทางโดเมนเนม (Domain Name) บล็อกไดอะแกรมดังรูปที่ 3.21

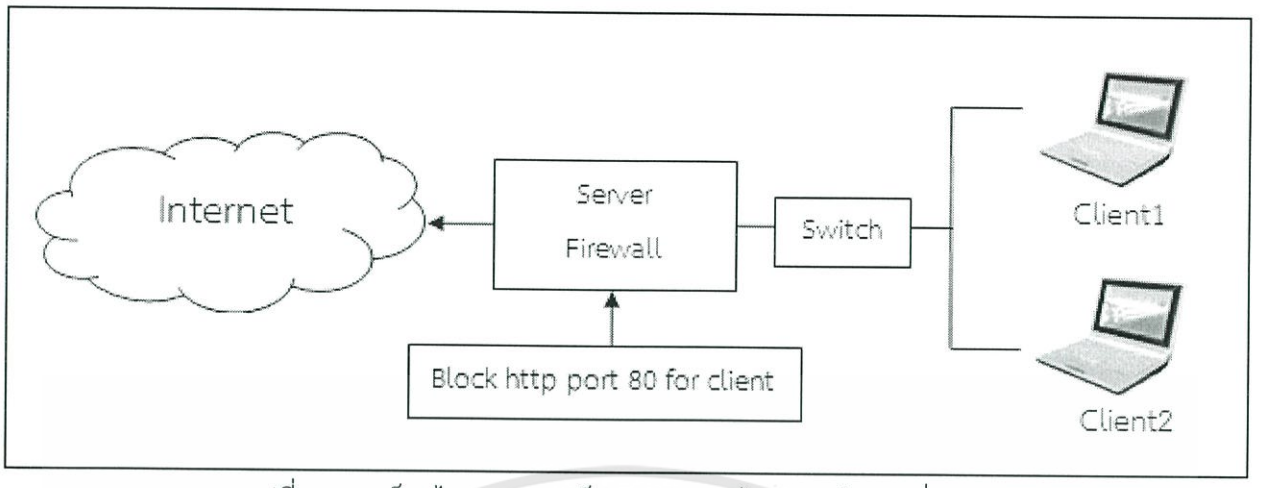


รูปที่ 3.21 บล็อกไดอะแกรมบล็อค DNS พอร์ท 53

3.3.3 ทดสอบการทำงานของตัวอุปกรณ์จัดการบล็อคแบบเฉพาะเจาะจง

3.3.3.1 ทดสอบบล็อคไคลเอนต์บางเครื่องไม่ให้ใช้งานเซอร์วิสแบบ http

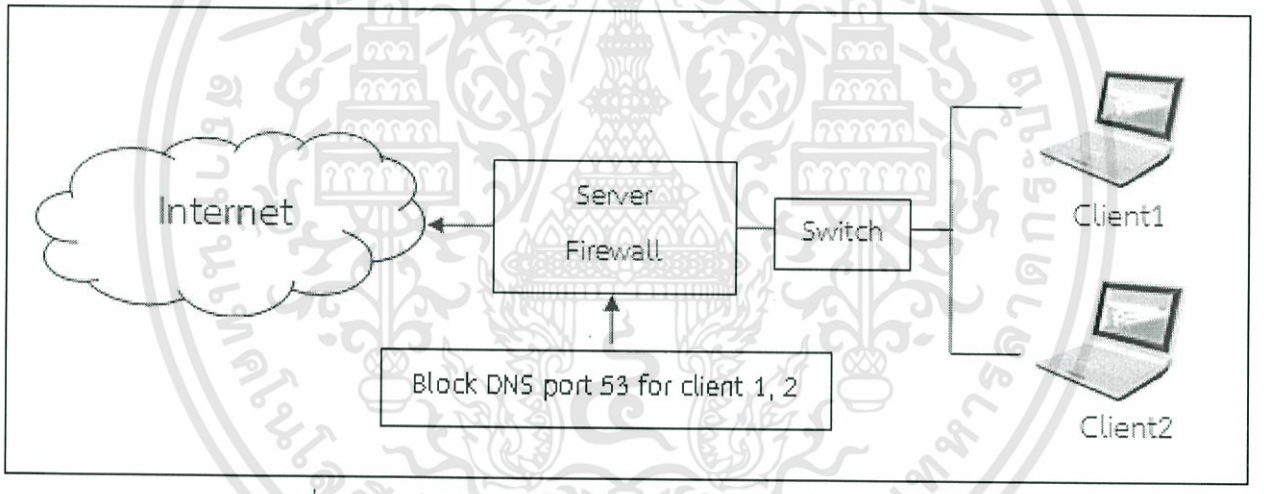
ทดสอบใช้ตัวอุปกรณ์กำหนดโพลิซีให้กับไฟร์วอลล์เพื่อบล็อค http (port 80) สำหรับไคลเอนต์บางเครื่อง เพื่อไม่ให้ไคลเอนต์นั้นๆใช้งานอินเทอร์เน็ตแบบ http (Port80) บล็อกไดอะแกรมดังรูปที่ 3.22



รูปที่ 3.22 บล็อกไดอะแกรมบล็อค http ของไคลเอนต์บางเครื่อง

3.3.3.2 ทดสอบบล็อคไคลเอนต์บางเครื่องไม่ให้ใช้งานเซอร์วิสแบบ DNS

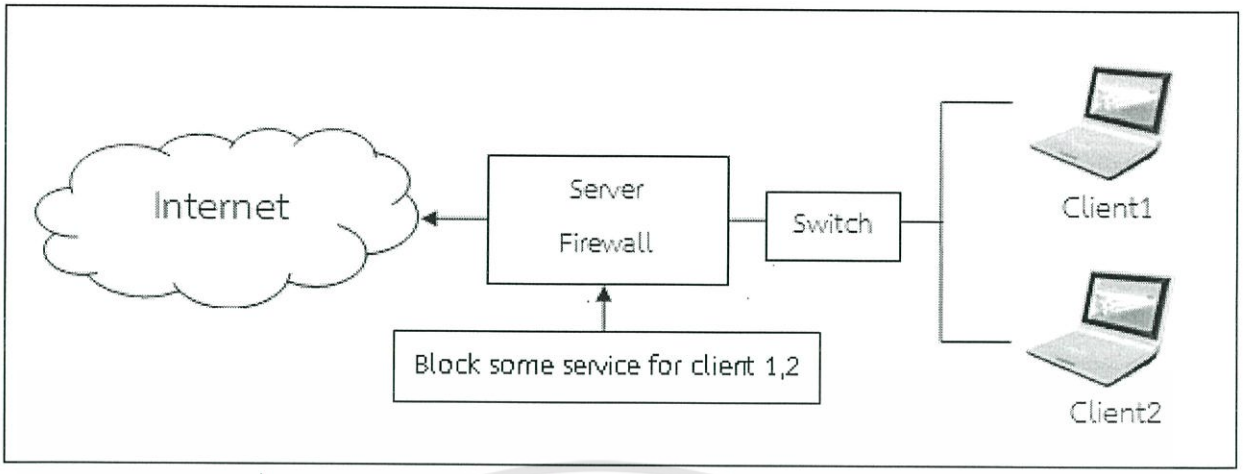
ทดสอบใช้ตัวอุปกรณ์กำหนดโพลิซีให้กับไฟร์วอลล์เพื่อบล็อคแบบเฉพาะเจาะจง โดยบล็อคไคลเอนต์บางเครื่องไม่ให้ใช้งานอินเทอร์เน็ตแบบ DNS (Port53) บล็อกไดอะแกรมดังรูปที่ 3.23



รูปที่ 3.23 บล็อกไดอะแกรมบล็อค DNS ไคลเอนต์บางเครื่อง

3.3.3.3 ทดสอบบล็อคเว็บไซต์ที่ไม่เหมาะสม

ทดสอบใช้ตัวอุปกรณ์กำหนดโพลิซีให้กับไฟร์วอลล์เพื่อบล็อคแบบเจาะจงเว็บไซต์ โดยบล็อคไคลเอนต์ไม่ให้ใช้งานเว็บไซต์ที่ไม่เหมาะสมได้ บล็อกไดอะแกรมดังรูปที่ 3.24



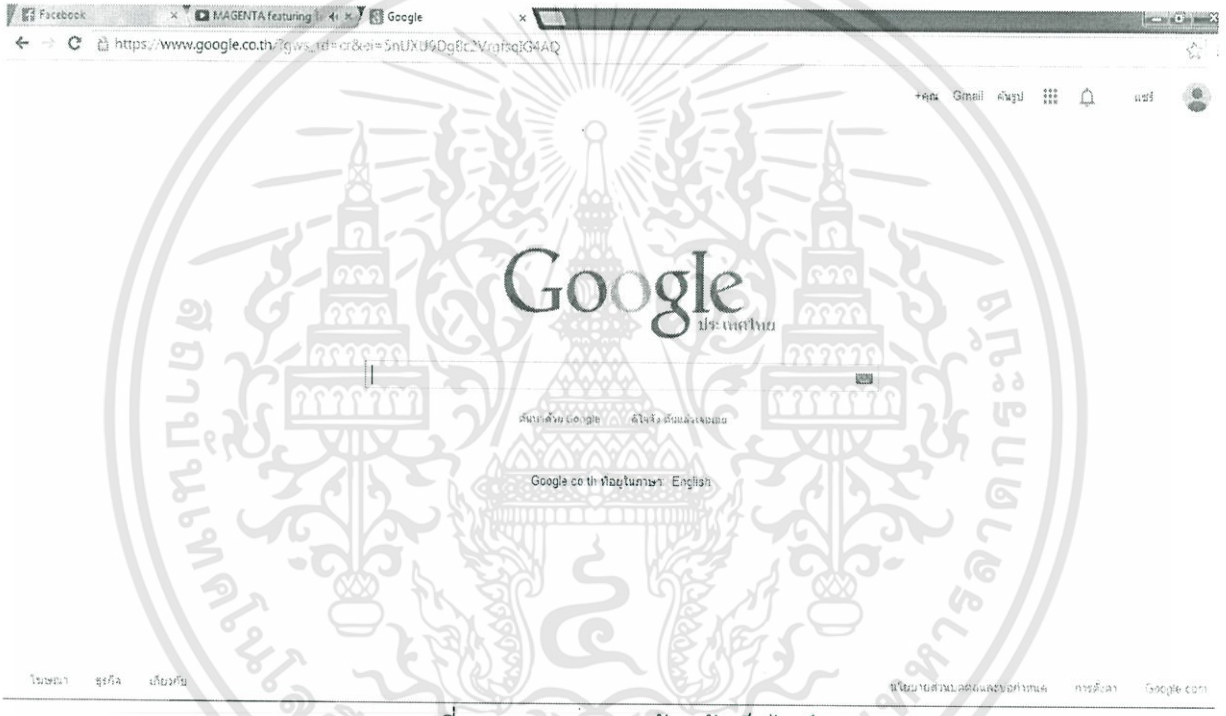
รูปที่ 3.24 บล็อกไดอะแกรมบล็อคเซอร์วิสเว็บไซต์แบบจำเพาะเจาะจง

บทที่ 4

ผลการทดลอง

4.1 ผลทดสอบการทำงานของอุปกรณ์เพื่อเริ่มต้นใช้งานระบบเครือข่าย

ทำการต่อตัวอุปกรณ์เข้ากับเซิร์ฟเวอร์เมื่อหน้าจอแอลซีดี (LCD)แสดงฟังก์ชันให้เลือก ทำการเลือกฟังก์ชัน 2 Policies setting และฟังก์ชันที่ 1 Set interface IP เพื่อเซ็ตอินเตอร์เฟซ WAN และอินเตอร์เฟซ LAN และเพื่อกำหนดโพลิซี (Policy) ให้กับไฟร์วอลล์ทดสอบให้ไคลเอนต์ (Client) ทำการเข้าใช้งานอินเทอร์เน็ตผ่านเบราว์เซอร์ต่างๆ เช่น internet explorer, Google chrome ฯ หน้าเว็บไซต์ (Web site)ผลเป็นดังรูปที่ 4.1

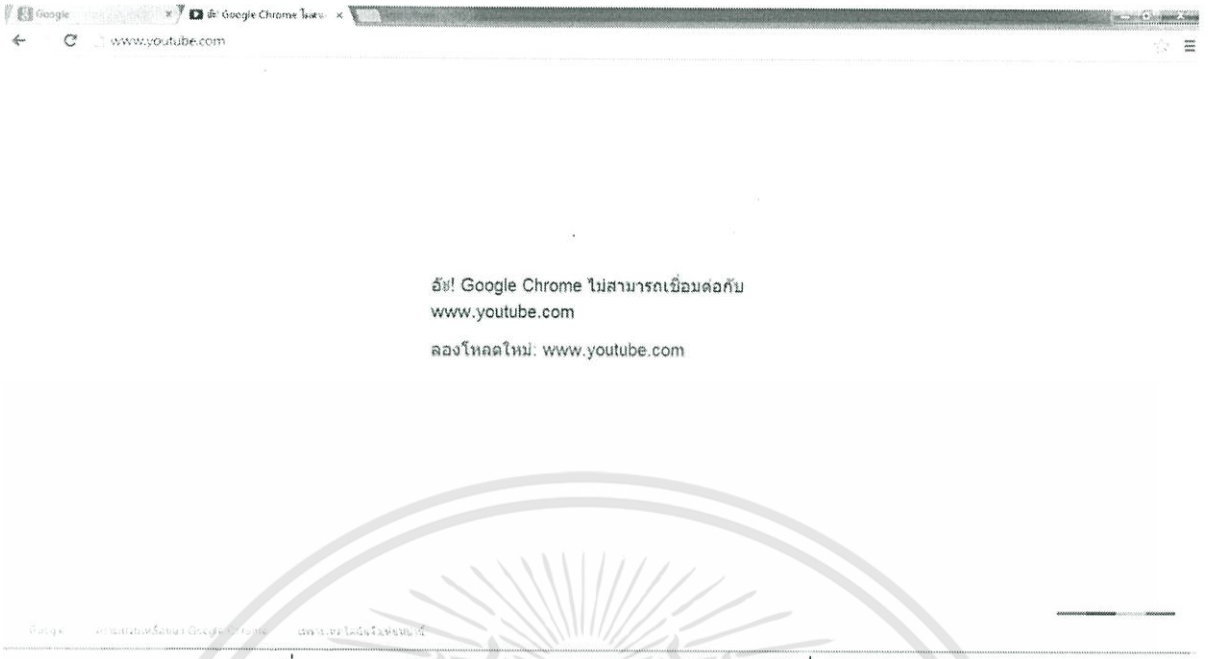


รูปที่ 4.1 การทดสอบเข้าหน้าเว็บไซต์

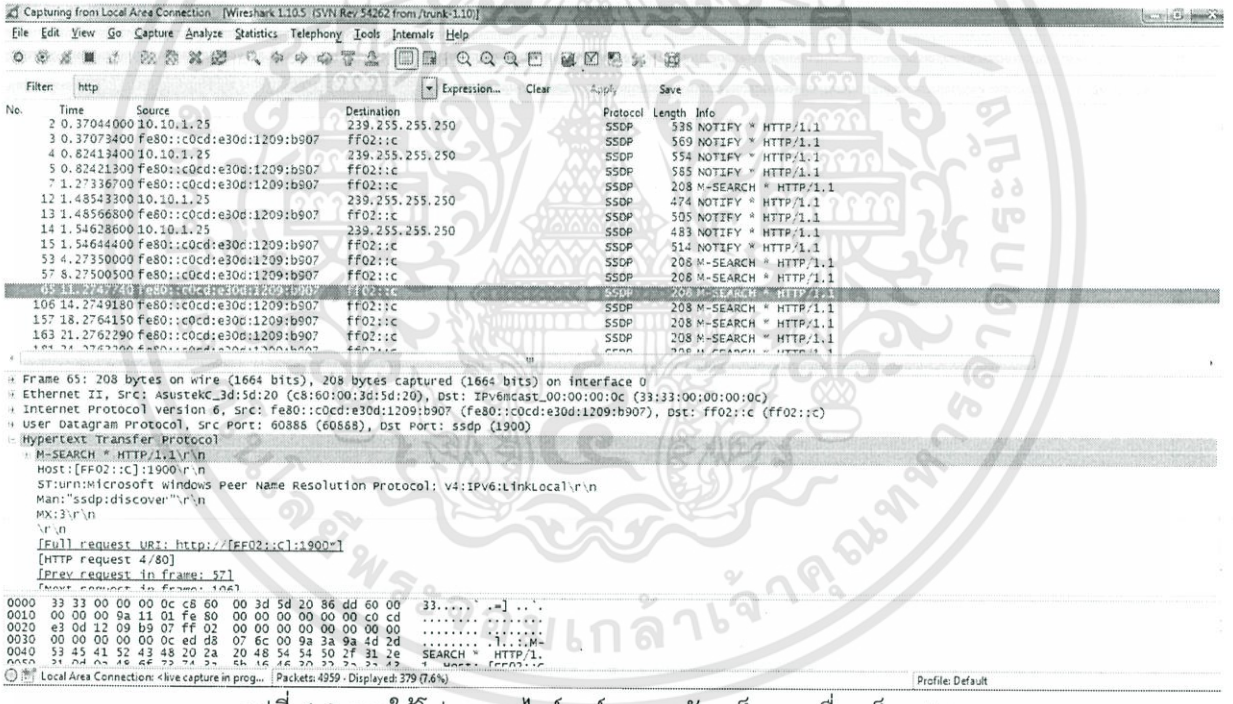
4.2 ผลทดสอบการกำหนดโพลิซีเฉพาะอย่าง

4.2.1 ผลการทดสอบบล็อค http

เลือกฟังก์ชัน2 Add rule wizard จากฟังก์ชัน Policies settingเพื่อกำหนดโพลิซีของไฟร์วอลล์โดยกำหนดให้บล็อคhttp ซึ่งเป็นพอร์ตทั่วไปของเว็บไซต์ต่างๆ(port80)ทำการทดสอบให้ไคลเอนต์เข้าใช้งานอินเทอร์เน็ต ผลเป็นดังรูปที่ 4.2 และใช้โปรแกรมไวร์ชาร์ก (Wireshark) ตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์ ผลเป็นดังรูปที่ 4.3



รูปที่ 4.2 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์เมื่อบล็อค http

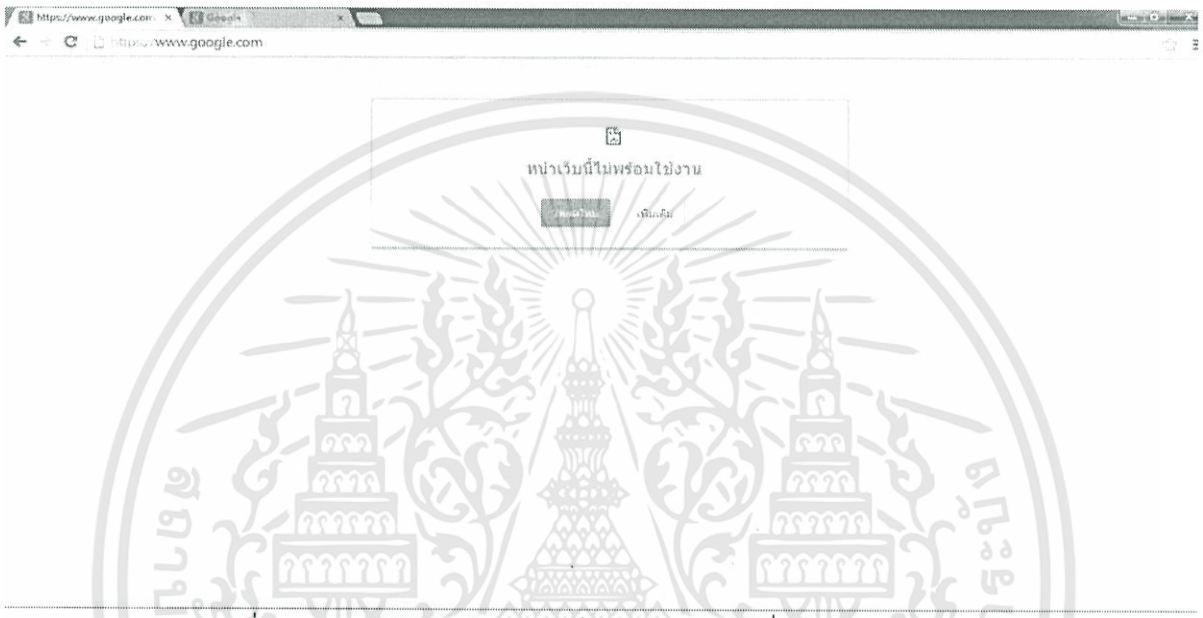


รูปที่ 4.3 การใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตเมื่อบล็อค http

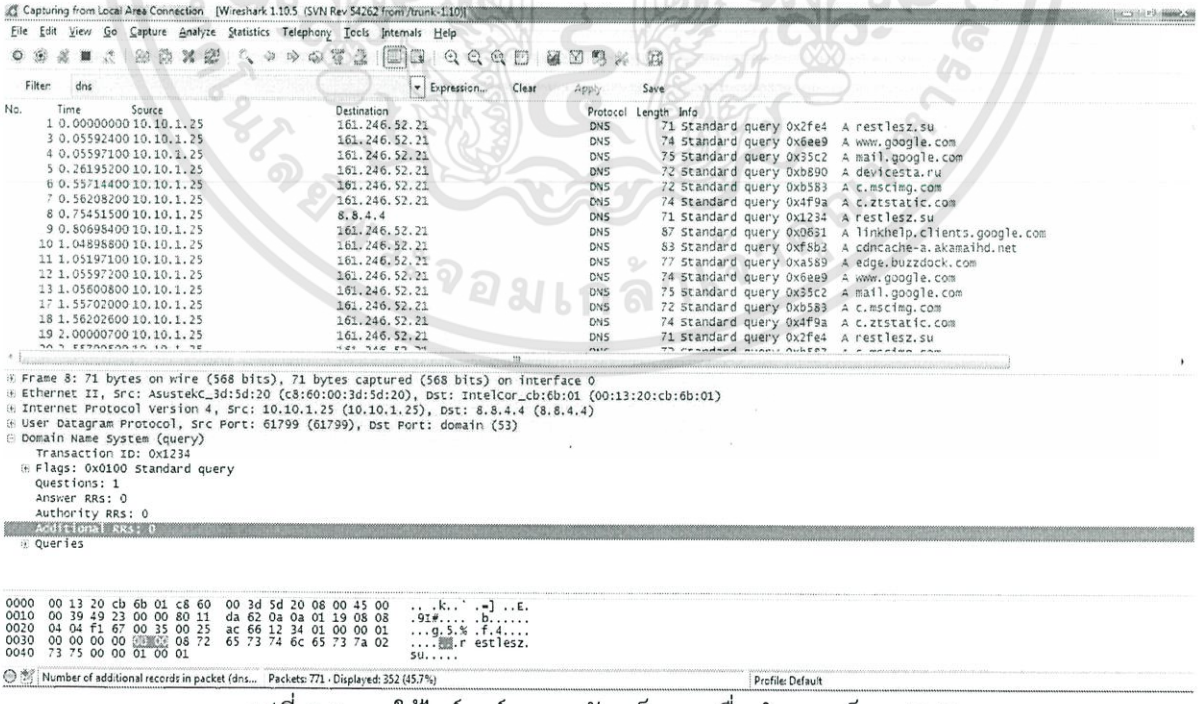
4.2.2 ผลการทดสอบบล็อค DNS

เลือกฟังก์ชัน2 Add rule wizard จากฟังก์ชัน Policies setting เพื่อกำหนดโพลิซีของไฟร์วอลล์โดยกำหนดให้บล็อคพอร์ต 53ซึ่งเป็นพอร์ตของDNS(Domain Name Service)ทำการทดสอบให้ไคลเอนต์เข้าใช้งานอินเทอร์เน็ต ผลเป็นดังรูปที่ 4.4และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์ ผลเป็นดังรูปที่ 4.5

จากนั้นทำการทดสอบให้ไคลเอนต์เข้าใช้งานอินเทอร์เน็ตด้วยไอพีแอดเดรส (IP Address)ผลเป็นดังรูปที่ 4.6และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์ ผลเป็นดังรูปที่ 4.7

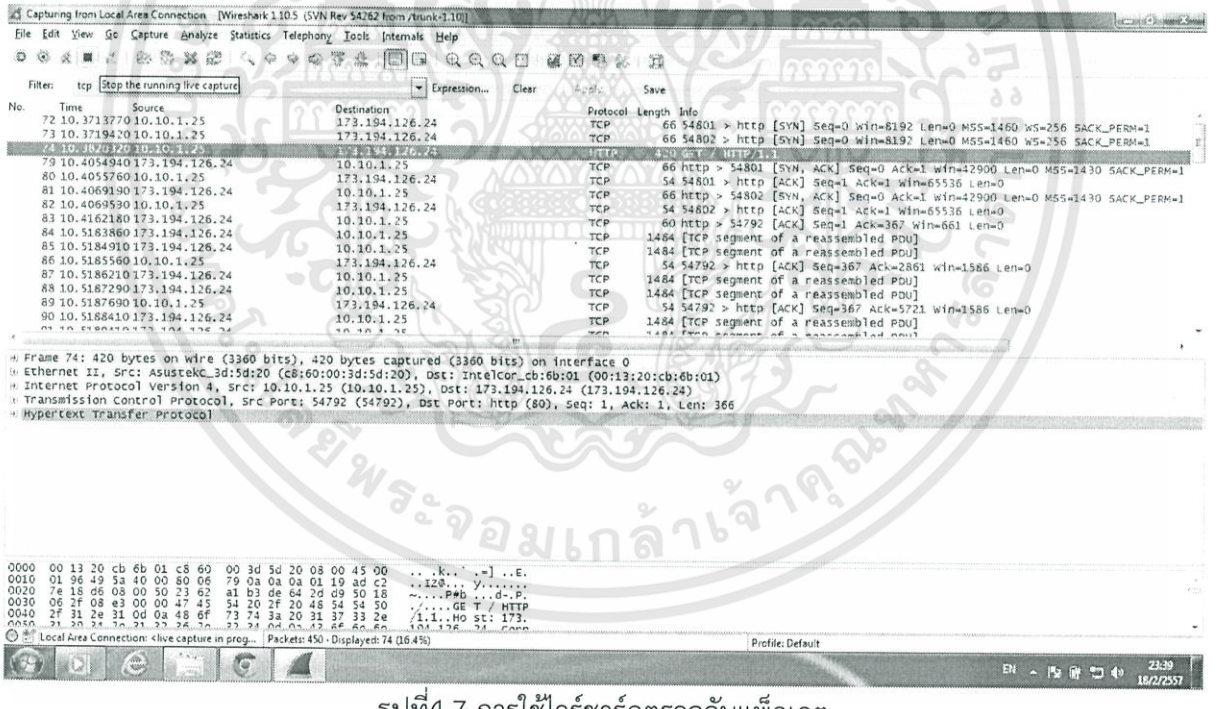


รูปที่ 4.4 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์เมื่อทำการบล็อค DNS



รูปที่ 4.5 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตเมื่อทำการบล็อค DNS

รูปที่ 4.6 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์ด้วยไอพีแอดเดรส



รูปที่4.7 การใช้ไวร์ชาร์กตรวจจับแพ็กเกต

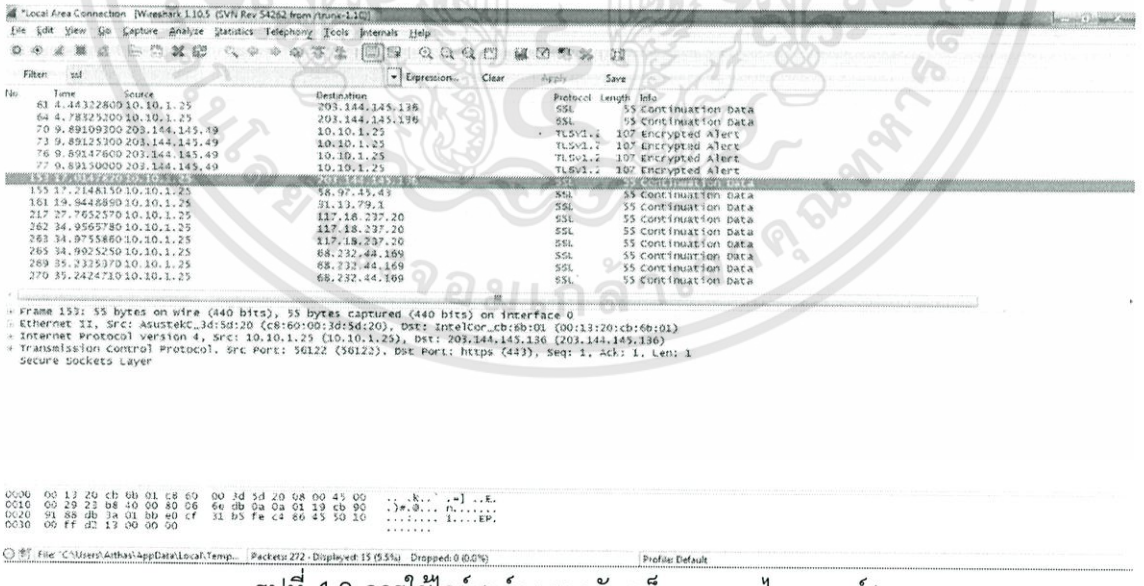
4.3 ผลการทดสอบการกำหนดโพลิซีแบบเฉพาะเจาะจง

4.3.1 ผลการทดสอบการบล็อคhttp สำหรับไคลเอนต์บางเครื่อง

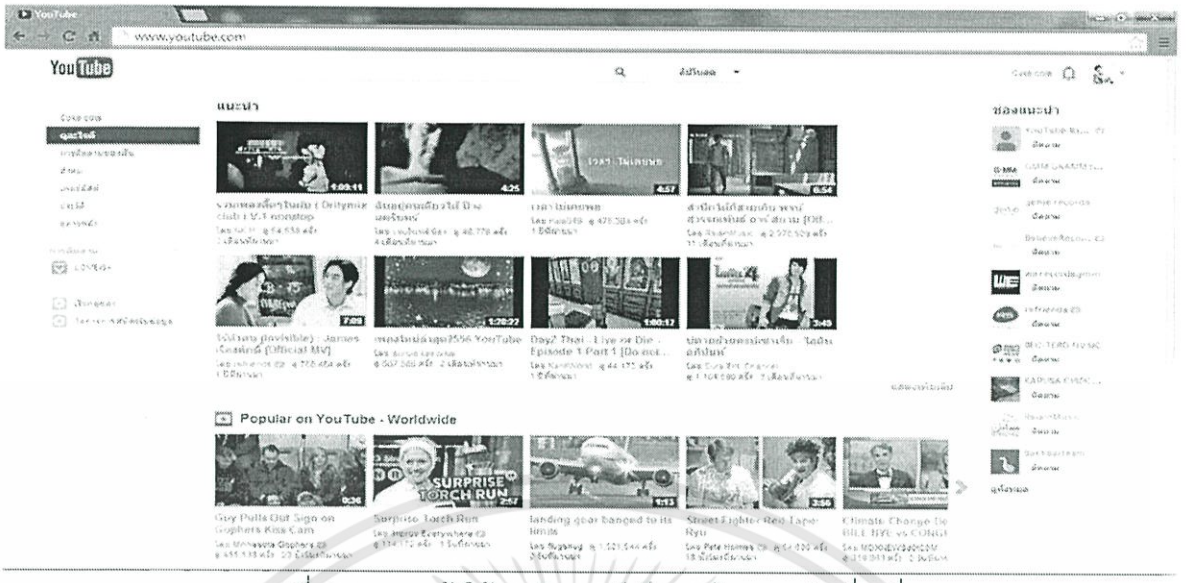
เลือกฟังก์ชัน2 Add rule wizard จากฟังก์ชัน Policies setting เพื่อกำหนดโพลิซีของไฟร์วอลล์โดยกำหนดให้บล็อคการเข้าใช้งานอินเทอร์เน็ตแบบ http สำหรับไคลเอนต์บางเครื่อง ในที่นี้กำหนดให้เป็นไคลเอนต์ 1 (Client 1)ทำการทดสอบให้ไคลเอนต์ 1 เข้าใช้งานอินเทอร์เน็ตแบบ http ผลเป็นดังรูปที่ 4.8และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์1 ผลเป็นดังรูปที่ 4.9

ทำการทดสอบให้ไคลเอนต์เครื่องอื่นให้ทดสอบเข้าใช้งานอินเทอร์เน็ตเพื่อเปรียบเทียบ ผลเป็นดังรูปที่ 4.10และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์อื่นเพื่อเปรียบเทียบ ผลเป็นดังรูปที่ 4.11

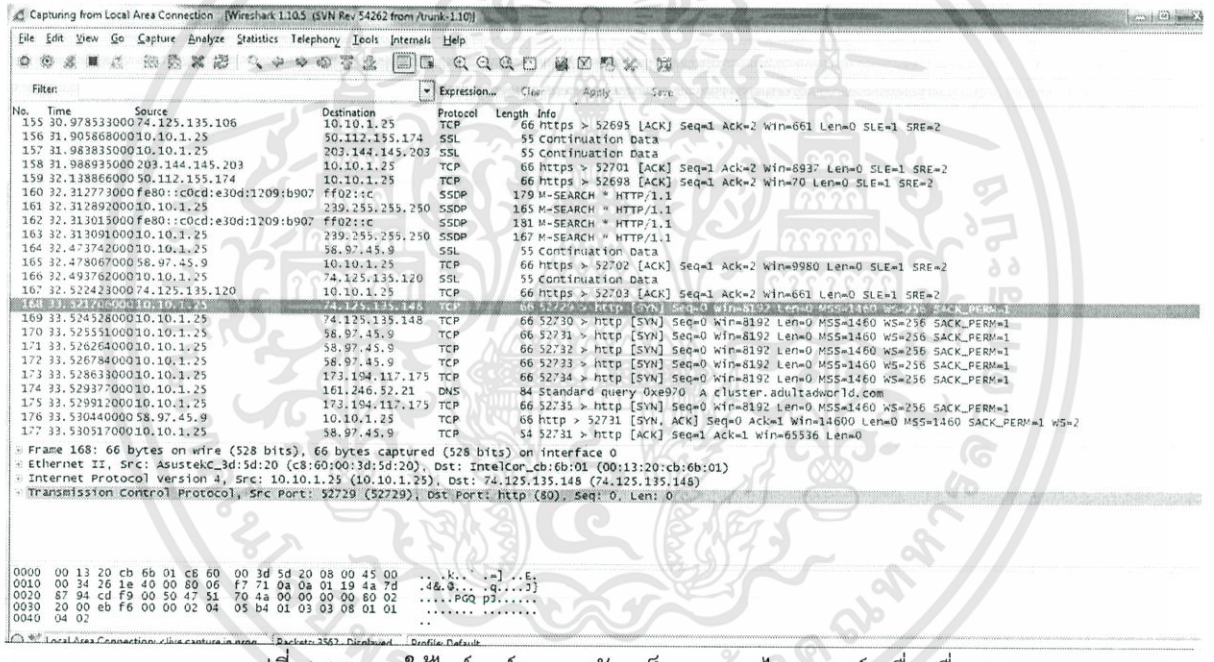
รูปที่ 4.8 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์1



รูปที่ 4.9 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตของไคลเอนต์1



รูปที่ 4.10 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์เครื่องอื่น

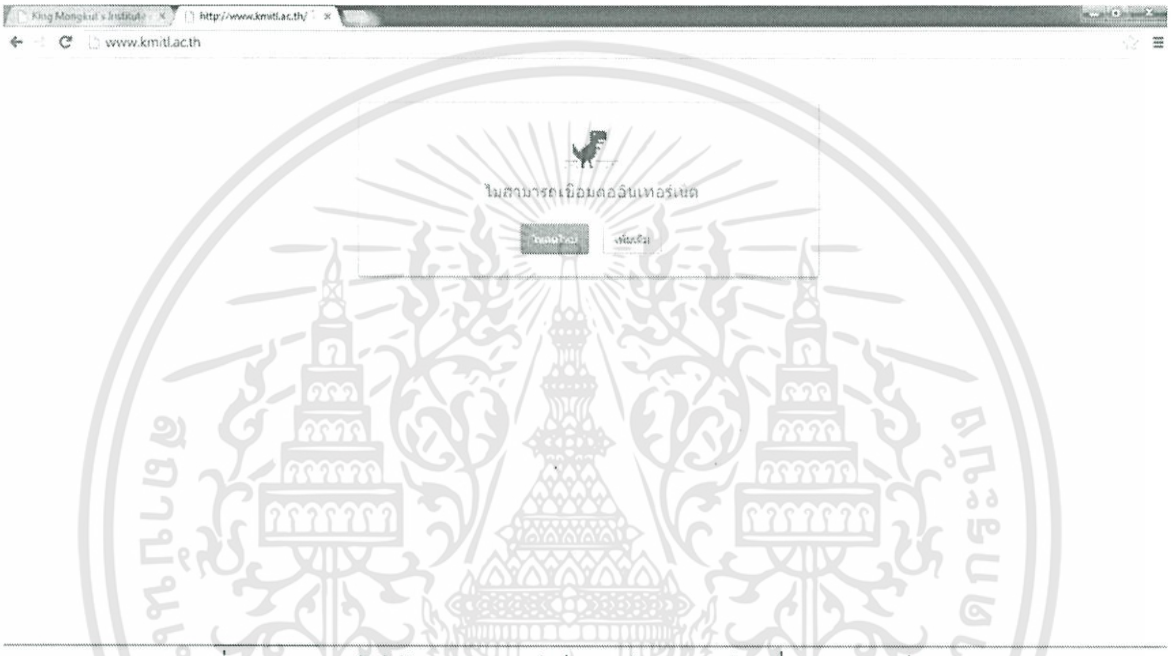


รูปที่ 4.11 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตของไคลเอนต์เครื่องอื่น

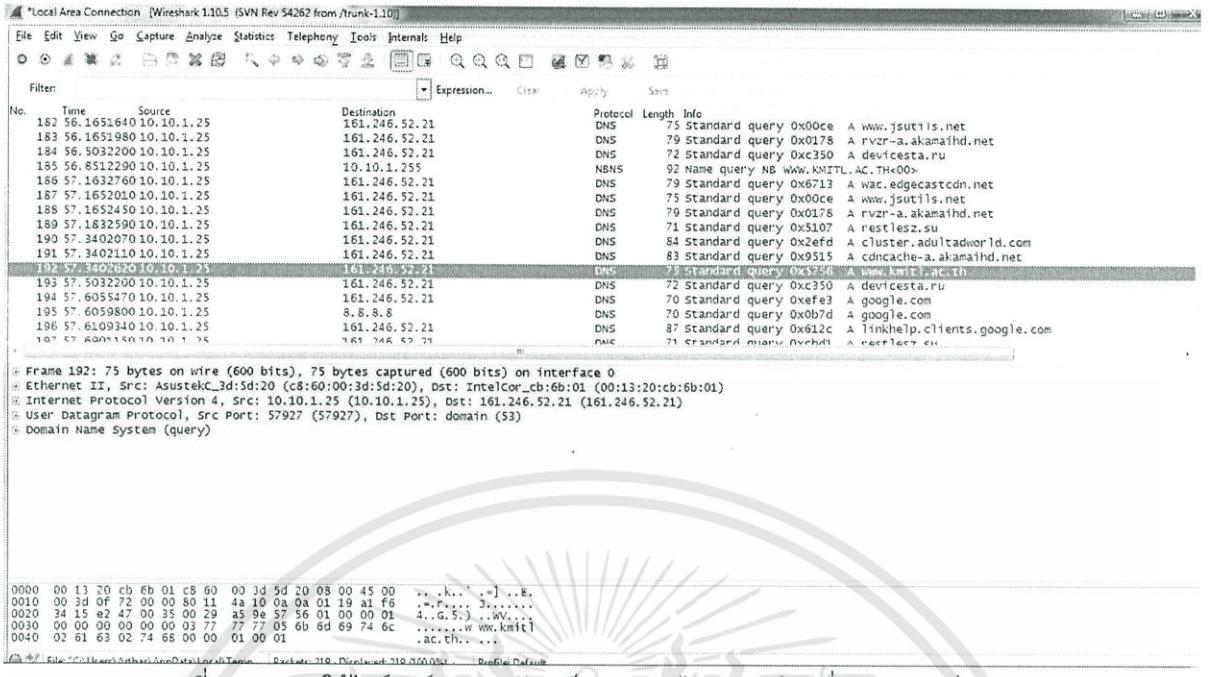
4.3.2 ผลการทดสอบบล็อคเครื่องลูกข่ายบางเครื่องไม่ให้เข้าใช้งานอินเทอร์เน็ตแบบ DNS

เลือกฟังก์ชัน2 Add rule wizard จากฟังก์ชัน Policies setting เพื่อกำหนดโพลิซีของไฟร์วอลล์โดยกำหนดให้บล็อคการเข้าใช้งานอินเทอร์เน็ตแบบ DNS สำหรับไคลเอนต์บางเครื่อง ในที่นี้กำหนดให้เป็นไคลเอนต์ 1 ทำการทดสอบให้ไคลเอนต์1 ใช้งานอินเทอร์เน็ตโดยใส่ชื่อเว็บไซต์แบบปกติ ผลเป็นดังรูปที่ 4.12 จากนั้นทำการทดสอบให้ไคลเอนต์1 เข้าใช้งานอินเทอร์เน็ตโดยใช้ไอพีแอดเดรสเพื่อเปรียบเทียบ ผลเป็นดังรูปที่ 4.14 และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์1 ผลเป็นดังรูปที่ 4.13 และ 4.15

ทำการทดสอบให้ไคลเอนต์เครื่องอื่นให้ทดสอบเข้าใช้งานอินเทอร์เน็ตเพื่อเปรียบเทียบ ผลเป็นดังรูปที่ 4.16และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์อื่น ผลเป็นดังรูปที่ 4.17

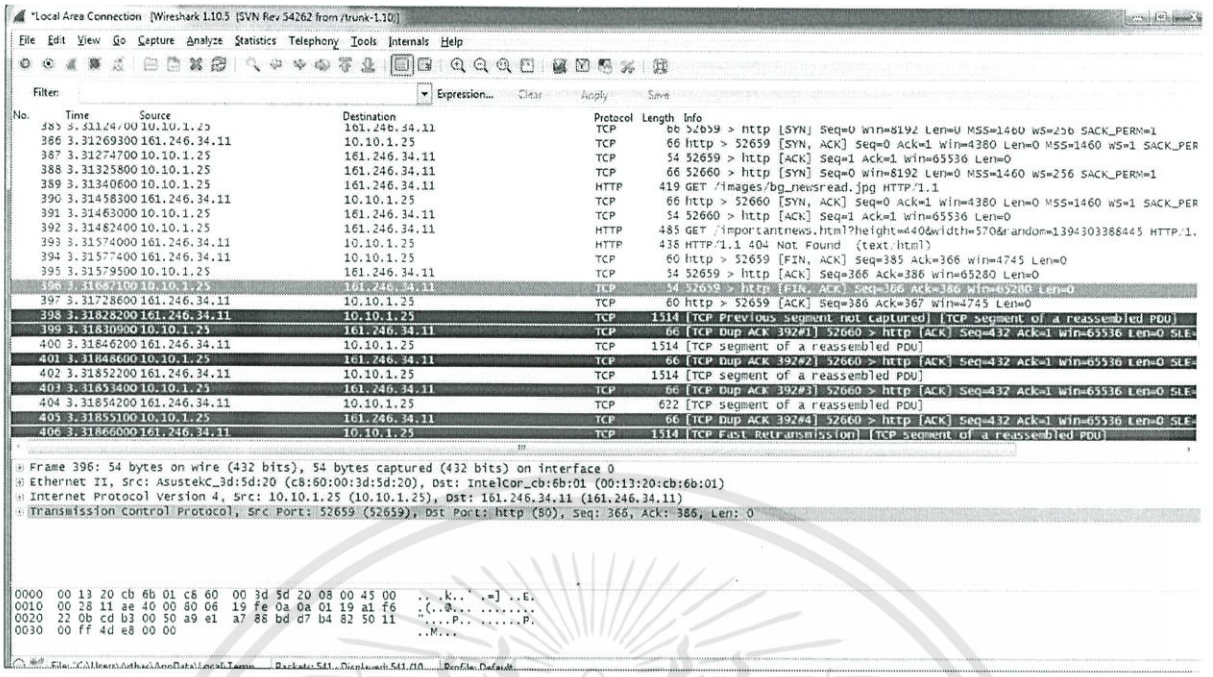


รูปที่ 4.12 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์1 เมื่อทำการบล็อคDNS



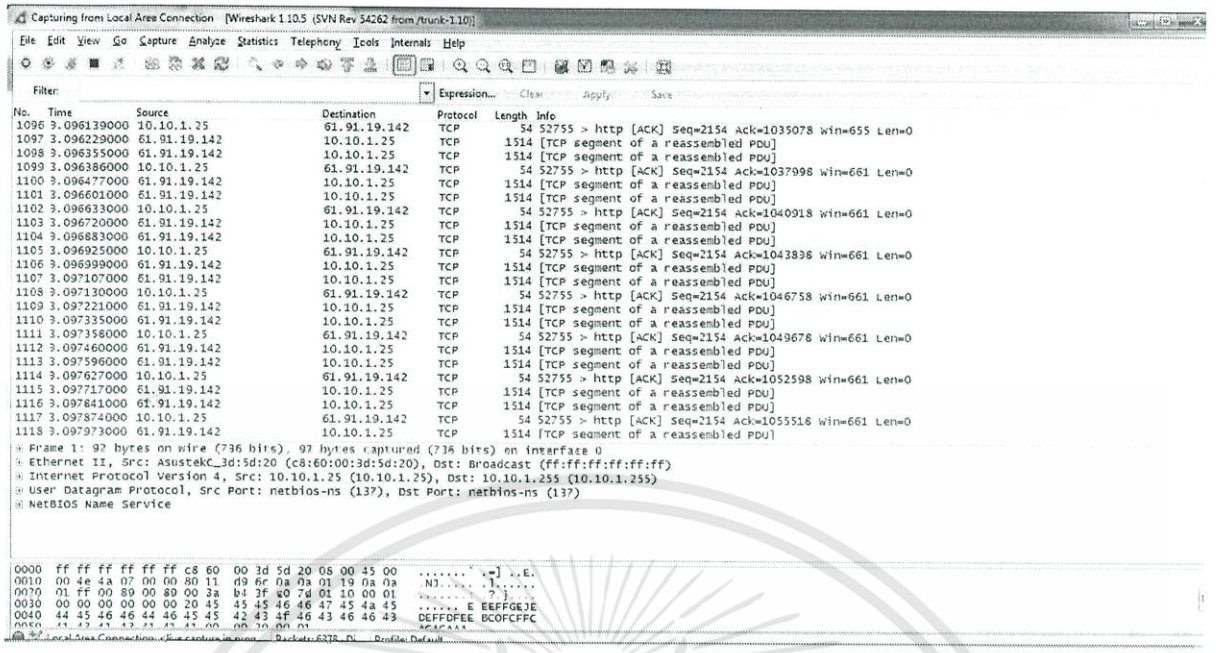
รูปที่ 4.13 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตของไคลเอนต์1 เมื่อทำการบล็อคDNS

รูปที่ 4.14 การเข้าใช้งานอินเทอร์เน็ตของไคลเอนต์1 ด้วยไอพีแอดเดรส



รูปที่ 4.15 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตของไคลเอนต์1 เมื่อใช้งานอินเทอร์เน็ตด้วยไอพีแอดเดรส

รูปที่ 4.16 การเข้าใช้อินเทอร์เน็ตของเครื่องไคลเอนต์เครื่องอื่น

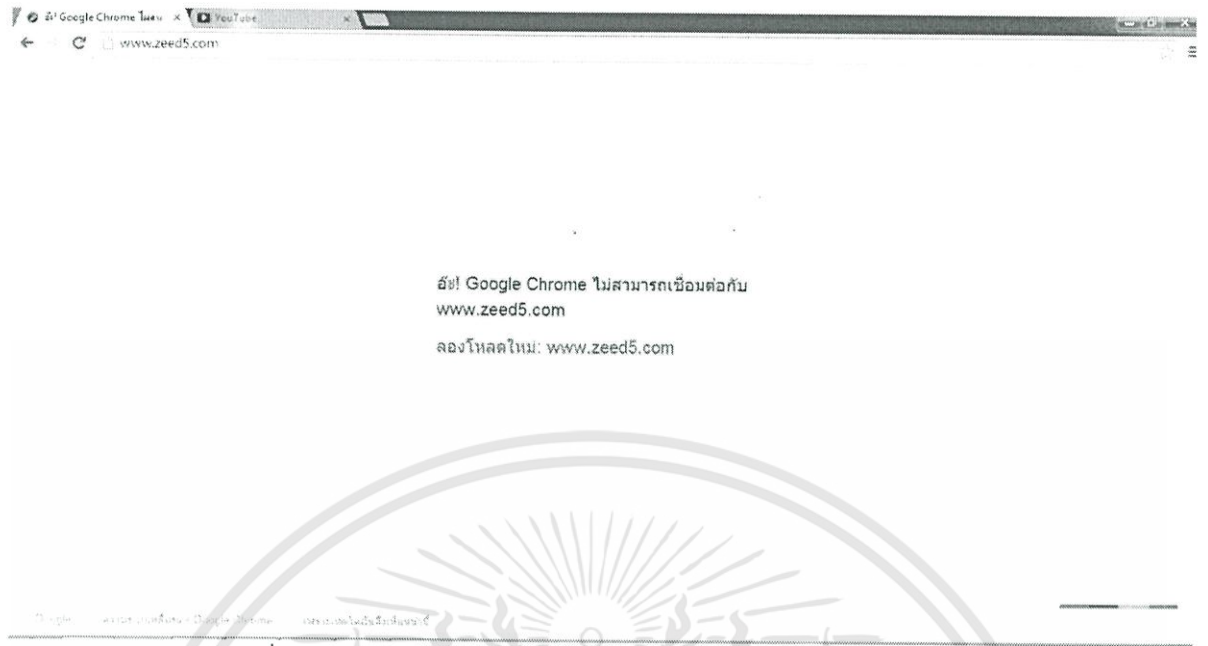


รูปที่ 4.17 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตของเครื่องไคลเอนต์เครื่องอื่น

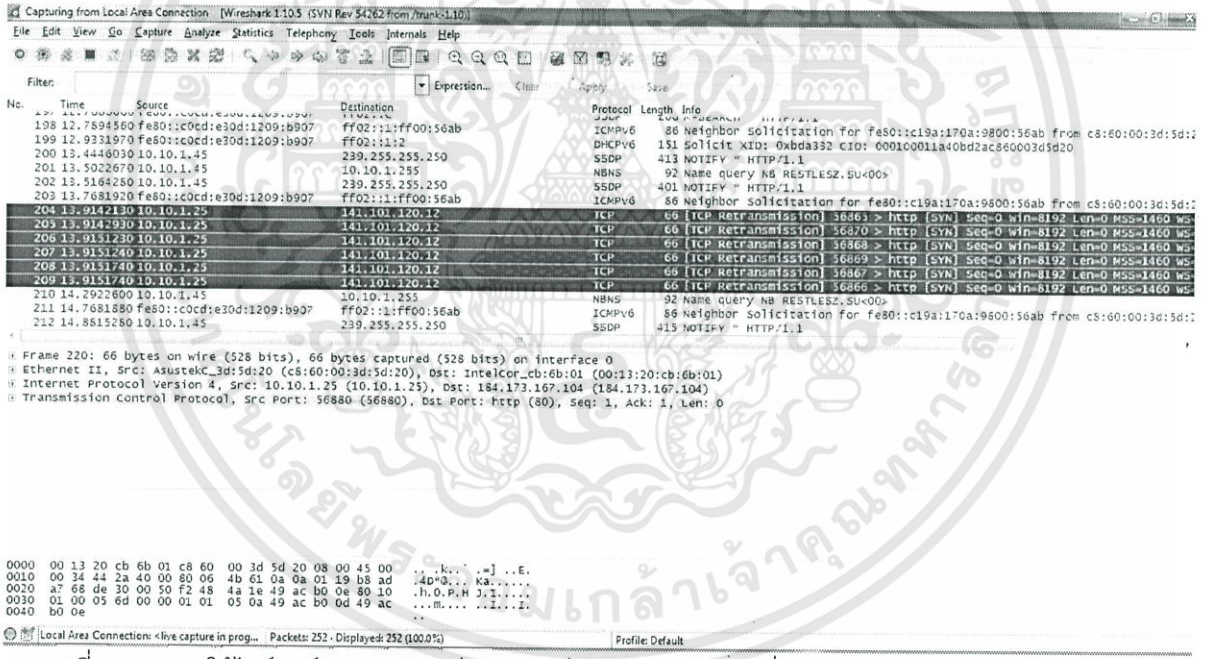
4.3.3 ผลการทดสอบบล็อคไคลเอนต์ไม่ให้ใช้งานเว็บไซต์ที่ไม่เหมาะสม

เลือกฟังก์ชัน2 Add rule wizard จากฟังก์ชัน Policies setting เพื่อกำหนดโพลิซีของไฟร์วอลล์ โดยกำหนดให้บล็อคการเข้าใช้งานอินเทอร์เน็ตในเว็บไซต์ที่ไม่เหมาะสมสำหรับไคลเอนต์ในที่นี้กำหนดให้บล็อค www.zeed5.comซึ่งเป็นเว็บไซต์ที่ไม่เหมาะสมประเภทสื่อลามก ทำการทดสอบให้ไคลเอนต์เข้าใช้งานอินเทอร์เน็ตในเว็บไซต์ดังกล่าวในขณะที่ยังไม่ได้ทำการบล็อคผลเป็นดังรูปที่ 4.18และหลังจากทำการบล็อคเว็บไซต์ดังกล่าว ทดสอบให้ไคลเอนต์เข้าใช้งานเว็บไซต์ดังกล่าวอีกครั้งเพื่อเปรียบเทียบ ผลเป็นดังรูปที่ 4.19 และใช้โปรแกรมไวร์ชาร์กตรวจจับแพ็กเกตที่เข้ามายังเครื่องไคลเอนต์ ผลเป็นดังรูปที่ 4.20

รูปที่ 4.18 การเข้าใช้งานเว็บไซต์ที่ไม่เหมาะสมก่อนการบล็อคเว็บไซต์



รูปที่ 4.19 การเข้าใช้งานเว็บไซต์ที่ไม่เหมาะสมหลังการบล็อคเว็บไซต์



รูปที่ 4.20 การใช้ไวร์ชาร์กตรวจจับแพ็กเกตการเข้าใช้งานเว็บไซต์ที่ไม่เหมาะสมหลังการบล็อคเว็บไซต์

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้ได้พัฒนาแอพพลิเคชั่นสำหรับรักษาความปลอดภัยให้กับระบบเครือข่ายบนพื้นฐานระบบปฏิบัติการลินุกซ์ อูบุนตู โดยใช้ไอพีเทเบิล (IP Tables) มาใช้เป็นโครงของแอพพลิเคชั่น ใช้โปรแกรมเน็ตบีนส์ (Net beans) ในการเขียนโค้ดโปรแกรมจากภาษาจาวาเพื่อสร้างชุดคำสั่ง และใช้โปรเซสบิวเดอไลบารี่ในการแปลพารามิเตอร์ต่างๆจากผู้ใช้โปรแกรมไปเป็นภาษาที่ไอพีเทเบิลเข้าใจและทำตามที่ผู้ใช้โปรแกรมกำหนด ใช้ไมโครคอนโทรลเลอร์ควบคุมคีย์แพดเพื่อกำหนดค่าคอนฟิกโพลิซีของไฟร์วอลล์ และใช้แอลซีดีเป็นส่วนแสดงผล ใช้ RS232 เป็นพอร์ตสื่อสารระหว่างคอมพิวเตอร์เซิร์ฟเวอร์กับบอร์ดไมโครคอนโทรลเลอร์ ซึ่งโปรแกรมจะทำการบล็อคแพ็คเกตต่างๆ หรือว่าอนุญาตให้แพ็คเกตใดผ่านเข้าออกจากเครือข่ายได้แล้วแต่ตามที่ผู้ใช้โปรแกรมกำหนด จากการทดลองโปรแกรมไฟร์วอลล์สามารถใช้งานได้จริงตามผลลการทดสอบ

5.2 ข้อเสนอแนะ

สำหรับตัวอุปกรณ์มีข้อจำกัดในด้านหน่วยความจำโปรแกรม ในการเพิ่มเติมฟังก์ชันอาจจำเป็นต้องตัดฟังก์ชันบางอย่างของเดิมออกเสียก่อนจึงจะสามารถเพิ่มเติมฟังก์ชันอื่นๆเข้ามาได้ ข้อจำกัดเรื่องของหน่วยความจำโปแกรม อาจใช้การเปลี่ยนบอร์ดสำเร็จรูปไปใช้รุ่นที่พัฒนาขึ้นมาให้มีหน่วยความจำโปรแกรมที่มากเพียงพอต่อขนาดของโปรแกรม

ไฟร์วอลล์ไม่ใช่อุปกรณ์ป้องกันการบุกรุกระบบที่สมบูรณ์ 100 เปอร์เซ็นต์ เพราะการโจมตีที่เกิดขึ้นในปัจจุบันนั้นมักจะโจมตีผ่านพอร์ตที่เป็นที่รู้จัก เช่น 21, 22, 25, 53, 80 ซึ่งไฟร์วอลล์มักจะอนุญาตให้ข้อมูลเข้าออกผ่านทางพอร์ตเหล่านี้เสมอ ดังนั้นควรใช้เครื่องมืออื่นๆ ช่วย เช่น การตั้งนโยบายความปลอดภัยที่รัดกุม การนำระบบเน็ตเวิร์คอินทรูชันดีเทคชัน (Network intrusion detection) มาใช้ หรือแม้แต่มีการจัดการระดับโฮสต์เบส (Host based) ที่ดี เช่น มีการทำฮาเดนนิ่งโอเอส (Hardening OS) ในเน็ตเวิร์คโอเปอเรติงซิสเต็ม (Network operating system) ทุกๆ เครื่องที่เปิดให้บริการแก่สาธารณะ ก็จะช่วยเพิ่มความปลอดภัยให้กับระบบมากยิ่งขึ้น

บรรณานุกรม

[1] ชนินทร์ เชาวมิตร. คู่มือยูนิกซ์เดสก์ทอป. กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

[2] พิเชษฐ์ ศิริรัตนไพศาลกุล. ระบบปฏิบัติการ (Operating system). กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

[3] ดร.ยรรยง เต็งอำนวย. ระบบปฏิบัติการ (Operating system). กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

[4] ปราการ โกลากุล. "ไฟร์วอลล์." http:// antivirus.nabia10.com/virus-t/at15.html.

[5] "การสร้าง Firewall ด้วย IPTABLES." http:// hadyaiinternet.com/index.php?topic-75.0.

[6] Ce_network_Thai. "Iptables." http:// ce-networks.blogspot.com/2011/04/iptables.html.

[7] ธีรภัทร มนตรีศาสตร์. "ไฟร์วอลล์." http://www.itdestination.com/articles/arnofirewall/.

[8] somporn. "PacketFilter." http://linuxfirewall.in.th/zeroshell.html/108firewall/105 firewall. html.

[9] เอกภพ สุทธิปาริชาติ. "set iptables."http://www.jobpub.com/articles/showarticle.asp?id=487.

[10] Arkom Thaicharoen. "คำสั่งลินุกซ์." http://www.slideshare.net/arkomt?utm_campaign=Pro filetracking&utm_medium=sssite&utm_source=ssslideview.

ภาคผนวก
คู่มือการใช้งานอุปกรณ์

เมื่อเริ่มใช้งานอุปกรณ์จะมีเมนูให้เลือก 4 เมนูโดยมีดังนี้
-Show System Info เลือกปุ่มที่ 1
-Policy Setting เลือกปุ่มที่ 2
-Chassic Control เลือกปุ่มที่ 3
-Show Policies เลือกปุ่มที่ 4

1.Show System Info เมนูนี้จะใช้สำหรับโชว์ค่าสถานะของเครื่องเซิร์ฟเวอร์ มีเมนูสองเมนู
-Refresh เลือกปุ่มที่1 เป็นการรีเฟรชค่า ให้เห็นค่าสถานะล่าสุดของเครื่องเซิร์ฟเวอร์
-Back To Main เลือกปุ่มที่2 เป็นการกลับไปหน้าเมนูหลัก

2.Policy Setting เมนูนี้จะมี 4 เมนู
-Set Interface IP เลือกปุ่มที่ 1 เมนูนี้จะไปกำหนดค่า ไอพี LAN ไอพี WAN และ เกตเวย์(Gateway) ให้กับระบบเครือข่าย ควรทราบก่อนว่าเครื่องเซิร์ฟเวอร์อยู่ในระบบเครือข่ายหมายเลขไอพีใด ทำให้สามารถกำหนดค่าไอพี LAN WAN และ เกตเวย์ได้

1.ใส่หมายเลขไอพีที่ต้องการ แล้วกด ENT เพื่อยืนยันค่า
2.ใส่หมายเลขซับเน็ตมาร์ค แล้วกด ENT เพื่อยืนยันค่า
3.กำหนดค่าให้เป็นไอพีของ LAN WAN หรือ เกตเวย์ ถ้าเลือกให้ LAN เลือกปุ่มหมายเลข 1 ถ้าเลือก WAN เลือกปุ่มหมายเลข 2 หากเลือกเกตเวย์เลือกปุ่มหมายเลข 3 หากต้องการยกเลิกการกำหนดค่าให้กดปุ่มใดๆก็ได้ที่ไม่ใช่หมายเลข 1,2,3 เมื่อกดหมายเลขที่ต้องการแล้ว ให้กดปุ่ม ENT เพื่อยืนยันค่า
4.จากนั้นโปรแกรมจะแสดงค่าพารามิเตอร์ที่ได้กำหนดไปจากข้อที่1,2,3หากต้องการแก้ไขค่าพารามิเตอร์ หรือต้องการกำหนดค่าเพิ่มเติม กดปุ่มหมายเลข 1 Edit หากต้องการกำหนดค่าโพลิซีไฟร์วอลล์กดปุ่มหมายเลข 2 Add Rule สุดท้ายปุ่มหมายเลข 3 Go back คือการกลับไปยังหน้าเมนู Set Interface IP

2.Add Rule Wizard เลือกปุ่มหมายเลข 2 เมนูนี้จะใช้กำหนดค่าโพลีซีไฟร์วอล์ ไฟวอล์ต้องมีค่า หมายเลขไอพี(IP) ซับเน็ตมาร์ค(Subnet Mask) หมายเลขพอร์ต(Port) โปรโตคอล (Protocal) และค่าแอคชั่น

1.โปรแกรมจะให้กำหนดค่าให้ต้นทาง ใส่ค่าหมายเลขไอพี แล้วกดปุ่ม ENT ถัดไปใส่ค่าซับเน็ตมาร์ค แล้วกดปุ่ม ENT ถัดไปใส่หมายเลขพอร์ต แล้วกดปุ่ม ENT ในกรณีการใส่ค่าพอร์ตหากต้องการให้หมายเลขพอร์ตต้นทางเป็นหมายเลขใดๆ ให้กดปุ่ม SET แล้ดกดปุ่ม ENT
2.เมื่อกำหนดค่าสำหรับต้นทางเสร็จแล้ว โปรแกรมจะให้เรากำหนดค่าสำหรับปลายทาง ใส่ค่าหมายเลขไอพี แล้วกดปุ่ม ENT ถัดไปใส่ค่าซับเน็ตมาร์ค แล้วกดปุ่ม ENT ถัดไปใส่หมายเลขพอร์ต แล้วกดปุ่ม ENT ในกรณีการใส่ค่าพอร์ตหากต้องการให้หมายเลขพอร์ตปลายเป็นหมายเลขใดๆ ให้กดปุ่ม SET แล้ดกดปุ่ม ENT
3.เลือกโปรโตคอลให้โพลิซี TCP กดปุ่มหมายเลข 1 UDP กดปุ่มหมายเลข 2 All กดปุ่มใดที่ไม่ใช่ 1,2 จากนั้นกด ENT
4.เลือกค่าแอคชั่นให้โพลีซี Allow กดปุ่มหมายเลข 1 สำหรับอนุญาติ(Accept) Deny กดปุ่มหมายเลข 2 สำหรับไม่อนุญาติ(Drop) หากต้องการยกเลิกโพลีซีทั้งหมดให้กดปุ่มอะไรก็ได้ที่ไม่ใช่ 1,2

จากนั้นกดปุ่ม ENT โปรแกรมจะแสดงค่าที่เรากำหนดไปทั้งหมดให้ดู จากนั้นจะให้กดปุ่ม ENT เพื่อกลับไปยังหน้าเมนู Policy Setting

-Show summary เลือกปุ่มหมายเลข 3 เมนูนี้จะแสดงค่าที่ได้กำหนด มาจาก เมนูที่1คือ Set Interface IP และยังสามารถเข้าไปแก้ไขการกำหนดค่าภายใน Set Interface IP ได้ด้วย ต้องการแก้ไขกดปุ่มหมายเลข 1 ต้องการกำหนดค่าโพลีซีไฟร์วอล์กดปุ่มหมายเลข 2 หากต้องการกลับไปหน้าเมนู Policy Setting กดปุ่มใดๆก็ได้ที่ไม่ใช่หมายเลข 1,2 จากนั้นกด ENT

-To Previous Menu เลือกปุ่มหมายเลข 4 เมนุนี้สำหรับกลับไปยังหน้าเมนูหลัก

3.Chassic Control มี 4 เมนูหลักคือ

-Reboot System กดปุ่มหมายเลข 1 สำหรับ รีสตาร์ทเครื่องเซิร์ฟเวอร์

-Factory default กดปุ่มหมายเลข 2 สำหรับ เคลียร์ค่าโพลิซีไฟล์วอลล์ทั้งหมด

-Shutdown System กดปุ่มหมายเลข 3 สำหรับปิดเครื่องเซิร์ฟเวอร์

-Back to Main กดปุ่มหมายเลข 4 สำหรับกลับไปยังหน้าเมนูหลัก

4.Show Policies เลือกปุ่มหมายเลข 4 ใช้ในการแสดงค่าโพลิซีไฟร์วอล์ที่ได้กำหนดไปทั้งหมดและสามารถแทรกโพลิซี ลบโพลิซี ในตำแหน่งใดก็ได้อีกด้วย

-หากต้องการดูโพลิซี กดปุ่มได้สองปุ่มคือ หัวลูกศรทางขวาคือดูโพลิซีถัดไป ส่วนหัวลูกศรทางซ้ายดูโพลิซีก่อนหน้า

-หากต้องการลบโพลิซีนั้นทิ้ง ให้กดปุ่มหมายเลข 0

-หากต้องการแทรกโพลิซีให้กด SET ทำให้เราไปยังเมนู Add Rule Wizard เมนูย่อยของเมนู Policy Setting เมื่อกำหนดค่าโพลิซีที่ต้องการแทรกเสร็จแล้วแล้ว โปรแกรมจะกลับมายังหน้าเมนู Show Policies พร้อมแสดงผลค่าโพลิซีที่แทรกสำเร็จ