การตรวจจับเพื่อป้องกันแพ็กเก็ตในระบบเครือข่าย

(Package sniffer for Packet protection in the network)

โดย

นายวิสวัส	เตชารักษ์	รหัสนิสิต	45360427
นายสราวุธ	ฉิมเฟื่อง	รหัสนิสิต	45360484
นายสุรเชษฐ์	บัวศรี	รหัสนิสิต	45360526

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรมคอมพิวเตอร์

หัวข้อโครงงานวิศวกรรมคอมพิวเตอร์	การตรวจจับเพื่อป้องกันแพ็กเก็ตในระบบเครือข่าย	
ผู้ดำเนินโครงงานวิศวกรรมคอมพิวเตอร์	นาย วิทวัส เตชารักษ์	45360427
	นาย สราวุธ ฉิมเฟื่อง	45360484
	นาย สุรเชษฐ์ บัวศรี	45360256
ที่ปรึกษาโครงงานวิศวกรรมคอมพิวเตอร์	ดร. พนมขวัญ ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2548	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ

(ดร.สุรเชษฐ์ กานต์ประชา)

..... กรรมการ

(อาจารย์จิราพร พุกสุข)

หัวข้อโครงงานวิศวกรรมคอมพิวเตอร์	การตรวจจับเพื่อป้องกันแพ็กเก็ตในระบบเครือข่าย
ผู้ดำเนินโครงงานวิศวกรรมคอมพิวเตอร์	นาย วิทวัส เตชารักษ์ รหัส 45360427
	นาย สราวุธ ฉิมเฟื่อง รหัส 45360484
	นาย สุรเชษฐ์ บัวศรี รหัส 45360256
ที่ปรึกษาโครงงานวิศวกรรมคอมพิวเตอร์	ดร.พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

.....

บทคัดย่อ

อินเตอร์เน็ตเป็นตัวกลางที่เชื่อมโยงคนทั่วทั้งโลกให้สามารถติดสื่อสารได้อย่างรวดเร็ว ทำให้ค้นหาข้อมูลข่าวสาร การศึกษาหาความรู้ เป็นไปอย่างรวดเร็วและทันต่อเหตุการณ์ต่างๆ อินเตอร์เน็ตนั้นใช้โปรโตคอล TCP/IP เป็นพื้นฐานในการติดต่อสื่อสารกัน ด้วยจุดอ่อนของโปรโตคอลซึ่งไม่ได้มีการออกแบบมาเพื่อที่จะรักษาความปลอดภัยและป้องกันตัวเอง ทำให้การบุกรุกระบบเน็ตเวิร์กผ่านทางระบบอินเตอร์เน็ตเป็นไปได้ง่าย

ในการสร้างโปรแกรมตรวจจับเพื่อป้องกันแพ็กเก็ตในระบบเครือข่ายนั้น ได้นำโปรแกรม snort version0.99 เป็นพื้นฐานในการพัฒนาโปรแกรม ซึ่งโปรแกรมที่ได้พัฒนานั้น สามารถตรวจจับแพ็กเก็ตที่มีโปรโตคอล TCP, ICMP และ UDP โดยสามารถกำหนดค่าเพื่อตรวจจับตามจำนวนแพ็กเก็ตที่ต้องการและตรวจจับการรับแพ็กเก็ตในช่วงเวลาที่กำหนด

นอกจากนี้การตรวจจับแพ็กเก็ต ที่มีโปรโตคอล ICMP และ UDP นั้น สามารถทำการตรวจจับการรับส่งแพ็กเก็ตที่มี IP Address หรือ ข้อมูลที่เหมือนกัน และโปรโตคอล ICMP นั้นยังสามารถตรวจจับการส่ง Error Message ได้ ส่วนโปรโตคอล TCP นั้น สามารถตรวจจับการทำงานของ TCP flag ต่างๆ เพื่อตรวจดูความผิดปกติของโปรโตคอล TCP

เมื่อตรวจดูแพ็กเก็ตที่ผิดปกติได้แล้ว ก็สามารถพัฒนาการป้องกันการบุกรุกทางระบบเครือข่ายต่อไป

Project title	Package Sniffer For Packet Protection in the Network	
Name	Mr. Wittawat Tacharuk	ID. 45360427
	Mr. Sarawut Chimfuang	ID. 45630484
	Mr. Surachet Buasri	ID. 45360526
Project Adviser	Dr. Panomkhawn Riyamongkol	
Major	Computer Engineering	
Department	Electrical and Computer Engineering	
Acedamic Year	2005	

.....

ABSTRACT

Nowadays, internet is useful in communicating between people. It is also quick to search for information and get more knowledge. The internet uses TCP/IP protocol to transport. This protocol has weakness in terms of security and protection. So, it is easy to be attacked and intruded.

The Package sniffer program in this project has been developed from snort program version 0.99. The developed program can detect packets which have protocol TCP, ICMP and UDP by counting number of packets and checking number of packets per time.

Moreover, this program can detect sending and receiving the same IP Address or data for protocol ICMP and UDP. Also, it can detect error message for protocol ICMP. In addition, TCP flags can be detected to check the incorrect protocol TCP.

Where the unusual packets can be detected, the protection and security in the network system can be developed furthermore.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้ คงไม่อาจสำเร็จได้หากไม่ได้รับการคำแนะนำและการสนับสนุนจากบุคคลต่าง ๆ ได้แก่ อาจารย์ ดร.พนมขวัญ ริยะมงคล อาจารย์ที่ปรึกษาเป็นผู้ให้การสนับสนุนแนวทางในการดำเนินงานและหลักการที่ต้องใช้ในการดำเนินโครงงาน รวมถึงอาจารย์ทุกท่านที่ได้ให้ความรู้ตลอดการศึกษาที่ผ่านมา และเพื่อน ๆ ที่คอยให้คำปรึกษา แนะนำและช่วยแก้ไขปัญหาที่เกิดขึ้นภายในการทำงาน

พวกข้าพเจ้าขอระลึกถึงพระคุณอันสุดจะประมาณของ ผู้มีพระคุณที่คอยชุบเลี้ยง และส่งเสียให้ข้าพเจ้าได้มีโอกาสเข้ามาศึกษายังที่มหาวิทยาลัยแห่งนี้ บิดามารดา ของพวกข้าพเจ้านั่นเอง

สุดท้ายขอขอบคุณทุกท่านที่เข้ามาหยิบเปิดโครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้ขึ้นมาอ่าน

นาย วิทวัส เตชารักษ์
นาย สราวุธ ฉิมเฟื่อง
นาย สุรเชษฐ์ บัวศรี

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

สารบัญรูป(ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันอินเตอร์เน็ตได้เข้ามามีบทบาทต่อการดำเนินชีวิตเป็นอย่างมาก เพราะสามารถที่จะเชื่อมโยงคนทั้งโลกให้สามารถติดต่อสื่อสาร เป็นที่รวมแหล่งข้อมูลเพื่อแลกเปลี่ยนความรู้ และเทคโนโลยีต่างๆ ตลอดจนมีบทบาทด้านข้อมูลในการตัดสินใจทางด้านธุรกิจ การค้าการลงทุน และการบริการต่างๆ ได้ง่ายและรวดเร็วไม่จำกัดเรื่องของเวลาและสถานที่ ทำให้การทำงานหรือการดำเนินกิจกรรมต่างๆ ดำเนินไปราบรื่น เป็นระบบ ระเบียบ ทันต่อเหตุการณ์ตั้งแต่อดีตจนถึงปัจจุบันและอนาคตต่อไป

อย่างไรก็ตามแม้เทคโนโลยีต่างๆ จะสนองคุณประโยชน์ได้อย่างมหาศาล แต่ก็ไม่ใช่ว่าจะไม่มีโทษเลย เพราะโดยทั่วไปรากฐานของอินเตอร์เน็ตนั้นเป็นการสื่อสารผ่านทางโปรโตคอล TCP/IP ซึ่งโปรโตคอลนี้เดิมออกแบบมาเพื่อการสื่อสารโดยตรง จึงไม่ได้คำนึงถึงเรื่องของความปลอดภัยมากนักเนื่องจากไม่ได้คาดการณ์ว่าอินเตอร์เน็ตจะขยายตัวไปทั่วโลกอย่างรวดเร็ว ด้วยเหตุนี้เองทำให้การบุกรุกระบบเน็ตเวิร์กเป็นไปได้ง่าย ในช่วงแรกๆนั้นผู้บุกรุกสามารถเข้าไปทำลายข้อมูลหรือขโมยข้อมูลได้อย่างไม่ยากนัก แต่เมื่อเวลาผ่านไปได้มีการพัฒนาระบบให้มีความซับซ้อนและมีประสิทธิภาพมากยิ่งขึ้นทำให้การบุกรุกเป็นไปได้ยาก แต่ก็ไม่สามารถป้องกันการก่อกวนของแฮกเกอร์ได้ ถึงแม้จะมีการพัฒนาระบบการป้องกันทำให้บุกรุกให้บุกรุกได้ยากขึ้น แต่จุดบกพร่องTCP/IP ก็ยังเป็นสิ่งที่มีคุณค่ายิ่งสำหรับการก่อกวน หรือมุ่งร้าย และที่สำคัญเน็ตเวิร์กไม่ค่อยมีการพัฒนาขึ้นเลยแม้ในขณะที่มีระบบป้องกันมากมาย แต่เส้นทางในการส่งข้อมูลกลับมีการพัฒนาขึ้นมาอย่างช้า ด้วยเหตุนี้ทำให้เส้นทางการส่งข้อมูลไม่สามารถส่งข้อมูลกันได้ ก็เป็นการก่อกวนอีกอย่างหนึ่งที่ตรวจสอบและป้องกันได้ยากยิ่งนัก

จากที่กล่าวมาเบื้องต้น จะเห็นได้ว่าการมีเทคโนโลยีทุกอย่างดีพร้อม ข้อมูลที่สำคัญขององค์กรของคุณ แม้จะเก็บไว้ในระบบที่มีการป้องกันที่ดีแล้วก็ตาม ทันทีที่ท่านต่อระบบของท่านเข้ากับเครือข่าย เข้ากับอินเตอร์เน็ต จะมั่นใจได้อย่างไรว่ามีความปลอดภัย จะมั่นใจได้อย่างไรว่าระบบที่มีอยู่นั้นทำงานได้ตามปกติ ไม่มีข้อผิดพลาด ถ้าระบบของท่านหยุดนิ่งไประยะเวลาหนึ่ง ธุรกิจของท่านจะเสียหายเท่าไร ท่านเสียผลประโยชน์มหาศาลแค่ไหน ในจุดนี้เองการป้องกันเหตุการณ์ต่างๆที่สามารถคาดคะเนความเป็นไปได้ที่จะเกิดขึ้น ก็จะเป็นการดีไม่น้อยกว่าการที่ต้องมาแก้ไขปัญหาภายหลัง

ดังนั้นโครงงานนี้ได้นำความรู้ทางด้านเครือข่าย การบุกรุกและการโจมตีต่างๆ มาศึกษาเพื่อการป้องกันการโจมตี ในรูปแบบต่าง ๆ เพื่อพัฒนาระบบตรวจจับการบุกรุกทางระบบเครือข่าย ใน

การป้องกันได้อย่างครอบคลุมการก่อกวนเพื่อมุ่งร้ายต่อระบบ และเพื่อเสริมสร้างประสิทธิภาพให้ระบบรักษาความปรอดภัยมีความแข็งแรงมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาการทำงานของโปรโตคอล และข้อบกพร่องของโปรโตคอลที่ใช้ในระบบเน็ตเวิร์ก

1.2.2 เพื่อค้นหาวิธีการในการสำรวจหาการโจมตี

1.2.3 เพื่อหาวิธีการในการป้องกัน และการแก้ไขจากการโจมตี

1.3 ขอบเขตของโครงงาน

1.3.1 ออกแบบและพัฒนาระบบตรวจจับการบุกรุกได้

1.3.2 ระบบสามารถใช้ในระบบเครือข่ายทั่วไปได้

1.3.3 ระบบสามารถตรวจจับ แจ้งเตือน และหาต้นทางการบุกรุกได้

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาความเป็นไปได้ของโครงงานและ กำหนดขอบเขตโครงงาน

1.4.2 ศึกษาการทำงานของโปรโตคอล รวมทั้งหาจุดอ่อนที่เป็นอันตรายต่อระบบ

1.4.3 ศึกษาการตรวจจับแพ็กเก็ตและเลือกเทคนิคที่จะใช้วิเคราะห์แพ็กเก็ต

1.4.4 พัฒนาโปรแกรมเพื่อวิเคราะห์และป้องกันการบุกรุกในรูปแบบต่างๆ

1.4.5 ทดสอบการทำงานของโปรแกรม กับการโจมตีแบบต่างๆ

1.4.6 สรุปผลการทดสอบโปรแกรม และแก้ไขข้อบกพร่องของโปรแกรม

1.4.7 ทดสอบการทำงานของโปรแกรม

1.4.8 จัดทำเอกสารประกอบโครงงาน

1.5 แผนการดำเนินงาน

ตาราง1.1 แผนการดำเนินงาน

กิจกรรม	ระยะเวลาในการดำเนินงาน								
	พ.ย.	ม.ค. - มี.ค.	มี.ค. – พ.ค.	พ.ค. - ก.ค.	ก.ค. – ก.ย.	ก.ย. - พ.ย.	พ.ย. - ม.ค.	ม.ค.-มี.ค.	เม.ย.
	2547	2548	2548	2548	2548	2548	2548-2549	2549	2549
1.กำหนดขอบเขตโครงงาน	←	→							
2. ศึกษาการทำงานของโปรโตคอลรวมทั้งหาจุดอ่อน	←	—	→						
3. ศึกษาการตรวจจับแพ็กเก็ตและเลือกเทคนิคที่จะใช้วิเคราะห์			←	—	→				
4. พัฒนาโปรแกรมเพื่อวิเคราะห์และป้องกันการบุกรุกในรูปแบบต่างๆ						←	—	→	
5.ทดสอบการทำงานของโปรแกรม กับการโจมตีแบบต่างๆ							←	—	→
6. สรุปผลการทดสอบโปรแกรมและแก้ไขข้อบกพร่องของโปรแกรม							←	—	→
7. ทดสอบการทำงานของโปรแกรม								←	→
8. จัดทำเอกสารประกอบโครงงาน								←	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 เข้าใจระบบการทำงานของระบบเครือข่าย TCP/IP และสามารถนำไปปฏิบัติได้

1.6.2 สามารถทำระบบตรวจจับและป้องกันการบุกรุกได้

1.7 งบประมาณของโครงงาน

1.7.1 ค่าอุปกรณ์และค่าเอกสาร	2200	บาท
1.7.2 ค่าหนังสือ	500	บาท
1.7.3 ค่าวัสดุคอมพิวเตอร์	300	บาท
รวม	3000	บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎี

2.1 ความรู้พื้นฐานทั่วไป[8]

2.1.1 บทนำ

ในการตรวจจับการบุกรุกเพื่อป้องกันแพ็กเก็ตที่มีเจตนาที่ไม่ดีหรือมุ่งร้ายที่จะบุกรุกระบบเครือข่าย ทำให้ต้องมีการศึกษาหลักการและทฤษฎีต่างๆ เกี่ยวกับระบบการตรวจจับการบุกรุก กิจกรรมต่างๆบนเน็ตเวิร์ก แนะนำความรู้เบื้องต้นเกี่ยวกับ TCP/IP ที่รวมไปถึงวิธีการรับส่งข้อมูล ข้อบกพร่อง และแนวทางในการรักษาความปลอดภัยต่างๆของข้อมูลและระบบเครือข่าย ซึ่งในการตรวจจับแพ็กเก็ตที่มีพฤติกรรมแปลกๆ หรือเข้าข่ายการบุกรุกนั้นต้องมีความรู้ในเรื่องต่างๆ เกี่ยวกับการรับ-ส่งข้อมูล ของ TCP/IP ที่เป็นประโยชน์ก่อนนำมาวิเคราะห์การทำงานต่อไป

2.1.2 ระบบตรวจจับการบุกรุก(Intrusion Detection System)

Intrusion Detection คือ การจำแนกการตอบสนองต่อกิจกรรมที่เกิดขึ้นในระบบเน็ตเวิร์กที่มุ่งร้ายต่อระบบเครือข่ายและทรัพยากรที่อยู่บนเน็ตเวิร์กนั้น ดังนั้น Intrusion Detection System (IDS) เปรียบเสมือนยามคอยตรวจตราความเป็นไปและพฤติกรรมของข้อมูลที่ผ่านเข้าในเน็ตเวิร์กว่าน่าสงสัยหรือมีสิ่งผิดปกติหรือไม่

กิจกรรมในระบบเน็ตเวิร์กที่เกิดขึ้นในแต่ละวันนั้นมีจำนวนนับไม่ถ้วน มีข้อมูลที่ส่งผ่านไปมาวนเวียนบนระบบเน็ตเวิร์กมากมาย บริการต่าง ๆ ที่อยู่บนอินเตอร์เน็ตที่ใช้งานอยู่ทุกวันนั้นต่างก็อาศัยเน็ตเวิร์กเป็นช่องทางการสื่อสารแทบทั้งสิ้น เป็นการดีไม่น้อยที่นำ IDS มาช่วยตรวจสอบเรียบร้อยของระบบเครือข่าย

2.1.3 กิจกรรรมบนระบบเน็ตเวิร์ก

2.1.3.1 ข้อสนเทศ (Information)

ข้อสนเทศเป็นข้อมูลที่ประสงค์ให้สามารถรับรู้ความหมายได้ มีหลากหลายรูปแบบ เช่น ตัวอักษร ข้อความ รูปภาพ ภาพเคลื่อนไหว เสียง ข้อสนเทศต่างๆที่สื่อสารกันบนเน็ตเวิร์กนั้นได้ถูกเปลี่ยนให้อยู่ในรูปของข้อมูลอิเล็กทรอนิคส์ก่อนการสื่อสาร และเมื่อผู้รับปลายทางได้รับข้อมูลเหล่านั้นแล้วก็จะแปลงกลับเป็นรูปแบบเดิม ข้อสนเทศเหล่านี้จะไม่สามารถสื่อความหมายได้หากไม่ได้ถูกนำไปแปลงให้อยู่ในรูปแบบที่ถูกต้อง อาทิเช่นหากนำข้อมูลภาพถ่ายไปแปลงเป็นข้อความก็จะอ่านไม่รู้เรื่อง ดังนั้นข้อสนเทศนั้นก็คือแอพพลิเคชั่น โดยมีบราวเซอร์ที่ทำหน้าที่แปลงข้อมูลสื่อผสมทั้งภาพ ข้อความ และ เสียงกลับมาอยู่ในรูปแบบที่ถูกต้องตรงกับที่ผู้ส่งต้องการและผู้ใช้สามารถเข้าใจข้อสนเทศนั้นได้

กิจกรรมที่ใช้ไปเพื่อการสื่อสารข้อสนเทศโดยระบบเน็ตเวิร์กเป็นสื่อกลางในการส่งข้อมูลจากต้นทางไปยังปลายทาง โดยที่รูปแบบการรับส่งข้อสนเทศนั้นจะขึ้นอยู่กับการตกลงและข้อกำหนดของโปรโตคอลที่ใช้ ซึ่งโปรโตคอลที่อยู่ในระดับสูง ๆ ที่เป็นที่รู้จักกันดี เช่น

- FTP เป็นการรับ – ส่ง แฟ้มข้อมูลระหว่างคอมพิวเตอร์
- HTTP เป็นการส่งข้อสนเทศในลักษณะที่เป็นไฮเปอร์เท็กซ์ซึ่งสามารถนำไปแสดงผลเป็นข้อความ หรือรูปภาพให้สามารถสื่อสารกันได้ระหว่างไคลเอ็นต์ที่เป็นบราวเซอร์กับเซิร์ฟเวอร์
- SMTP , POP ใช้ในการรับและส่งจดหมายอิเล็กทรอนิคส์จากผู้ส่งต้นทางไปยังกล่องจดหมายบนเซิร์ฟเวอร์ปลายทาง

2.1.3.2 สัญญาณควบคุม (Control Signal)

ข้อมูลส่วนที่เป็นสัญญาณควบคุมมีความสำคัญต่อการสื่อสารข้อมูลมาก เพราะเป็นส่วนหนึ่งของโปรโตคอล ผู้ใช้ทั่วไปจะไม่ค่อยมีโอกาสเกี่ยวข้องกับส่วนนี้ เพราะกลไกของการสื่อสารจะเป็นผู้จัดการสัญญาณพวกนี้ หน้าที่ของสัญญาณควบคุมข้อมูลส่วนนี้จะมีอยู่ 2 ประการ คือ

- ควบคุมจังหวะการรับส่งข้อมูล ลำดับการส่งนั้นจะส่งส่วนที่เป็นควบคุมไปก่อน เรียกว่า เฮดเดอร์ (Header) แล้วจึงส่งส่วนที่เนื้อข้อมูลตามไป ในการแยกระหว่าง เฮดเดอร์ กับส่วนของข้อมูล โดยใช้ความยาวของข้อมูลเป็นตัวกำหนด เฮดเดอร์ส่วนใหญ่จะมีความยาวคงที่ หรือหากไม่คงที่ก็จะระบุในเฮดเดอร์ว่ามีความยาวเท่าไร การนำข้อมูลไปใช้อย่างถูกต้องนั้นต้องอ่านข้อมูลจากเฮดเดอร์มาประกอบด้วยเสมอ เช่น *ความยาวของข้อมูล* หากผู้รับไม่รู้ว่าข้อมูลมีความยาวเท่าไรก็จะไม่สามารถถอดได้อย่างถูกต้อง , *ลำดับของข้อมูล* บางครั้งข้อมูลที่ได้รับมาอาจจะกระจัดกระจายและต้องนำมาประกอบรวมกันใหม่เพื่อให้ข้อมูลสมบูรณ์ หากลำดับของข้อมูลไม่ถูกต้องการประกอบรวมข้อมูลเหล่านั้นก็จะมีการล้มเหลว

โปรโตคอล TCP/IP ประกอบด้วยส่วนของสัญญาณควบคุมมากมายในทุก ๆ เลเยอร์ เพื่อทำหน้าที่ควบคุมและประสานงานในการสื่อสารข้อมูลตั้งแต่ในระดับล่างสุดคือ

- Link Layer : PPP Header , Ethernet Header
- Internet Layer : IP Header
- Transport Layer : TCP Header , UDP Header

- การสั่งงานให้อุปกรณ์ในเน็ตเวิร์กกระทำการอย่างหนึ่งอย่างใด ข้อมูลบางชนิดในบางโปรโตคอลยังสามารถสั่งงานให้อุปกรณ์ในเน็ตเวิร์ตทำงานได้ อาทิเช่นการสั่งให้เราต์เตอร์เปลี่ยนเส้นทางข้อมูล , การตรวจสอบสถานะของโฮสต์ปลายทาง , การตรวจสอบเส้นทางเดินของข้อมูล เป็นต้น สิ่งสำคัญที่จะชี้ให้เห็นคุณสมบัติของข้อมูลประเภทสัญญาณควบคุมนี้คือ ระบบเน็ต

เวิร์กมิใช่เป็นแค่ทางผ่านของข้อมูล และข้อมูลมิใช่เป็นเพียงสิ่งที่ได้เป็นสิ่งที่เดินทางโดยอาศัยระบบเน็ตเวิร์กเป็นสื่อกลาง แต่ทั้งสองสิ่งทำหน้าที่สอดคล้องกันและมีผลกระทบต่อกัน

2.1.4 แนวความคิดพื้นฐานของระบบตรวจจับการบุกรุก

ข้อมูลที่สามารถมองเห็นได้โดยผ่านแอพพลิเคชั่นจะเป็นข้อมูลที่รับกันตามปกติ ถูกต้องตามโปรโตคอลทุกประการ เพราะหากที่ส่วนใดส่วนหนึ่งของข้อมูลนั้นเกิดผิดพลาด ไม่เป็นไปตามโปรโตคอล แล้วข้อมูลนั้นก็จะไม่แสดงให้เห็นแต่อย่างใดซึ่งแค่จุดอ่อนนี้เพียงจุดเดียวก็สามารถนำไปใช้ในการ DoS (Denial of Service) หรือการก่อกวนเป้าหมายได้นั่นเอง

หากเปรียบระบบหรือคอมพิวเตอร์เสมือนเป็นบ้านที่ไม่มีประตู ระบบปฏิบัติการ และแอพพลิเคชั่นในเป็นเจ้าของบ้าน และข้อมูลที่สื่อสารไปมาบนเน็ตเวิร์กก็จะเป็นเหมือนคนเดินถนนทั่วไป เมื่อบ้านไม่มีประตูใครที่อยู่ข้างนอกอยากจะเข้ามาในบ้านก็เดินเข้าตามปกติ เจ้าบ้านมีหน้าที่เดินสอบถามทุกคนที่เข้ามาเพื่อให้ทราบว่าเป็นคนที่ต้องการติดต่อด้วยหรือไม่ หากไม่ใช่ก็จะบอกให้เขากลับ หากใช่ก็จะเชื้อเชิญมา อย่างไรก็ตามเจ้าของบ้านก็ไม่สามารถจะห้ามไม่ให้คนอื่นเดินเข้ามาและไล่คนที่ไม่ต้องการออกไปได้

โดยส่วนใหญ่แอพพลิเคชั่นที่ให้บริการจะถูกออกแบบเพื่อบริการให้ดีที่สุด ดังนั้นการแกล้งเจ้าของบ้านก็ทำได้ไม่ยากนัก เช่น ส่งคนเข้าไปในบ้านที่เดียวพร้อม ๆ กันหลาย ๆ คนจนเจ้าของบ้านไม่มีเวลาที่จะไปทำงานอื่น ๆ (Ping Flood), ส่งคนเข้ามาในบ้านแต่พอเจ้าของบ้านถามอะไรก็ไม่ยอมตอบปล่อยให้คอยคำตอบอยู่ (Syn Flood), ส่งคนหลาย ๆ แบบมาหาเจ้าของบ้านเพื่อสืบว่าเจ้าของบ้านยินดีต้อนรับคนประเภทไหน (Port Scanning) เป็นต้น เมื่อแอพพลิเคชั่นไม่ได้ถูกออกแบบมาให้ระวังเรื่องความปลอดภัย หากถูกก่อกวนมาก ๆ จนไม่สามารถรับมือได้ก็จะหยุดทำงานในที่สุด

ดังนั้น IDS ก็จะเสมือนยามรักษาการณ์ที่ทำหน้าที่เป็นผู้ช่วยเจ้าของบ้าน กล่าวคือจะช่วยดูจากลักษณะของคนนั้น ๆ และรู้พฤติกรรมของการก่อกวนเป็นอย่างดี หากมีพฤติกรรมต้องสงสัยก็จะรีบรายงานให้เจ้าของบ้านทราบทันที IDS มีโอกาสที่จะรายงานผิดพลาด หากการใช้งานปกตินั้นใกล้เคียงกับพฤติกรรมการบุกรุกก็จะรายงานว่าเป็นเช่นเดียวกับการบุกรุกได้

นับว่า IDS เป็นเครื่องมือที่สำคัญต่อการรับมือกับการบุกรุกทุกประเภทจากแฮกเกอร์ แต่ถึงแม้ว่า IDS จะมีความสามารถมากเพียงใด ก็มิใช้เครื่องมือสำเร็จรูปที่จะสามารถตรวจจับการบุกรุกได้ถูกต้องทั้งหมด นอกจากนี้ IDS ยังเป็นเครื่องมือที่จะวิเคราะห์เทคนิคที่ใช้ในการบุกรุกและหาแนวทางป้องกันนั้นจะต้องใช้คนที่มีความสามารถรู้ด้านระบบเน็ตเวิร์กเป็นอย่างดี จึงสามารถจำแนกและตีความสิ่งที่ตรวจจับจาก IDS มาเป็นผลในทางป้องกันได้จริง IDS มีข้อดีและข้อเสียดังนี้ คือ

2.1.4.1 ข้อดีของการใช้งาน IDS

2.1.4.1.1 การตอบสนองทันทีทันใด

การวิเคราะห์การบุกรุกนั้นทำได้โดยการใช้เครื่องมือทำการจัดเก็บบันทึกข้อมูลที่มีการสื่อสารกันบนเน็ตเวิร์ก และนำข้อมูลที่ได้มาวิเคราะห์ แต่การวิเคราะห์ลักษณะดังกล่าว จะกระทำได้ก็ต่อเมื่อได้เกิดเหตุการณ์ไปแล้ว จะเป็นการวิเคราะห์ข้อมูลย้อนหลัง ซึ่ง IDS จะช่วยแก้ไขข้อบกพร่องในส่วนนี้ คือสามารถตรวจจับได้ทันทีที่มีความผิดปกติเกิดขึ้น และช่วยให้ทำการแก้ได้ทันท่วงที และทำงานโดยอัตโนมัติและทำงานอยู่ตลอดเวลา

2.1.4.1.2 การมีรากฐานความรู้ของการวิเคราะห์

เทคนิคและกลวิธีในการบุกรุกหรือก่อกวนนั้นได้พัฒนาขึ้นทุกวัน วิธีการตรวจจับและวิเคราะห์จำเป็นต้องพัฒนาตามให้สอดคล้องกันจึงจะตรวจจับได้อย่างมีประสิทธิผล โดย IDS นั้นสามารถช่วยแบ่งเบาภาระของนักวิเคราะห์ลงได้มาก โดยหากรู้รูปแบบพฤติกรรมแน่ชัดว่าเป็นการมุ่งร้ายก็ให้จัดเก็บข้อมูลรูปแบบเหล่านั้นใน IDS เมื่อมีกิจกรรมดังกล่าวเกิดขึ้นอีกในเน็ตเวิร์ก IDS ก็จะสามารถตรวจพบได้ทันที และเมื่อค้นพบรูปแบบใหม่ก็จัดเก็บลงใน IDS อีก ทำให้ IDS เสมือนมีฐานความรู้ในการวิเคราะห์การบุกรุกได้ดีในระดับหนึ่ง และขีดความสามารถก็จะเพิ่มขึ้นเรื่อย ๆ

2.1.4.1.3 การช่วยตรวจสอบข้อบกพร่องของระบบป้องกันอื่น ๆ

เน็ตเวิร์กของผู้ใช้อาจมีการป้องกันการบุกรุกอยู่โดยใช้ ไฟร์วอลล์ (Firewall) แต่อย่างไรก็ตามไฟร์วอลล์มิใช่เครื่องมือที่จะป้องกันการบุกรุกได้อย่างอัตโนมัติ จะต้องอาศัยผู้ที่บริหารระบบกำหนดกฎให้เหมาะสมกับการใช้งาน และถึงแม้จะมีการตั้งกฎที่เหมาะสมแล้วก็ตาม แต่กฎเหล่านั้นอาจไม่สามารถป้องกันการบุกรุกได้ การบริหารไฟร์วอลล์ที่ดีก็ควรมีการตรวจสอบย้อนหลัง (Audit) และการทดสอบการเจาะระบบ (Penetration test)

IDS สามารถช่วยได้มาก โดยติดตั้ง IDS ไว้หลังไฟร์วอลล์ และทำการทดสอบเจาะระบบด้วยวิธีต่าง ๆ เพื่อดูว่ามีเทคนิคใดที่สามารถเจาะผ่านไฟร์วอลล์ได้บ้าง และหากมีแพ็กเก็ตใดผ่านเข้าไปได้ IDS ก็จะตรวจพบ ทำให้ผู้บริหารระบบสามารถปรับปรุงให้รัดกุมมากขึ้น

2.1.4.2 ข้อเสียของการใช้งาน IDS

2.1.4.2.1 การละเมิดความเป็นส่วนบุคคล

เนื่องจาก IDS มีพื้นฐานจากการนำข้อมูลทั้งหมดที่สื่อสารกันมาทำการวิเคราะห์ ซึ่งข้อมูลเหล่านั้นจะต้องครอบคลุมถึงข้อมูลทั่วไปที่มีการสื่อสารกันตามปกติ และมีการที่จะทราบว่ามีความผิดปกติหรือไม่นั้นก็จะต้องอ่านข้อมูลทั้งหมดด้วย ดังนั้นไม่ว่าจะมีกิจกรรมใด ๆ ที่เกิดขึ้นในเน็ตเวิร์ก ก็สามารถถูกเปิดอ่านได้จาก IDS นั้นหมายความว่า IDS เปรียบเสมือนการที่ตำรวจดักฟังโทรศัพท์

2.1.4.2.2 การตอบโต้อัตโนมัติ

IDS ที่มีจำหน่ายอยู่ในท้องตลาดจะมีส่วนหนึ่งที่ให้ผู้ใช้งานสามารถกำหนดการดำเนินการอย่างใดอย่างหนึ่งเมื่อตรวจพบการบุกรุกเกิดขึ้น เช่น ส่งจดหมายเตือนผู้ดูแลระบบ, ส่งไปยังไฟวอลล์เพื่อจำกัดการเข้าออกของข้อมูล และสิ่งสำคัญที่สุดอาจจะส่งผลเสียหายอันใหญ่หลวงต่อเจ้าของได้ก็คือการโจมตีกลับไปยังต้นกำเนิดการบุกรุก โดยที่ใช้วิธีการของ IDS ในโลกความเป็นจริงนั้นไม่สามารถตัดสินใจได้ว่าใครเป็นแฮกเกอร์ได้อย่างง่ายได้ และในเวลาอันรวดเร็ว การที่กำหมดให้ IDS ทำการตอบโต้กลับไปในทันทีโดยมีข้อมูลเพียงผิวเผินนั้น นอกจะไม่ช่วยให้ระบบเน็ตเวิร์กปลอดภัยแล้ว ยังอาจจะทำให้ระบบเครือข่ายนั้นกลายเป็นแฮกเกอร์ที่คอยโจมตีผู้อื่นเอง

2.1.4.2.3 การเตือนภัยผิดพลาด

IDS ได้ถูกกำหนดให้ตรวจจับกิจกรรมประเภทดังกล่าวแล้ว ก็จะมีการเตือนทันทีที่ตรวจพบ และเป็นหน้าที่ของนักวิเคราะห์ระบบที่จะทำการสืบค้นข้อมูลต่าง ๆ มาประกอบการวินิจฉัยอีกครั้ง และ IDS ถูกกำหนดให้มีความไวเป็นพิเศษมักจะสามารถตรวจจับพฤติกรรมที่ก้ำกึ่งนั้นได้มากเป็นพิเศษ นอกจากนี้การปล่อยให้ IDS มีการเตือนอย่างไม่เหมาะสมจะทำเกิดข้อมูลในลักษณะที่เป็นการบุกรุกจริงและการเตือนผิดพลาดผสมกันอยู่

2.2 ความรู้พื้นฐานเกี่ยวกับการตรวจจับการบุกรุก[8]

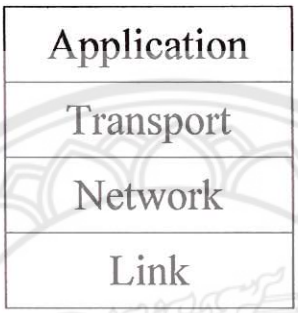
2.2.1 ความรู้เบื้องต้นเกี่ยวกับ TCP/IP

โปรโตคอล TCP/IP เป็นชุดของโปรโตคอลที่มีการพัฒนามาตั้งแต่ปี 1960 โดยมีวัตถุประสงค์ให้สามารถใช้สื่อสารจากต้นทางผ่านเน็ตเวิร์กไปยังปลายทาง และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ในระยะแรกเริ่มใช้โปรโตคอลนี้ในวงแคบๆ เฉพาะทาง

ราชการ จนในช่วงปี 90 จึงนำมาใช้ในทางธุรกิจและเป็นจุดเริ่มต้นของอินเตอร์เน็ตในปัจจุบัน แต่อย่างไรก็ตามโปรโตคอลนี้ยังมีข้อบกพร่องอีกมากมาย และด้วยข้อบกพร่องนี้กลับกลายเป็นเครื่องมือในการโจมตีอย่างดีของเหล่า แฮกเกอร์

2.2.2 การแบ่งชั้น (Layering)

TCP/IP เป็นชุดโปรโตคอลที่ประกอบด้วยโปรโตคอลย่อยหลายตัว โดยแต่ละตัวจะทำหน้าที่ในแต่ละชั้นหรือเลเยอร์ ซึ่งรับผิดชอบและแปลความหมายของข้อมูลในแต่ละระดับของการสื่อสาร ซึ่งในภาพรวมแล้วสามารถแบ่งเป็น 4 เลเยอร์ ดังรูป



รูปที่ 2.1 แสดงเลเยอร์ของTCP/IP [8]

2.2.2.1 หน้าที่ความรับผิดชอบของแต่ละเลเยอร์มีดังนี้

- Link layer ในเลเยอร์นี้จะเป็นดีไวซ์ไดรเวอร์ที่ทำงานอยู่บนระบบปฏิบัติการแต่ละระบบทำหน้าที่รับผิดชอบในการรับส่งข้อมูลตั้งแต่กายภาพ, สัญญาไฟฟ้า จนถึงการแปลความจากระดับสัญญาณไฟฟ้าจนเป็นข้อมูลทางคอมพิวเตอร์, โปรโตคอลระดับนี้ เช่น Ethernet และ SLIP (Serial Line Internet Protocol)

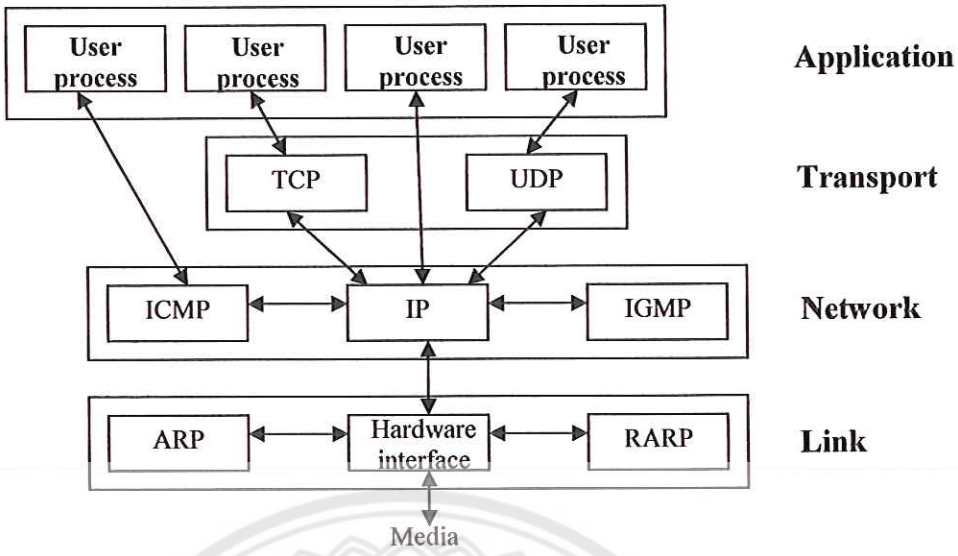
- Network Layer รับผิดชอบในการรับ – ส่งข้อมูลในเน็ตเวิร์ก ส่งผลข้อมูลไปจนถึงจุดหมายปลายทาง โปรโตคอลนี้ได้แก่ IP, ICMP

- Transport Layer รับผิดชอบในการรับส่งข้อมูลระหว่างเครื่องหนึ่ง ไปยังอีกเครื่องหนึ่ง และจะส่งข้อมูลขึ้นไปให้ Application layer นำไปใช้งานต่อ มีโปรโตคอลที่จัดอยู่ในเลเยอร์นี้คือ TCP และ UDP ซึ่งมีลักษณะในการรับส่งข้อมูลที่แตกต่างกันออกไป

- Application Layer เป็นเลเยอร์ที่แอพพลิเคชั่นเรียกใช้โปรโตคอลระดับล่าง ๆ ลงไปเพื่อวัตถุประสงแตกต่างกัน เช่น

2.2.2.2 TCP/IP Layering

ชุดของโปรโตคอล TCP/IP ประกอบด้วยโปรโตคอลหลายตัวทำงานร่วมกัน โดยเลเยอร์ ต่าง ๆ และมีหน้าที่แตกต่างกันออกไป ดังภาพ



รูปที่ 2.2 แสดงเลเยอร์ของโปรโตคอลต่าง ๆ ในชุดTCP/IP [8]

จากภาพที่แสดงให้เห็นถึงโปรโตคอลแต่ละเลเยอร์ที่เมื่อรวมกันเป็นชุด TCP/IP

TCP : อยู่ในทรานสปอร์ตเลเยอร์ ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลให้มีความเสถียรภาพและเชื่อถือได้

UDP : อยู่ในทรานสปอร์ตเลเยอร์ ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลเช่นเดียวกันแต่ไม่มีกลไกการรับส่งที่มีเสถียรภาพและเชื่อถือได้ โดยปล่อยหน้าที่นี้ให้กับแอพพลิเคชั่นเป็นตัวทำหน้าที่นี้แทน

IP : อยู่ในเน็ตเวิร์กเลเยอร์ เป็นโปรโตคอลหลักในการสื่อสารข้อมูล ซึ่งกลไกที่สำคัญที่ทำให้สามารถเคลื่อนที่ไปยังปลายทางได้ดีก็คือโปรโตคอล IP

ICMP: (Internet Control Message Protocol) อยู่ในเน็ตเวิร์กเลเยอร์ ทำหน้าที่เสริมให้การทำงานของ IP ให้สมบูรณ์ โดยจะเป็นโปรโตคอลที่คอยส่งข่าวสารและแจ้งความผิดพลาดให้แก่ IP แต่ในบางโอกาสแอพพลิเคชั่นเลเยอร์ก็เรียกใช้ ICMP โดยตรงเพื่อใช้ประโยชน์จากความสามารถของ ICMP ด้วยเช่นกัน

IGMP: (Internet Group Management Protocol) อยู่ในเน็ตเวิร์กเลเยอร์ ทำหน้าที่ในการส่ง UDP ดาต้าแกรมไปยังกลุ่มของโฮสต์ หรือโฮสต์หลาย ๆ ตัวพร้อมกัน

ARP : (Address Reservation Protocol) อยู่ในลิงค์เลเยอร์ ทำหน้าที่เปลี่ยนระหว่างแอดเดรสที่ใช้โดย IP ให้เป็นแอดเดรสของ Network Interface

RARP:(Reverse ARP) อยู่ในลิงค์เลเยอร์เช่นกัน แต่ทำหน้าที่กลับกันกับ ARP คือเปลี่ยนระหว่างแอดแดรสของ Network Interface ให้เป็นแอดเดรสที่ใช้โดย IP

2.2.2.3 Internet Address

ทุกอินเตอร์เฟซที่ต่ออยู่บนอินเตอร์เน็ตจะมีเลขประจำตัวเพื่อใช้การสื่อสารข้อมูลที่เรียกว่า Internet Address หรือเรียกย่อ ๆ ว่า IP Address โดย IP Address นี้เป็นหมายเลขจำนวน 32 บิต แต่ละที่จะกำหนดให้เลขทั้ง 32 บิต นั้นถูกนับต่อเนื่องกันไป ตั้งแต่ 0 – 2^{32} โดยทำการแบ่งออกเป็นขนาด 8 บิต จำนวน 4 ชุด และคั่นแต่ละชุดด้วยจุด ดังตัวอย่าง 192.168.133.255

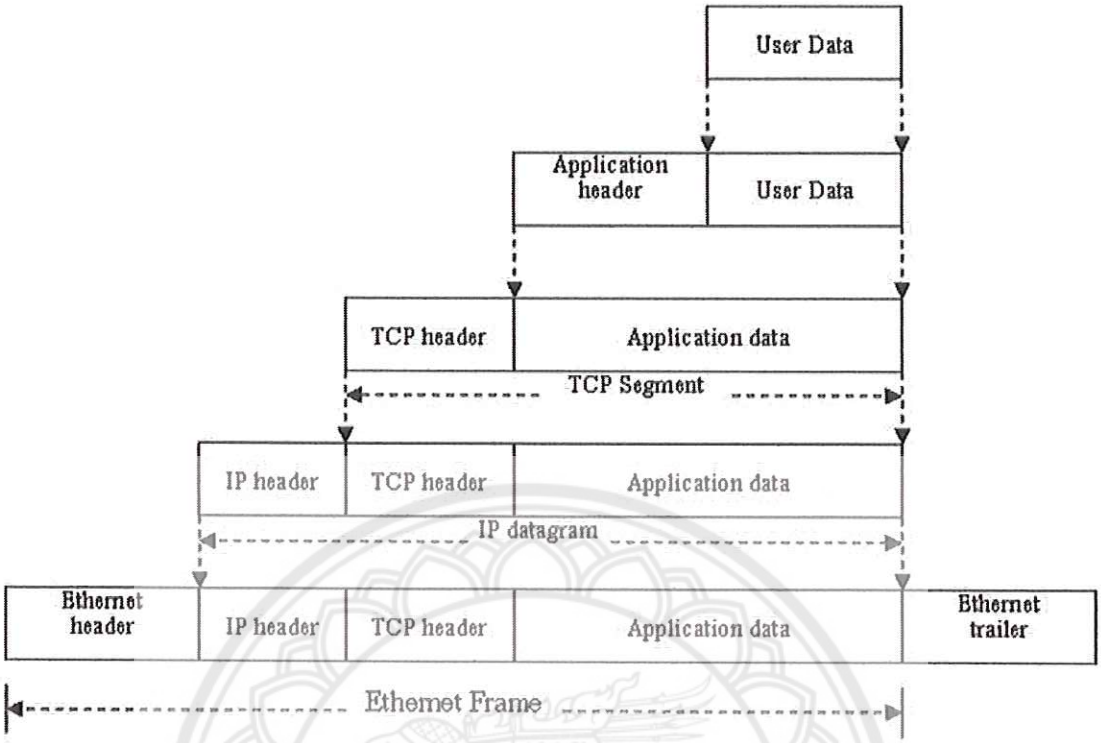
Class	Range
A	0.0.0.0 – 127.255.255.255
B	128.0.0.0 – 191.255.255.255
C	192.0.0.0 – 223.255.255.255
D	224.0.0.0 – 239.255.255.255
E	240.0.0.0 – 255.255.255.255

รูปที่ 2.3 แสดง Rang ของ IP [8]

2.2.2.4 การ Encapsulation

การ Encapsulate คือการนำข้อมูลที่ต้องการส่งมารวมกับข้อมูลที่เป็นส่วนควบคุมของโปรโตคอล โดยข้อมูลส่วนที่เป็นส่วนควบคุมนั้นจะถูกนำมาไว้ในส่วนหัวของข้อมูล เรียกว่าเฮดเดอร์ (Header) ซึ่งในการรับข้อมูลนั้นผู้ที่รับข้อมูลจะได้รับเฮดเดอร์ก่อนจากนั้นก็นำไปแปลจะทราบว่าข้อมูลที่ตามมานั้นมีลักษณะอย่างไร

ส่วนประกอบของเฮดเดอร์ของโปรโตคอล แอดเดรสต้นทาง , แอดเดรสปลายทาง , ความยาวข้อมูล , รหัสตรวจสอบความผิดพลาดข้อมูล ซึ่งสิ่งที่จะต้องเน้นให้เห็นชัดก็คือ จะมีข้อมูลสำคัญเฉพาะโปรโตคอลที่ทำการ Encapsulate มาเท่านั้น



รูปที่ 2.4 แสดงการ Encapsulation ข้อมูลผ่านชั้นของโปรโตคอลแต่ละระดับ[8]

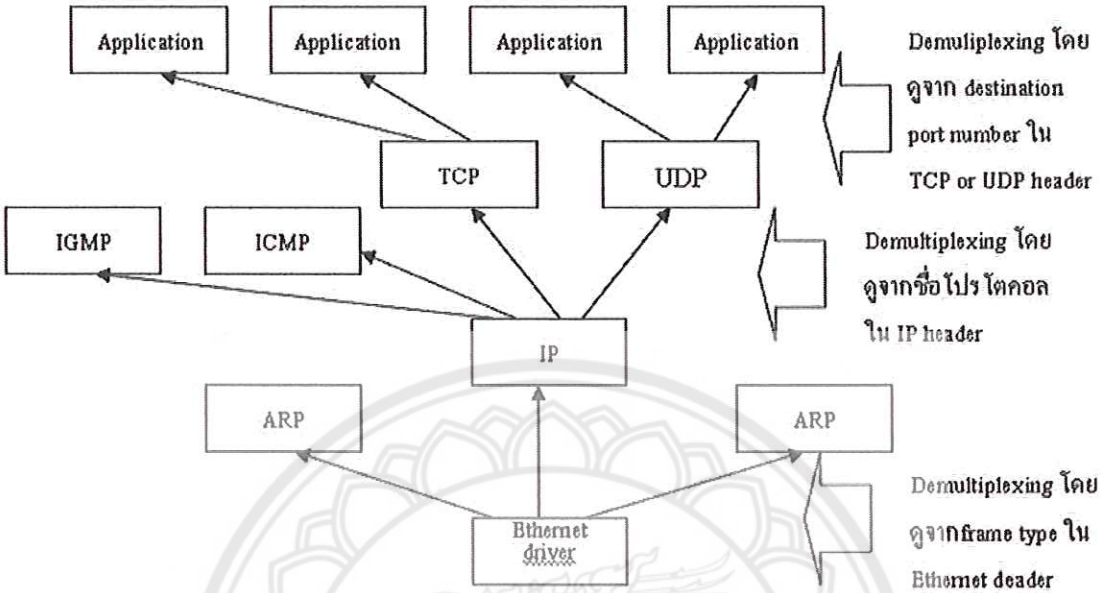
ในการรับส่งข้อมูลนั้น ข้อมูลที่รับส่งกันจริง ๆ บนเน็ตเวิร์กนั้นจะประกอบด้วย 2 ส่วนคือ ข้อมูลจริงกับข้อมูลของโปรโตคอล เปรียบเสมือนการส่งจดหมายซึ่งจะต้องประกอบด้วยเนื้อหาและซองจดหมายที่เขียนที่อยู่ ติดแสตมป์ ถ้ามีแต่จดหมายอย่างเดียวไปรษณีย์ก็ไม่รู้ว่าจะส่งให้ใคร การ Encapsulate ก็คือการเอาจดหมายมาใส่ และฝ่ายที่รับข้อมูลก็จะต้องแกะซองออกตามลำดับ โดยแกะซอง Ethernet ก่อน แล้วจะเจอซอง IP แกะซอง IP จะเจอซอง TCP แกะซอง TCP ก็จะเจอข้อมูลที่ต้องการ

การ Encapsulate ในแต่ละระดับก็จะมีการเรียกข้อมูลที่อยู่ในซองแตกต่างกันออกไป ข้อมูลที่ทำการ Encapsulate เรียบร้อย แล้วจาก TCP ส่งไปยัง IP เรียกว่า TCP Segment, ในระดับ IP ก็จะถือว่า TCP Segment เป็นข้อมูลทั้งชุด เมื่อไปรวมกับ IP Header ส่งไปยังเลเยอร์ Datalink จะเรียกว่า IP Datagram, ในระดับ Datalink เมื่อส่งลงไปจะนำ IP Datagram มาใส่ซอง ขนาดของข้อมูลทั้งหมดเรียกว่า Ethernet Frame

2.2.2.5 การ Dumultiplexing

Dumultiplexing คือกระบวนการย้อนกลับของ Encapsulation การทำ Encapsulation คือ การนำข้อมูลมาใส่ซองทีละชั้นตามเลเยอร์ที่ส่งไป การ Demultiplexing ก็คือการรับซองข้อมูลที่ปิดผนึกใส่ซองมาอย่างมิดชิด เพื่อทำการแกะออกทีละชั้นตามเลเยอร์จนถึงบนสุด คือ แอพพลิเคชั่นเลเยอร์ จึงได้ข้อมูลเนื้อความจริง ๆ ที่ต้องการสื่อสารกันในการ Demultiplexing นั้น

แต่ละเลเยอร์จะนำข้อมูลมารวมกันให้ครบตามขนาดที่ต้องการ ซึ่งในที่สุดท้ายก็จะได้รับเฉพาะข้อมูลเท่านั้น และเฮดเดอร์ถูกถอดออกไปหมด



รูปที่ 2.5 แสดง Dumultiplexing [8]

การDemultiplex และ Encapsulate เป็นส่งที่คู่กันและสอดคล้องกัน อุปกรณ์ที่จะสื่อสารบนเน็ตเวิร์กได้จะต้องมีทั้งส่วนที่ทำหน้าที่ทั้งสอง โดยการ Demultiplexing ใช้ในตอนที่รับข้อมูลจากเน็ตเวิร์ก และการ Encapsulate ใช้ในตอนที่จะทำการส่งข้อมูลอยู่ในแต่ละเลเยอร์ของโปรโตคอล

2.2.2.6 Port Number

เลเยอร์บนสุดของ TCP/IP คือ แอพพลิเคชั่นเลเยอร์ แต่ละแอพพลิเคชั่นจะสามารถแยกได้หลายอย่าง ในความเป็นจริงที่ใช้งานปัจจุบันเองก็มีอยู่มากที่มีแอพพลิเคชั่นมากกว่า 1 แอพพลิเคชั่น ที่ทำงานอยู่ในเครื่องเดียวกัน เช่นในเซิร์ฟเวอร์เครื่องเดียวอาจจะเป็นทั้ง FTP server, Web server และ Mail server เป็นต้น [ภาคผนวก ก.]

พอร์ต (port) จะเป็นปัญหาของคำถามข้างต้น ในโปรโตคอล TCP/IP ได้ถูกออกแบบให้มีหมายเลขพอร์ตอยู่ในเฮดเดอร์ เพื่อระบุว่าข้อมูลเซกเมนต์นี้เป็นแอพพลิเคชั่นอะไร ในโอสต์นั้น แอพพลิเคชั่นแต่ละตัวที่ให้บริการอยู่ในเครื่องต่างจะมีหมายเลขพอร์ต เพื่อจะสามารถเลือกนำข้อมูลมาใช้ว่าเป็นแอพพลิเคชั่นของตนเองหรือไม่ หมายเลขพอร์ตที่ใช้เป็นมาตรฐาน ได้แก่ พอร์ต 20 , 21 เป็นของ FTP, พอร์ต 23 Telnet, พอร์ต 25 SMTP และพอร์ต 80 HTTP เป็นต้น [ภาคผนวก ก.]

2.2.2.7 Reserved Port

ในระบบปฏิบัติการ Unix มีการสงวนพอร์ตบางส่วนไว้ให้สำหรับโพรเซสที่มีสิทธิพิเศษของ Super user เท่านั้นที่สามารถใช้พอร์ตในช่วง 1 – 1023 ได้ แต่สำหรับ Window NT มิได้สงวนไว้แต่อย่างใด ทั้ง TCP และ UDP ต่างก็ใช้งานพอร์ตลักษณะเดียวกันคือ ใช้ระบุแอพพลิเคชั่น หมายเลขพอร์ตที่สามารถระบุได้จะเป็นข้อมูลชุดขนาด 16 บิต นั่นหมายความว่าสามารถมีพอร์ตที่สามารถใช้งานได้ทั้งสิ้น = 65535 พอร์ต ในแต่ละโปรโตคอล ดังนั้นจำนวนพอร์ตทั้งหมดนั้นสามารถใช้งานได้เมื่อใช้โปรโตคอล TCP/IP คือ 128K นั่นเอง โดยเป็นของ TCP = 64K และของ UDP อีก 64 K

2.2.2.8 IP : Internet Protocol

IP เป็นโปรโตคอลที่ทำหน้าที่ในการนำข้อมูลไปส่งยังจุดหมายไม่ว่าที่ใดในอินเตอร์เน็ต โปรโตคอลต่าง ๆ ใน TCP/IP ทั้ง TCP, UDP และ ICMP ต่างก็ต้องอาศัยระบบนี้ทั้งสิ้น เพราะโปรโตคอล IP นี้มีกลไกที่ค่อนข้างฉลาดในการหาเส้นทางขนส่ง โดยการหาเส้นทางที่สั้นที่สุด และดีที่สุด

ถึงแม้ว่า IP จะเป็นโปโตคอลที่เชี่ยวชาญในการรับส่งข้อมูลไปได้ไกล ๆ แต่ก็มีจุดด้อยคือ Unreliable และ connectionless (เปรียบเสมือนเป็นระบบขนส่งที่ชำนาญรวดเร็วแต่ไม่รับประกันว่าข้อมูลจะไปถึงปลายทางหรือไม่) การที่ IP มีข้อด้อยทั้ง 2 ประการนี้ ดังนั้นโปรโตคอลเลเยอร์อื่นที่ใช้ IP เป็นตัวส่งข้อมูลจำเป็นต้องหาหนทางในการลดข้อด้อยเหล่านี้ เพื่อให้การรับส่งข้อมูลมีเสถียรภาพและเชื่อถือได้

2.2.2.9 IP Header

4-bit version	4-bit header length	8-bit type of service	16 bit total length (in bytes)
16 bit indentification		3 bit flags	13 bit fragment offset
8 bit time to live (TTL)	8 bit protocol	16 bit header checksum	
32 bit source IP address			
32 bit destination IP address			
Option (if any)			
data			

รูปที่ 2.6 แสดง IP Header [8]

จากภาพ IP Header ขนาดของ IP Header โดยปกติจะมีขนาด 20 ไบต์ ยกเว้นในกรณีที่มีการเพิ่มเติมออปชั่นบน IP Header จากภาพจะแสดงเป็นท่อนละ 32 บิต ก็คือ 4 ไบต์ โดยการส่ง

ข้อมูลจะเรียงลำดับให้ไบต์แรกก่อนแล้วค่อยตามด้วยไบต์ถัดไป สำหรับข้อมูลแต่ละส่วนในเฮดเดอร์มีความหมายดังนี้

บิต 0 – 3 version of TCP/IP	ปัจจุบันเป็นเวอร์ชั่น 4
บิต 4 – 7 Header Length	ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีออปชั่นค่าในส่วนนี้จะเป็น 5 หมายความว่าความยาวข้อมูลมีขนาด 5*32 บิตหรือเท่ากับ 20 ไบต์
บิต 8 – 15 Type of service	ปัจจุบันไม่ใช้งานแล้ว ถูกออกแบบมาเพื่อเก็บค่าของ Minimize delay, Maximize Throughput, Maximize reliability, Minimize Monetary cost เพื่อใช้เป็นตัวบ่งชี้ให้เราต์เตอร์ตัดสินใจในการเลือกเราต์ข้อมูลแต่ละดาต้าแกรม อย่างไรก็ตามในปัจจุบันไม่มีการนำส่วนนี้ไปใช้งานแต่อย่างไร
บิต 16 – 31 Total length	เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP Datagram ด้วยขนาด 16 บิตของฟิลด์นี้แสดงว่าขนาดความยาวข้อมูลของ IP ดาต้าแกรมจะมีขนาดสูงสุด 65535 ไบต์ ขึ้นอยู่กับขนาด MTU โดยทั่วไปแล้วถึงแม้ว่าจะสามารถส่งข้อมูลใน 1 ดาต้าแกรมได้สูงสุด 65535 ไบต์ แต่เมื่อถูกส่งลงไปเลเยอร์ล่างก็จะถูกแฟรมเมนต์ก่อนทำการส่งจริงอยู่ดี และแอพพลิเคชั่นเลยอร์ส่วนใหญ่ก็จะรับส่งข้อมูลครั้งละ 512 ไบต์ใน 1 ดาต้าแกรม ฟิลด์นี้เป็นฟิลด์ที่จำเป็นที่ต้องระบุไว้ในทุกดาต้าแกรมเพื่อทำให้สามารถถอดข้อมูลออกมาได้
บิต 32 – 47 Identification	เป็นหมายเลขของดาต้าแกรมที่ส่งในกรณีที่มีการกระจายของดาต้าแกรมเดียวกัน
บิต 48 – 50 แฟลก	ใช้ในกรณีที่มีการแฟรกเมนต์ของดาต้าแกรม
บิต 51 – 63 Fragment offset	ใช้ในการกำหนดตำแหน่งของข้อมูลใน 1 ดาต้าแกรมที่ถูกแฟรกเมนต์กลับมาต่อเรียงกันในตำแหน่งของข้อมูลที่ถูกต้อง

บิต 64 – 71 Time to live	เป็นฟิลด์ที่กำหนดจำนวนครั้งสูงสุดที่ดาต้าแกรมนี้จะถูกเราต์ เพื่อป้องกันมิให้ดาต้าแกรมถูกเราต์ไปโดยไม่มีที่สิ้นสุด ด้วยคุณสมบัติของ IP ที่ถูกเราต์ไปเรื่อย ๆ จนกว่าจะถึงปลายทาง ซึ่งในบางครั้งอาจจะยังหาเส้นทางไม่ได้ TTL จะเป็นตัวบอกจำนวนสูงสุดของการเราต์ หากว่าจำนวนครั้งที่ดาต้าแกรมถูกเราต์ไปเท่ากับ TTL แล้วยังไม่ถึงปลายทางก็ให้ทำการดร็อปดาต้าแกรมนี้ทิ้ง แล้วแจ้งกลับมายังต้นทางว่า Time out คือหมดเวลาก่อน ทั้งนี้ในการเราต์ผ่านเราต์เตอร์ 1 ครั้ง ค่า TTL จะลดลง 1 หาก เมื่อค่า TTL ลดลงจนถึง 0 เมื่อใดแสดงว่าดาต้าแกรมนั้นไทม์เอาท์ไปแล้วและจะไม่ถูกเราต์อีกต่อไป
บิต 72 – 79 Protocol	อย่างที่ได้กล่าวไปแล้วข้างต้นว่ามีโปรโตคอลย่อยหลายตัวที่อาศัย IP ในการขนส่งข้อมูล ดังนั้นเพื่อระบุว่าข้อมูลที่ IP กำลังส่งอยู่นี้เป็นของโปรโตคอลอะไรก็ต้องระบุไว้ในฟิลด์นี้
บิต 80 – 95 Header checker	เป็นส่วนตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์เพื่อป้องกันการผิดพลาดในการส่งข้อมูล ซึ่งจะป้องกันเฉพาะข้อมูลในเฮดเดอร์เท่านั้น มิใช่ข้อมูลทั้งหมดในดาต้าแกรม
บิต 95 – 127	คือ IP Address ของผู้ส่งข้อมูลดาต้าแกรม Source IP Address
บิต 128 -163	คือ IP Address ของปลายทางผู้รับ Destination IP Address

2.2.2.10 ข้อมูลดาต้าแกรม

IP Header เป็นส่วนสำคัญที่สุดของ IP ดาต้าแกรม ก่อนที่จะทำการศึกษาถึงช่องว่างและจุดบกพร่องของ IP จำเป็นต้องเข้าใจการทำงานปกติของ IP เสียก่อน แล้วจากนั้นเมื่อเห็นการโจมตีที่ระดับ IP ก็จะสามารถป้องกันและเข้าใจการโจมตีนั้นได้ไม่ยากนัก ค่าต่าง ๆ ในแต่ละฟิลด์ของ Header ล้วนเป็นกลไกที่ควบคุมการเดินทางของดาต้าแกรม และถ้านำมาใช้ผิดวิธีก็อาจจะกลายเป็นเครื่องมือในการบุกรุกได้ กรณีที่ค่าในเฮดเดอร์ผิดปกตินั้นมีความเป็นไปได้อยู่ 2 กรณี

- Encapsulate ทำงานผิดพลาด เนื่องจาก Encapsulator ก็คือโปรแกรมฝังตัวอยู่กับอุปกรณ์สื่อสาร สำหรับกรณีที่เป็นอุปกรณ์ TCP/IP ทั่วไปหรืออาจจะเป็น TCP Stack ในระบบปฏิบัติการในกรณีที่เป็นเครื่องคอมพิวเตอร์ ซึ่งโปรแกรมเหล่านี้มีโอกาสที่จะถูกเขียนมาอย่างไม่รอบคอบ และในบางกรณีอาจจะคำนวณค่าเฮดเดอร์ผิดพลาดได้

- การจงใจใส่ค่าที่ผิดพลาดลงใสเฮดเดอร์เพื่อมุ่งร้าย ในกรณีนี้แฮกเกอร์จะไม่ได้ส่งข้อมูลผ่านกระบวนการ Encapsulate ตามปกติ แต่จะเขียนโปรแกรมเพื่อสร้าง IP แพ็กเก็ตขึ้นมาเองโดยที่มีข้อมูลในเฮดเดอร์เป็นค่าผิดแปลกจากปกติเพื่อหวังผลให้กระบวนการ Demultiplexing ของผู้รับทำงานผิดปกติไปจากเดิม

2.2.2.11 IP Routing

IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังที่หมายปลายทางโดยผ่าน IP เพื่อส่งข้อมูลไปยังทุก ๆ ที่ในโลกบนอินเตอร์เน็ตที่ดีที่สุดในขณะนี้คือการที่ IP มีกระบวนการ IP Routing นี้เอง การสื่อสารข้อมูลแต่ละครั้ง ข้อมูลจะต้องเดินทางผ่านโครงข่ายอันสลับซับซ้อนมากมาย แต่ในที่สุดข้อมูลก็สามารถสงถึงกันได้ในเวลาอันรวดเร็ว

2.3 ARP : Address Resolution Protocol[8]

ARP (Address Resolution Protocol) เป็นกระบวนการเปลี่ยนค่าระหว่าง IP ไปเป็น Ethernet Address (หรือเป็นที่รู้จักกันในชื่อ MAC Address คือแอดเดรสทางฮาร์ดแวร์ของอุปกรณ์เช่นการ์ด LAN) IP Address นั้นเป็นเพียงลอจิคัลแอดเดรสที่ผู้ใช้กำหนดขึ้นมาให้สำหรับโฮสต์ตัวใดตัวหนึ่ง อาจจะเปลี่ยนค่าเป็นแอดเดรสอื่นเมื่อไรหรืออย่างไรก็ได้ตามที่ผู้ใช้ต้องการ การตั้งค่า IP Address ผิดก็เพียงแต่ทำให้ไม่สามารถสื่อสารได้ แต่เมื่อใดผู้ใช้ตั้งค่า IP ได้ถูกต้อง โฮสต์นั้นก็จะกลับมาสื่อสารได้อีกครั้ง การสื่อสารข้อมูลได้หรือไม่ในระดับ IP จึงเป็นเพียงสภาวะทางลอจิคัลที่ไม่ตายตัว ต่างกับในระดับอีเธอร์เน็ต ซึ่งอุปการณ์ทุกชนิดจะมีหมายเลยประจำตัว (Ethernet Address) ซึ่งเป็นเลขที่ระบุติดตัวอยู่กับอุปกรณ์นั้น ๆ เป็นเลขขนาด 48 บิต(6ไบต์) จะสามารถมีค่าที่แตกต่างกันได้ = 2^{48} = 281,474,976,710,656 ค่า และอุปกรณ์ Ethernet ทุกชิ้นที่ผลิตขึ้นมาในโลกใบนี้จะไม่มี Ethernet Address ที่ซ้ำกันเลย

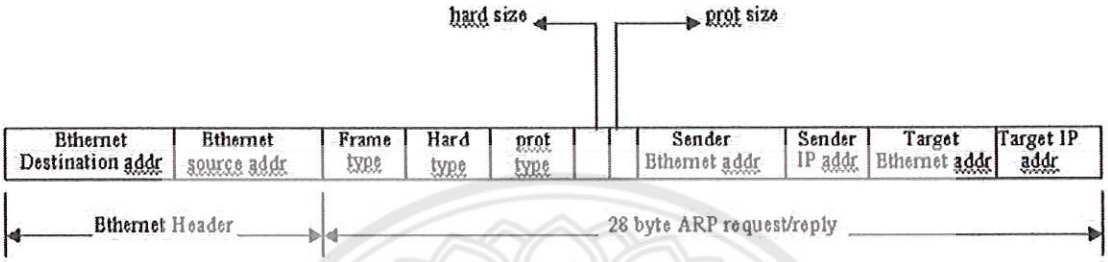
การที่จะศึกษาไปถึงกระบวนการในการเปลี่ยน IP Address กับอีเธอร์เน็ตแอดเดรสนั้น จะขอให้ดูกระบวนการ Encapsulation จาก IP เป็น Ethernet จะทำให้สามารถเข้าใจได้ง่าย และเร็วขึ้น

2.3.1 ARP Cache

กระบวนการ ARP จะเกิดขึ้นเสมอก่อนที่ดาต้าแกรมของ IP จะถูก Encapsulate ของบน Ethernet และกระบวนการ ARP Request – Reply ก็จะให้เวลาระยะหนึ่งและใช้แบนด์วิดธ์ของเน็ตเวิร์กบางส่วนไปด้วย ทำให้ประสิทธิภาพในการรับส่งลดลง จึงมีการกำหนดอายุของข้อมูล

ใน ARP Cache กล่าวคือข้อมูลใน ARP Cache จะยังคงเชื่อถือได้ในเวลาหนึ่งเท่านั้น หากล่วงเลยเวลาที่กำหนดไว้อาจทำให้การจับคู่ระหว่าง MAC – IP อาจจะเปลี่ยนไปแล้วก็ได้ ซึ่งจะต้องมีการ ARP Request ใหม่เพื่อให้ได้ข้อมูลปัจจุบันและทันสมัยที่สุด โดยทั่วไปอายุของข้อมูลใน ARP Cache จะเก็บไว้ 20 นาที หากเลยจากนั้นค่าใน Cache ก็จะถูกลบทิ้ง นั่นหมายถึงหากต้องการติดต่อก็จะต้องทำ ARP Request ใหม่นั่นเอง

2.3.2 ARP Packet Format



รูปที่ 2.7 แสดง Arp Encapsulation [8]

ในแพ็กเก็ตของ ARP จะประกอบด้วย

ไบต์ 0 – 5 Ethernet Destination Address สำหรับ Ethernet ทั่วไปจะหมายถึงแอดเดรสของปลายทาง แต่สำหรับกรณีของ ARP เนื่องจากเป็นการส่งข้อมูลถึงทุกโฮสต์ที่อยู่บนเน็ตเวิร์ก หรือที่เรียกว่าบรอดคาสต์ ดังนั้นทุกบิตของฟิลด์นี้จึงต้องเป็น "1" ทั้งหมดคือ FF FF FF FF FF FF

ไบต์ 6 – 11 Ethernet Source Address เป็นแอดเดรสของผู้ที่ส่ง ARP Request เอง เพื่อให้โฮสต์ที่ต้องการตอบกลับสามารถตอบกลับมาได้อย่างถูกต้อง

ไบต์ 12 – 13 Ethernet Frame Type ระบุถึงโปรโตคอลที่ Encapsulate อยู่ใน Ethernet Frame นี้ สำหรับ ARP จะต้องเป็น 0X0806

ไบต์ 14 – 15 Hard Type ระบุประเภทของ Hardware Address ที่ ARP กำลังถามอยู่ ในกรณีนี้คือ Ethernet Address ค่าจะต้องเป็น 1

ไบต์ 16 – 17 Prot Type ระบุโปรโตคอลที่ต้องการถาม หมายถึงต้องการถาม Hardware Address ของโปรโตคอลอะไร กรณีนี้ก็คือ IP

ไบต์ 18 Hard Size ระบุขนาดของฮาร์ดแวร์แอดเดรส = 6 สำหรับ Ethernet Address

ไบต์ 19 Port Size ระบุขนาดของแอดเดรสในโปรโตคอลที่ถาม = 4 สำหรับ IP

ไบต์ 20 – 21 OP Field เป็นการระบุว่าเป็น ARP ชนิดใด

1 = ARP Request

2 = ARP Reply

3 = RARP Request

4 = RARP Reply

ไบต์ 22 – 27 Sender Ethernet Address ของผู้ส่งซึ่งจะมีค่าซ้ำกับใน ไบต์ที่ 6 – 11

ไบต์ 28 – 31 Sender IP Address คือค่า IP Address ของผู้ส่ง

ไบต์ 32 – 37 Target Ethernet Address จะว่างไว้สำหรับ ARP Request เพราะยังไม่รู้ (ถ้ารู้แล้วคงไม่ต้องทำ ARP Request อีก)

ไบต์ 38 – 41 Target IP Address คือค่า IP Address ที่กำลังต้องการหาค่า Ethernet Address

2.3.3 ARP Reply

หลังจากมีการส่งกระจาย ARP Request ลงไปยังเน็ตเวิร์กแล้วโฮสต์ทุกตัวที่อยู่บนเน็ตเวิร์กจะได้รับ ARP Request นี้ ซึ่งก็จะนำค่าในไบต์ที่ 3 - 41 มาเปรียบเทียบกับ IP Address ของตอนเอง หากไม่ตรงกันก็ไม่ต้องทำอะไร ถ้าตรงกันก็จะตอบกลับด้วย ARP Reply โดยใช้แพ็กเก็ตเดิมของ ARP ที่อธิบายตอนต้น แต่จะเปลี่ยนค่าของข้อมูลดังนี้

- เปลี่ยนค่าใน OP field จาก 1 เป็น 2 แสดงว่าเป็น ARP Reply
- นำค่าใน Source Address ทั้ง Hardware และ IP ไปใส่ลงใน Target Address ทั้ง Hardware และ IP (เพื่อเป็นการส่งกลับไปยังผู้ส่ง ARP Request มา)
- นำค่า Hardware Address และ IP แอดเดรสของตนเองใส่ลงในฟิลด์ Ethernet Source Address และ Sender IP Address

เมื่อมีการ ARP Request และมีผู้ส่ง ARP Reply นั่นหมายถึงกระบวนการหา Ethernet Address ของ IP Address ได้เสร็จสมบูรณ์ ต่อไปก็จะเริ่มส่ง IP Datagram ได้

2.4 ICMP : Internet Control Message Protocol[8]

ICMP เป็นโปรโตคอลหนึ่งที่อยู่ในชุดของ TCP/IP Suite มีหน้าที่ส่งข่าวสารและคำสั่งควบคุมของ IP โดยเฉพาะการรับส่ง Error Message ด้วยลักษณะของ ICMP แล้วนับได้ว่าอยู่ทั้งในเลเยอร์เดียวกับ IP หรือเลเยอร์ที่สูงกว่า IP คือเทียบเท่า TCP และ UDP ก็ได้ ขึ้นอยู่กับลักษณะของ Message ที่ ICMP ทำการสื่อสาร

2.4.1 ICMP Encapsulation

8-bit Type	8-bit code	16-bit checksum
ข้อมูลภายในขึ้นกับ Type และ Code		

รูปที่ 2.8 แสดง ICMP Encapsulation [8]

บิตที่ 0 – 7 (ไบต์ที่ 0) ICMP Type เป็นฟิลด์ขนาด 8 บิตบอกถึงประเภท ICMP ที่กำลังสื่อสารอยู่

บิตที่ 8 – 15 (ไบต์ที่ 1) ICMP Code เป็นฟิลด์ขนาด 8 บิตที่เก็บข้อมูลรหัสของ ICMP Message

บิตที่ 16 – 31 (ไบต์ที่ 2 – 3) Check sum ใช้เป็นตัวตรวจสอบความถูกต้องของ ICMP Message ทั้งหมด

โดยทั่วไปความยาวของ ICMP Message จะไม่คงที่แน่นอน ขึ้นอยู่กับลักษณะของ ICMP Message นั้น ๆ ว่าจะมีข้อมูลตามหลังมามากน้อยแค่ไหน

2.4.2 การใช้งานของ ICMP

โดยทั่วไป ICMP จะใช้งานในสองลักษณะคือ

- **Query** ใช้สำหรับสอบถามสถานะระหว่างกัน
- **Error** ใช้สำหรับรายงานข้อผิดพลาดที่เกิดขึ้น

- เกิดข้อผิดพลาดในการส่ง ICMP Error Message เสียเอง ถึงแม้ว่า ICMP เป็นตัวที่คอยรายงานความผิดพลาดของ IP แต่ตัว ICMP เองก็ยังต้องอาศัย IP เป็นตัวนำมันกลับไปยังปลายทางอยู่ดี การที่ ICMP อาจจะเดินทางกลับไม่ถึงปลายทางด้วยเหตุใดก็แล้วแต่ย่อมเป็นสิ่งที่มีโอกาสเกิดขึ้นได้ เช่นการที่ Host A ต้องการส่งข้อมูลไปยัง Host B แต่ Host B ไม่ได้เปิดใช้งานอยู่ ดังนั้น Router ตัวที่ต่ออยู่กับ Host B จะต้องส่ง ICMP Type 3 Code1 (Host Unreachable) กลับไปยัง Host A แต่ Router ระหว่างทางนั้นมีเราต์ติ้งเทเบิลที่ไม่ถูกต้องทำให้ ICMP แพ็กเก็ตไม่สามารถส่งต่อกลับไปยัง Host A ได้ด้วยเช่นกัน ในกรณีนี้ หากไม่มีข้อกำหนดไว้ก็จะทำให้เกิด ICMP ย้อนกับไปมาไม่รู้จบ

- ข้อมูลที่ IP Address ปลายทางไม่ได้เฉพาะเจาะจงโฮสต์ตัวเดียว แต่เป็นปลายทางประเภทบรอดคาสต์ (ทุก ๆ โฮสต์ในเน็ตเวิร์ก) และมัลติคาสต์ (โฮสต์หลายตัวพร้อมกัน)

- ดาต้าแกรมที่ทำหน้าที่เหมือนการบรอดคาสต์ของ Link Layer

- ดาต้าแกรมที่แพ็กเก็ตถูกแฟรกเมนต์มานั้น แยกมาเป็นหลายแพ็กเก็ต ยกเว้นดาต้าแกรมแรกเท่านั้น เนื่องจากดาต้าแกรมที่ถูกแฟรกเมนต์มานั้น จริง ๆ แล้วมีที่มาจากดาต้าแกรมขนาดใหญ่เพียงอันเดียว ดังนั้นหากมีปัญหาในการสื่อสาร การแจ้งกลับไปก็ควรจะแจ้งเพียงครั้งเดียว ไม่ควรแจ้งตอบกลับทุกแฟรกเมนต์

- ดาต้าแกรมที่ต้นทางไม่ได้เฉพาะเจาะจงโฮสต์ ดังนั้นแอดเดรสในลักษณะนี้เมื่อนำมาใช้เป็นแอดเดรสต้นทางก็จะไม่ได้รับ ICMP Message เช่น แอดเดรสต้นทางเป็นศูนย์ทั้งหมด แอดเดรสที่เป็น loop back (ส่งเข้าหาตัวเอง) แอดเดรสที่เป็นบรอดคาสต์หรือมัลติคาสต์แอดเดรส

เหล่านี้จะเป็นแอดเดรสที่ไม่ระบุถึงโฮสต์ใดโฮสต์หนึ่งจริง ๆ ดังนั้น ICMP จึงไม่ทำการส่ง Message ไปให้

2.5 UDP : User Datagram Protocol[8]

UDP เป็นโปรโตคอลพื้นฐานที่อาศัย IP เป็นพาหนะในการส่งข้อมูลโดยตัว UDP นั้นจัดอยู่ใน Transport Layer ลักษณะของโปรโตคอลจะจัดการครั้งละ 1 ชุดของข้อมูลที่เรียกว่า UDP datagram โดยข้อมูลแต่ละดาต้าแกรมจะไม่มีความสัมพันธ์ เพราะความสัมพันธ์ระหว่างดาต้าแกรมจะถูกจัดการโดยโปรโตคอลอื่นในแอพลิเคชั่นเลเยอร์แทน

UDP ดาต้าแกรมจะถูก Encapsulate ลงใน IP ดาต้าแกรม โดยเมื่อ Encapsulate แล้ว 20 ไบต์แรกจะเป็นของ IP Header และในไบต์ที่ 9 ของ IP Header จะต้องมีค่า = 17 ด้วย คุณสมบัติสำคัญของ UDP คือจัดรูปแบบข้อมูลอย่างง่ายให้อยู่ในรูปของ UDP Datagram และส่ง – รับข้อมูลชุดนี้ให้ถึงปลายทางเท่านั้น ไม่มีกลไกใด ๆ ในการตรวจสอบยืนยันการรับส่งในตัวของ UDP เอง ดังนั้นแอพพลิเคชั่นที่ใช้ UDP จะต้องระลึกเสมอว่า UDP เป็นโปรโตคอลที่ไม่เสถียรภาพและไม่รับประกันการส่ง – รับข้อมูล (Unreliable) และจะต้องเสริมกลไกในส่วนที่รับประกันการรับส่งข้อมูลในแอพพลิเคชั่นนั้นเองเสมอ โดยทั่วไปแอพพลิเคชั่นที่มีความจำเป็นในเรื่องเสถียรภาพ และความถูกต้องของข้อมูลนั้นควรจะหลีกเลี่ยงการใช้ UDP เพราะแอพพลิเคชั่นจะต้องเสียเวลาในการจัดการเรื่องนี้มากจึงจะทำงานได้ โดยส่วนใหญ่ควรจะใช้ TCP แทนซึ่งจะลดปัญหานี้ได้

2.5.1 UDP Header

16-bit source port number	16-bit destination port number
16-bit UDP length	16-bit UDP checksum
Data	

รูปที่ 2.9 แสดง UDP Encapsulation [8]

จากภาพจะเห็นว่า UDP Header ได้มีการกำหนดฟิลด์ไว้อย่างง่าย ๆ มีขนาดทั้งหมดของเฮดเดอร์เพียง 8 ไบต์เท่านั้น โดยแต่ละฟิลด์มีความหมายดังนี้

ไบต์ 0 – 1 Source Port Number หมายเลขพอร์ตของต้นทางที่ส่งข้อมูลดาต้าแกรมนี้

ไบต์ 2 – 3 Destination Port Number หมายเลขพอร์ตของปลายทางที่จะเป็นผู้รับข้อมูลดาต้าแกรมไปใช้งาน

ไบต์ 4 – 5 UDP Length เป็นฟิลด์ที่ระบุความยาวของ UDP ดาต้าแกรม คือ UDP Header + UDP data ขนาดต่ำสุดของ UDP Length = 8 หมายถึง ดาต้าแกรมนี้มีเฉพาะ UDP Header ซึ่งเท่ากับ 8 ไบต์ และไม่มี UDP Data เลย

ไบต์ 6 – 7 UDP Chechsum ทำหน้าที่ตรวจสอบความถูกต้องของ UDP ดาต้าแกรมทั้งหมด ถึงแม้ว่า UDP จะถูก Encapsulate อยู่ใน IP ดาต้าแกรมและมี IP Checksum คุมอยู่แล้วก็ตาม แต่ IP Chechsum นั้นจะทำหน้าที่เฉพาะตรวจสอบความถูกต้องของ IP Header เท่านั้น มิได้ครอบคลุมทั้ง IP ดาต้าแกรมแต่อย่างใด นั่นหมายถึงหากมีความผิดพลาดในส่วนของที่เป็นข้อมูลของ IP ก็จะไม่สามารถทราบได้ ซึ่งต่างจาก UDP Checksum ซึ่งทำหน้าที่ตรวจสอบความถูกต้องทั้ง UDP ดาต้าแกรมและเป็นกลไกของ UDP เองแยกต่างหากจาก IP ดังนั้นการตรวจสอบความถูกต้อง Checksum เมื่อใช้ UDP ก็จะมีขึ้นตอน คือการตรวจสอบความถูกต้องด้วย Checksum เมื่อใช้ UDP ก็จะมี 2 ขั้นตอนคือการตรวจสอบความถูกต้องในระดับ IP Header โดยใช้ Checksum ของ IP หลังจากนั้นจึงค่อยทำการตรวจสอบความถูกต้องของ UDP โดยใช้ UDP Checksum อีกทีหนึ่ง

2.5.2 UDP Checksum

<table>
<tr><td colspan="3">32 bit source IP address</td></tr>
<tr><td colspan="3">32 bit destination IP address</td></tr>
<tr><td>Zero</td><td>8-bit protocol</td><td>16-bit UDP length</td></tr>
<tr><td colspan="2">16-bit source port number</td><td>16-bit destination port number</td></tr>
<tr><td colspan="2">16-bit UDP length</td><td>16-bit UDP checksum</td></tr>
<tr><td colspan="3">data</td></tr>
<tr><td colspan="2">Pad byte</td><td></td></tr>
</table>

รูปที่ 2.10 แสดง UDP Checksum [8]

กลไกในการหาค่า checksum เพื่อใช้ตรวจสอบความถูกต้องของ UDP จะคล้ายกับการหาค่า checksum ของ IP คือค่า checksum ที่ได้จะเป็นค่าผลรวมของข้อมูลขนาด 16 บิตทั้งหมด และแปลงเป็น one's compliment แต่จะมีจุดแตกต่างจากการหาค่า checksum ของ IP อยู่ 2 ประการ คือ

- ขนาดของ UDP ดาต้าแกรมจะไม่คงที่เนื่องจากขนาดของส่วนที่เป็น Data อาจเปลี่ยนแปลงได้ตามขนาดของข้อมูลจริง

- ถึงแม้ว่าจริง ๆ แล้ว UDP Header จะมีขนาด 8 ไบต์ แต่ในการหาค่า checksum นั้นจะนำบางค่าใน IP Header มารวมเป็นส่วนหนึ่งของ UDP Header (เรียกว่า UDP Pseudo Header) จากนั้นจึงหาค่า checksum ทั้งหมดอีกทีหนึ่ง

ฟิลด์ที่ UDP นำมาจาก IP Address ดังแสดงในภาพข้างบน ได้แก่ Source IP Address Destination IP Address, Zero, Protocol, UDP length ทั้งนี้เพื่อให้ค่า UDP Checksum ได้การตรวจสอบซ้ำในส่วนที่สำคัญสำหรับ UDP ด้วยว่าทำการรับ – ส่งถูกต้องทั้งต้นทางและปลายทาง

ในบางระบบที่นำ UDP ไปใช้งานอาจกำหนดให้ checksum เป็น option คือเลือกที่จะมีหรือไม่มี checksum ได้โดยหวังผลในแง่ประสิทธิภาพในการรับ – ส่งข้อมูลจะสูงขึ้นเพราะไม่ต้องเสียเวลาในการตรวจสอบ checksum และปล่อยหน้าที่ในการป้องกันความผิดพลาดของข้อมูลให้เป็นของ Datalink Layer ซึ่งก็ได้ผลในแง่ของการช่วยเพิ่มประสิทธิภาพ แต่ใน Datalink Layer เอง ก็จะมีข้อผิดพลาดเกิดขึ้นพอสมควร จะทำให้ระบบโดยรวมไม่มีเสถียรภาพมาก ดังนั้นการใช้งาน UDP ควรจะต้อง Enable checksum เสมอ

2.5.3 ขนาดของ UDP Datagram

ในทางทฤษฎีแล้ว UDP Datagram จะมีขนาดได้สูงสุดเท่ากับ 65535 ไบต์ โดยแบ่งเป็น IP Header เสีย 20 ไบต์ และเป็น UDP Header อีก 8 ไบต์ คงเหลือส่วนที่เป็นข้อมูลเท่ากับ 65507 ไบต์อย่างไรก็ตามการนำ UDP ไปใช้งานส่วนใหญ่จะใช้งานของดาต้าแกรมที่ต่ำกว่านี้ โดยทั่วไปขีดจำกัดของ UDP Datagram จะมีอยู่ 2 ประการคือ

- Application: ในการรับส่งข้อมูลแต่ละครั้ง แอพพลิเคชั่นจะต้องทำการจองพื้นที่หน่วยความจำเพื่อทำการรับ-ส่งข้อมูลให้เหมาะสมกับสภาพของโฮสต์และคุณสมบัติของตัวแอพพลิเคชั่นเองด้วย เช่น หากกำหนดขนาดใหญ่เกินไปก็จะทำให้เปลืองหน่วยความจำ หรือการรับส่งข้อมูลแต่ละครั้งของแอพพลิเคชั่นนั้น ๆ อาจจะต้องการข้อมูลจำนวนหนึ่งเท่านั้น มิได้ต้องการเต็มทั้ง 64k ดังนั้นโดยทั่วไปขนาดของ UDP Datagram จะถูกตั้งค่าไว้เท่ากับ 8192 ไบต์
- ขนาดของ IP Datagram: โดยส่วนใหญ่การนำ TCP/IP ไปใช้งานกับระบบปฏิบัติการต่าง ๆ จะมีการกำหนดขนาดของ IP ดาต้าแกรมให้ต่ำกว่า 64k

จึงทำให้ UDP ดาต้าแกรม ซึ่งต้องอาศัยอยู่ใน IP ดาต้าแกรมอีกชั้นหนึ่งถูกจำกัดขนาดไปด้วยโดยปริยาย

2.6 TCP : Transmission Control Protocol[8]

TCP เป็นโปรโตคอลที่แตกต่างจากโปรโตคอลอื่น โดยจะเห็นได้ว่าโปรโตคอลในระดับ IP หรือแม้กระทั่ง UDP จะสนใจข้อมูลเพียง 1 ดาต้าแกรม กลไกของโปรโตคอลจะมีหน้าที่ตรวจสอบความถูกต้องเพียงเฉพาะดาต้าแกรมนั้นๆ อย่างเดียว เมื่อจะทำการส่งดาต้าแกรมใหม่ก็จะถือว่าเป็นข้อมูลชุดใหม่ที่ไม่มีความสัมพันธ์ใดๆ กับดาต้าแกรมอื่น

แต่สำหรับ TCP แล้วจะเห็นได้ว่าข้อมูลนั้นเป็น stream คือมีความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านส่งและด้านรับ เพื่อให้แน่ว่าทั้ง 2 ฝั่งมีความพร้อมและสามารถสื่อสารกันได้จริง จึงจะมีการส่งข้อมูลเกิดขึ้น จนมีการรับส่งข้อมูลแล้วถูกต้องตรงกันทั้ง 2 ฝ่าย ด้วยลักษณะเช่นนี้ การสื่อสารด้วย TCP จึงเสมือนว่าทั้ง 2 ฝ่าย คือฝ่ายรับและฝ่ายส่ง โดยทั้งสองฝ่ายได้ทำการต่อสายเน็ตเวร์ก ถึงกัน (connected) ตลอดเวลาที่ทำการรับส่งข้อมูลจนกระทั่งการสื่อสารทั้งหมดเสร็จสิ้น จึงจะทำการยกเลิกการเชื่อมต่อนั้นเสีย

2.6.1 TCP Services

คุณสมบัติของ TCP มีข้อดีคือ

- ข้อมูลที่ส่งผ่าน TCP จะถูกนำมาแตกแยกออกเป็นส่วนๆ เพื่อให้เหมาะสมกับการส่ง โดย TCP เป็นตัวพิจารณาว่าขนาดเท่าใดจึงจะทำให้การรับ - ส่งนั้นมีประสิทธิภาพและน่าเชื่อถือสูงสุด

- ในการส่งข้อมูลแต่ละครั้งของ TCP จะมีการจับเวลาไว้เสมอเพื่อรอให้อีกฝั่งหนึ่งตอบยืนยันการรับข้อมูลกลับมา หากถึงเวลากำหนดที่ข้อมูลจะถึงปลายทางแต่ยังไม่มีการตอบกลับ TCP จะถือว่าข้อมูลไม่ถึงปลายทาง จึงทำการส่งข้อมูลซ้ำ หรือยกเลิกการติดต่อ ฯลฯ ด้วยเหตุนี้ทำให้การรับ-ส่งข้อมูลทุกครั้งแอพพลิเคชั่นสามารถทราบได้ว่าข้อมูลได้ถึงปลายทางหรือไม่

- ทุกๆครั้งที่ TCP ได้รับข้อมูลจากอีกฝั่งหนึ่งจะมีการตอบยืนยันไปยังผู้ส่งเสมอ

- TCP มี checksum ซึ่งจะครอบคลุมทั้ง TCP Header และ TCP Data เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูลนั้นถูกต้อง และไม่ได้ถูกแก้ไขระหว่างทาง หาก TCP ทำการตรวจสอบกับ checksum แล้วมีข้อผิดพลาดเกิดขึ้น TCP จะทำการทิ้งข้อมูลที่ได้รับและหรือไม่ตอบกลับไปยังผู้ส่ง

- เนื่องจาก TCP อาศัย IP ในการส่งข้อมูล ซึ่งอาจถูกแฟรกเมนต์ได้ และทำให้ข้อมูลนั้นถึงปลายทางได้ไม่ถูกต้อง หน้าที่ของ TCP เมื่อรับข้อมูลที่ถูกแฟรกเมนต์มาประกอบรวมกันให้ถูกต้องสมบูรณ์ก่อนส่งไปยัง Application Layer ต่อไป

- การรับ-ส่งข้อมูล อาจจะเกิดกรณี IP Datagram นั้นถูกส่งซ้ำขึ้นได้ TCP ที่รับข้อมูลซ้ำดังกล่าวจะต้องทราบว่าเป็น IP Datagram ที่ซ้ำกัน และไม่นำข้อมูลไปใช้งาน

- TCP มีกลไกควบคุมการไหลของข้อมูล (flow control) โดยอาศัยการรับส่งที่ถูกต้องและสัมพันธ์กันทั้ง 2 ฝ่าย ในขณะเดียวกันข้อมูลที่ส่งนั้นต้องอาศัย IP หลายดาต้าแกรมจึงจะได้ข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลต้องเตรียมบัฟเฟอร์ไว้จำนวนหนึ่ง เพื่อเป็นตัวรับข้อมูลและทำการเรียงข้อมูลใหม่ทั้งชุดในบัฟเฟอร์ ตรวจความถูกต้องและส่งไปยังแอพพลิเคชั่น ดังนั้นในการส่งข้อมูลในแต่ละครั้งจะขึ้นอยู่กับบัฟเฟอร์ที่เพียงพอของฝ่ายรับข้อมูลเท่านั้น

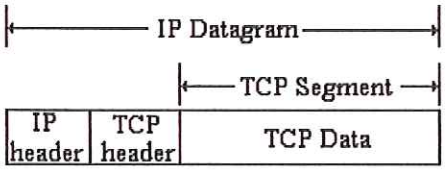
2.6.2 TCP Header

ใน TCP header จะทำการหมายเลขพอร์ตต้นทางและหมายเลขพอร์ตปลายทาง แต่ที่แท้จริงแล้วข้อมูลอีกส่วนหนึ่งที่ใช้ในการสื่อสาร คือ IP Address ของต้นทางและปลายทางก็ต้องระบุไว้ด้วยเช่นกัน และค่าของ IP Address ทั้งคู่จะอยู่ใน IP Header ส่วนคู่ของ IP Address และหมายเลขพอร์ตนั้นเรียกว่า ซ็อคเก็ต (socket) ซึ่งในการสื่อสารแต่ละครั้งต้องมีซ็อคเก็ตของต้นทางและปลายทาง ข้อมูลจึงจะรับ-ส่งได้ รายละเอียดและหน้าที่ของแต่ละฟิลด์ใน TCP header มีดังนี้

- Source port number : หมายถึงพอร์ตที่โฮสต์ต้นทางใช้ในการสื่อสารกันของเซสชั่นและ TCP/IP จะใช้พอร์ตนี้ไปตลอดโดยทั่วไปเรียกกันว่า ไคลเอนต์พอร์ต คือพอร์ตที่ไคลเอนต์เปิดขึ้นมาเพื่อรอการตอบรับจากเซิร์ฟเวอร์ ไคลเอนต์พอร์ตจะมีหมายเลขไม่แน่นอนและเปลี่ยนแปลงทุกครั้งที่มีการเชื่อมต่อใหม่ พอร์ตนี้จะเปิดในระยะเวลาสั้นๆ ค่าที่เป็นไปได้ของพอร์ตนี้ขึ้นอยู่กับการจัดสรรของระบบปฏิบัติการ ในการกำหนดขอบเขตของพอร์ตเหล่านี้ส่วนใหญ่มีค่าอยู่ในช่วง
1024 – 5000

- Destination Port Number : หมายถึงหมายเลขพอร์ตบนโฮสต์ปลายทางที่โฮสต์ต้นทางต้องการติดต่อด้วย พอร์ตนี้ถูกเรียกว่า เซิร์ฟเวอร์พอร์ต หมายเลขพอร์ตขึ้นอยู่กับแอพพลิเคชั่นที่ให้บริการ โดยทั่วไปแอพพลิเคชั่นแต่ละประเภทจะมีหมายเลขพอร์ตเป็นมาตรฐานให้ไคลเอนต์ได้เรียกใช้บริการ

- Sequence Number : เป็นฟิลด์ที่ระบุถึงหมายเลขที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้ง 2 ฝ่ายได้ทราบว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกันและมีลำดับที่ถูกต้องเนื่องจากการสื่อสารข้อมูลผ่าน IP ทำให้ข้อมูลถูกแบ่งออกเป็นส่วนๆไม่เรียงกัน หากไม่มีจุดอ้างอิงก็จะไม่สามารถอ่านข้อมูลที่ถูกต้องได้ ในการส่งและการรับข้อมูลใช้ฟิลด์นี้เป็นตัวยืนยันระหว่างกันเสมอ



รูปที่ 2.11 แสดง Encapsulation ของข้อมูล TCP ใน IP datagram [8]

16 – bit source port number								16 – bit destination port number
32 – bit sequence number								
32 – bit acknowledge number								
4-bit Header Length	Reserved (6 bits)	U R G	A C K	P S H	R S T	S Y N	F I N	16 – bit windows size
16 – bit TCP checksum								16 – bit urgent pointer
Option (ถ้ามี)								
Option (ถ้ามี)								

20 bytes

รูปที่ 2.12 แสดง TCP Header [8]

- Acknowledge Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่ต่างกันที่ Sequence Number ใช้ในการตอบรับโดยในการอ้างอิงนั้นผู้ส่งข้อมูลเป็นฝ่ายกำหนดตัวเลขขึ้นมาและส่งไปพร้อมกับการสร้างการเชื่อมต่อครั้งใหม่ แต่สำหรับฝ่ายที่ถูกติดต่อก็จำเป็นต้องกำหนดหมายเลขที่ใช้อ้างอิงในการตอบรับเช่นกัน ค่าที่อยู่ใน Acknowledge Number ก็คือหมายเลขที่ใช้อ้างอิงในการตอบรับนี้ ค่า Acknowledge Number และค่า Sequence Number ต้องพิจารณาประกอบกับ Flag จึงจะสามารถแปลความหมายของ TCP Segment ได้อย่างสมบูรณ์

- Header Length : โดยปกติความยาวของ TCP Header จะเท่ากับ 20 ไบต์ ถ้าหากมีการใช้ค่า Option อาจทำให้ค่าของเฮดเดอร์ยาวขึ้นตามข้อมูลที่ต้องเพิ่มจาก Option นั้น แต่ทั้งหมดแล้วจะไม่เกิน 60 ไบต์

- Flag : เป็นข้อมูลระดับบิตที่ใช้บอกคุณสมบัติของ TCP Segment ที่กำลังส่งอยู่นั้น และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลซึ่ง Flag ทั้งหมดมี 6 บิต ดังนี้

- URG : นำ Urgent Pointer ใช้งาน เพื่อทำการส่งข้อมูลอย่างรวดเร็วเมื่อผู้รับต้องการ

- ACK :เป็นตัวแสดงข้อมูลในฟิลด์ Acknowledge Number นั้นนำมาใช้งาน

- PUSH : เป็นตัวแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยังแอพพลิเคชั่นโดยเร็ว ทำงานโดยอัตโนมัติ

- RST : reset การติดต่อ

- SYN :ใช้ในการเริ่มต้นขอติดต่อปลายทาง

- FIN : ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

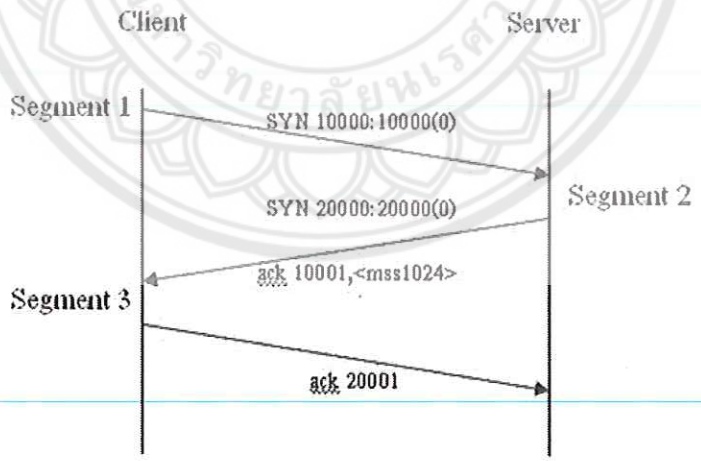
- Window size : เป็นขนาดของการรับส่งข้อมูลในแต่ละครั้ง ที่ทางฝ่ายผู้รับจะรับได้ เนื่องจากในการรับข้อมูลนั้นทางผู้รับต้องจัดเตรียมหน่วยความจำในการพักข้อมูล หากไม่มีการตกลงถึงขนาดที่ทางฝ่ายรับสามารถรับได้ ก็จะทำให้การส่งข้อมูลไม่สมดุล และฝ่ายรับอาจจะประมวลผลไม่ทัน ทำให้ต้องส่งข้อมูลซ้ำหลายครั้ง

- Checksum : เป็นฟิลด์ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน TCP Segment

- Urgent Pointer : ใช้ระบุหมายเลข Sequence Number ของ TCP Segment ล่าสุดที่อยู่ในโหมด Urgent

- Option : ข้อมูลเพิ่มเติมที่อยู่ใน TCP Header เมื่อมีการตั้งค่า option บางอย่างที่ต้องการข้อมูลเพิ่มเติมซึ่งไม่มีใน TCP header เช่น MSS,Strict Route

2.6.3 Connection Establishment



รูปที่ 2.13 แสดงการ connection Estabishment [8]

ก่อนที่ TCP จะทำการส่งข้อมูลนั้นต้องทำการ Establishment หรือการสร้างให้มีการ Connection เกิดขึ้นเสียก่อน เปรียบเสมือนการต่อสายของทั้งสองฝั่งเข้าด้วยกัน ซึ่งมีขั้นตอนการสร้างดังนี้

- เครื่องไคลเอนต์จะทำการส่งเซกเมนต์ โดยเปิด SYN Flag ระบุหมายเลขพอร์ตที่ต้องการติดต่อบนเซิร์ฟเวอร์และระบุหมายเลขลำดับของข้อมูล (ISN- Initial Sequence Number)

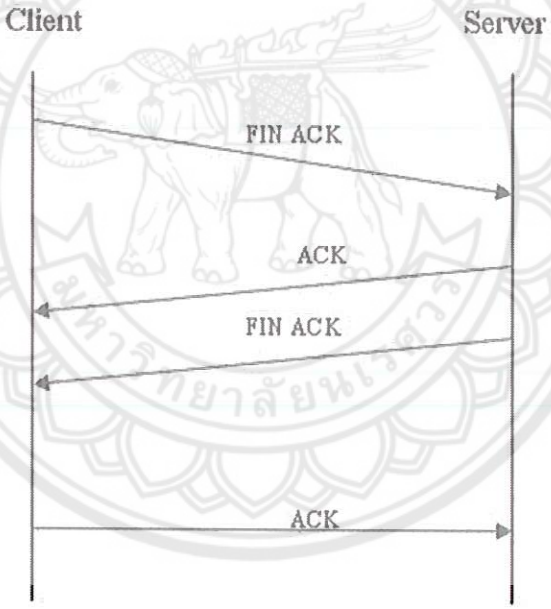
- เครื่องเซิร์ฟเวอร์เมื่อได้รับข้อมูลเซกเมนต์จากข้อ 1 ก็จะตอบกลับด้วยการเพิ่มค่า ISN ที่ได้รับขึ้นมาอีก 1 พร้อมทั้งระบุหมายเลขลำดับ ISN ของตนเองและเปิด SYN กับ ACK Flag

- ไคลเอนต์เมื่อได้รับการตอบกลับจากเครื่องเซิร์ฟเวอร์ตามข้อ 2 ก็จะทำการตอบรับกลับไป โดยการเพิ่มค่า ISN ของเซิร์ฟเวอร์ขึ้นอีก 1 และเปิด ACK Flag

เมื่อผ่านทั้ง 3 ขั้นตอนแล้ว ทั้งไคลเอนต์และเซิร์ฟเวอร์ก็เชื่อมต่อกันแล้ว ก็จะสามารถรับ- ส่งข้อมูลตลอดจนกว่าจะมีการยุติการเชื่อมต่อ ทั้ง 3 ขั้นตอนข้างต้นนั้นเรียกว่า Three-ways handshakes

2.6.4 Connection Termination

หลังการรับ-ส่งข้อมูลยุติลง จะต้องทำการหยุดการเชื่อมต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ออกไป โดยมีขั้นตอนอยู่ 4 ขั้นตอนคือ



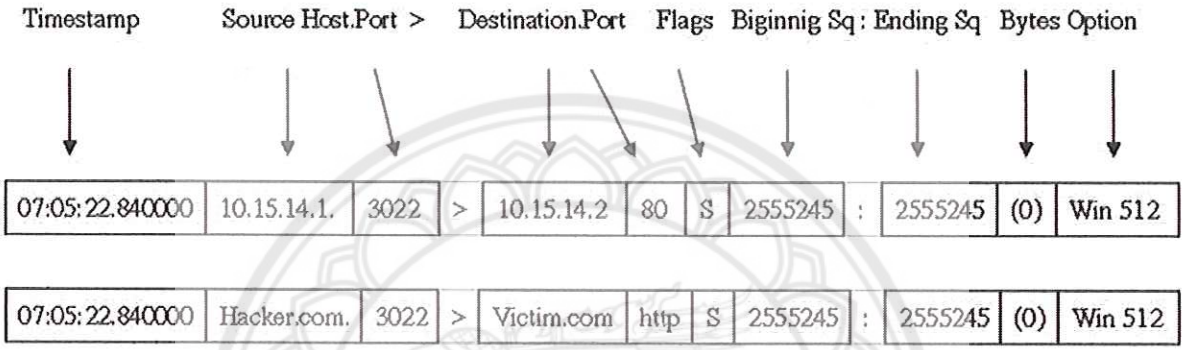
รูปที่ 2.14 แสดงการ connection [8]

- ไคลเอนต์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังเซิร์ฟเวอร์
- เซิร์ฟเวอร์ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อมกับ ACK Flag
- เซิร์ฟเวอร์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังไคลเอนต์
- ไคลเอนต์ตอบรับยุติการสื่อสารด้วย ISN+1 พร้อมกับ ACK Flag

2.7 วิธีการอ่านแพ็กเก็ต[8]

ในบทนี้จะกล่าวถึงการวิเคราะห์แพ็กเก็ตที่เกิดขึ้นเพื่อวัตถุประสงค์ต่างๆ โดยข้อมูลแพ็กเก็ตที่นำมาแสดงนั้นได้มาจากการใช้โปรแกรม TCPDUMP ซึ่งเป็นโปรแกรมที่เป็นสนิฟเฟอร์ชนิดหนึ่ง มีความสามารถดักอ่านข้อมูลมาเก็บไว้ และสามารถเรียกดูได้ตามเงื่อนไขต่าง ๆ ที่สนใจได้ ตัวอย่างของโปรโตคอล เพื่อเป็นพื้นฐานให้เข้าใจส่วนประกอบต่างๆ

2.7.1 โปรโตคอล TCP/IP



รูปที่ 2.15 แสดงการวิเคราะห์ของโปรโตคอล TCP/IP [8]

ข้อมูลที่ปรากฏประกอบด้วยฟิลด์ดังนี้

- Timestamp : แสดงเวลาที่ได้รับแพ็กเก็ตนี้ แสดงเป็นหน่วย ชั่วโมง : นาที : วินาที.เศษของวินาที

- Source Host : แสดง IP Address ต้นทางของแพ็กเก็ต สามารถแสดงเป็น 2 แบบ คือ IP Address โดยตรง เช่นในตัวอย่างคือ 10.15.14.1 หรือหากสามารถแปล IP เป็นชื่อโฮสต์ได้ก็จะแสดงเป็นชื่อโฮสต์ เช่นตัวอย่างบรรทัดที่ 2 โฮสต์ต้นทางชื่อ hacker.com เป็นต้น

- Port : แสดงหมายเลขของ TCP พอร์ตต้นทางว่าเป็นพอร์ตหมายเลขใด สามารถแสดงได้ 2 แบบคือ หมายเลขพอร์ตสำหรับพอร์ตที่ไม่ใช่บริการมาตรฐาน และชื่อของบริการหากหมายเลขพอร์ตนั้นเป็นพอร์ตบริการมาตรฐาน เช่น HTTP , SMTP , ECHO, CHARGEN เป็นต้น

- Destination Host , Port : เช่นเดียวกับ Source Hose, Port เปลี่ยนจากต้นทางเป็นปลายทาง

- Flags : แสดง TCP Flag ที่มาพร้อมกับแพ็กเก็ตนี้ ค่าที่แสดงมีดังต่อไปนี้ S = SYN , F = FIN , P = Push , U = Urgent ,R = Reset

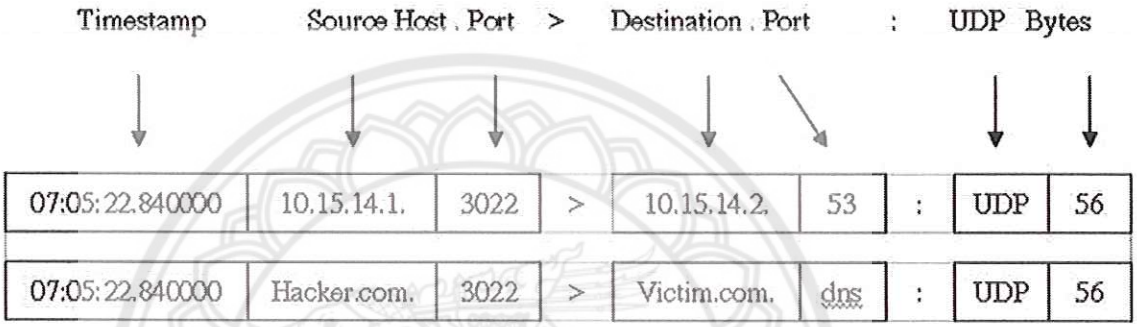
- Beginning Sequence Number : หมายเลข Initial Sequence Number (ISN) ซึ่ง TCP ใช้ในการควบคุมการสื่อสาร ทั้งในระหว่าง 3 – Way Handshacke และเมื่อสามารถเชื่อมต่อได้แล้วและเริ่มส่งข้อมูล

- Ending Sequence Number : หมายเลข ISN บวกกับขนาดของข้อมูลที่จะส่ง เพื่อเป็นการบอกว่าแพ็กเก็ตที่จะ ACK กลับมาต้องใช้หมายเลขนี้

- Bytes : ขนาดของข้อมูลที่ส่งมาพร้อมแพ็กเก็ตนี้ ในระหว่าง 3 – Way Handshacke ขนาดของข้อมูลจะเป็น 0 เสมอจนกว่าจะเริ่มสร้างข้อมูล

- Option : เป็นค่า TCP Option ที่โฮสต์ต้นทางต้องการบอกโฮสต์ปลายทางจากตัวอย่างจะเป็นการแสดงเมื่อ TCP Option ได้กำหนดค่า windows size เท่ากับ 512 ไบต์

2.7.2 โปรโตคอล UDP



รูปที่ 2.16 แสดงการวิเคราะห์ของโปรโตคอล UDP [8]

ข้อมูลที่ปรากฏประกอบด้วยฟิลด์ดังนี้

- Timestamp : แสดงเวลาที่ได้รับแพ็กเก็ตนี้ แสดงเป็นหน่วย ชั่วโมง : นาที : วินาที.เศษของวินาที

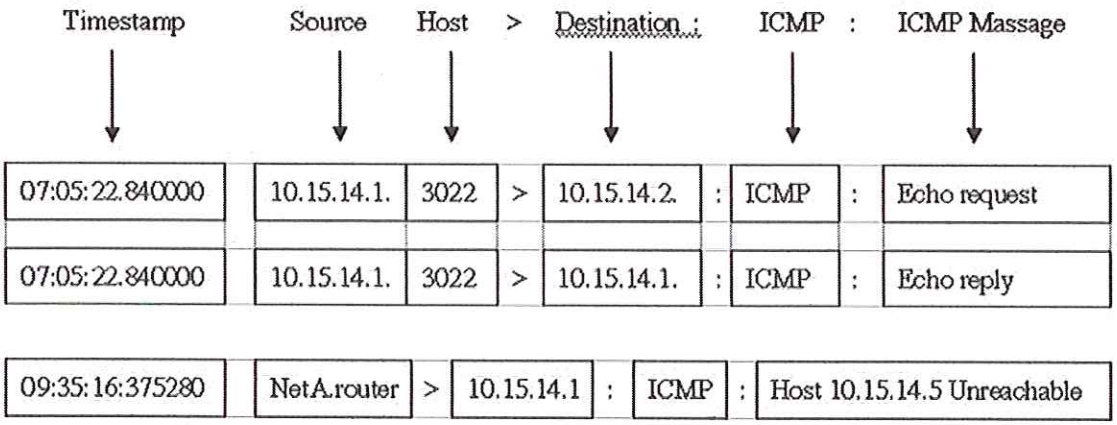
- Source Host : แสดง IP Address ต้นทางของแพ็กเก็ต สามารถแสดงเป็น 2 แบบ คือ IP Address โดยตรง เช่นในตัวอย่างคือ 10.15.14.1 หรือหากสามารถแปล IP เป็นชื่อโฮสต์ได้ก็จะแสดงเป็นชื่อโฮสต์ เช่นตัวอย่างบรรทัดที่ 2 โฮสต์ต้นทางชื่อ hacker.com เป็นต้น

- Port : แสดงหมายเลขของ UDP พอร์ตต้นทางว่าเป็นพอร์ตหมายเลขใด สามารถแสดงได้ 2 แบบคือ หมายเลขพอร์ตสำหรับพอร์ตที่ไม่ใช่บริการมาตรฐาน และชื่อของบริการหากหมายเลขพอร์ตนั้นเป็นพอร์ตบริการมาตรฐาน เช่น ECHO, NDS เป็นต้น

- Destination Host , Port : เช่นเดียวกับ Source Hose, Port เปลี่ยนจากต้นทางเป็นปลายทาง

- Bytes : ขนาดของข้อมูลใน UDP แพ็กเก็ตนี้

2.7.3 โปรโตคอล ICMP



รูปที่ 2.17 แสดงการวิเคราะห์ของโปรโตคอล ICMP [8]

ข้อมูลที่ปรากฏประกอบด้วยฟิลด์ดังนี้

- Timestamp : แสดงเวลาที่ได้รับแพ็กเก็ตนี้ แสดงเป็นหน่วย ชั่วโมง : นาที : วินาที.เศษของวินาที

- Source Host, Port : แสดง IP Address ต้นทางของแพ็กเก็ต สามารถแสดงเป็น 2 แบบคือ IP Address โดยตรง เช่นในตัวอย่างคือ 10.15.14.1 หรือหากสามารถแปล IP เป็นชื่อโฮสต์ได้ก็จะแสดงเป็นชื่อโฮสต์ เช่นตัวอย่างบรรทัดที่ 2 โฮสต์ต้นทางชื่อ hacker.com เป็นต้น สำหรับ ICMP ในบางกรณีแพ็กเก็ตอาจมีจุดกำเนิดมาจากเราต์เตอร์ได้ ทำให้ฟิลด์นี้ปรากฏเป็นของเราต์เตอร์แทน เช่นในกรณีของ ICMP Hose Unreachable เป็นต้น

- Destination Host, Port : เช่นเดียวกับ Source Hose เปลี่ยนจากต้นทางเป็นปลายทาง

- ICMP massage : หมายถึง Massage ที่ส่งมากับ ICMP แพ็กเก็ตนี้ ซึ่งโปรแกรม TCPDUMPได้ทำการแปลมาจากตารางของ ICMP โดยอัตโนมัติ แท้ที่จริงข้อมูลที่อยู่ในแพ็กเก็ตจะเป็นรหัสมา และต้องนำไปเทียบกับตารางก่อนจึงจะได้ Massage ออกมา

สำหรับโปรแกรม TCPDUMP นั้นเป็นตัวอย่างของโปรแกรมที่ใช้เพื่อจับแพ็กเก็ตทำงานอยู่บนระบบปฏิบัติการ Solaris , UNIX และ Linuk ทั่วไปสามารถนำมาใช้งานเพื่อการศึกษาการทำงานของโปรโตคอลได้เป็นอย่างดี และยังสามารถเรียกดูข้อมูลได้อย่างหลากหลาย

2.8 ดักอ่านข้อมูลด้วย Packet Sniffer[8]

Packet Sniffer เป็นเครื่องมือสำหรับการดักอ่านข้อมูลที่สื่อสารอยู่บนเน็ตเวิร์กเพื่อให้ได้มาซึ่งข้อมูลของผู้อื่นที่ไม่ใช่ของตนเอง มีลักษณะการนำไปใช้งานใกล้เคียงกับดักฟังโทรศัพท์ที่ทำโดยการนำโทรศัพท์ไปพ่วงกับเครื่องที่ต้องการ ทำให้เครื่องที่ต่อพ่วงอยู่นั้นสามารถได้ยินการ

สนทนาโต้ตอบที่เกิดขึ้นโดยที่เจ้าของโทรศัพท์เองก็ไม่รู้ตัว และมักจะเกิดขึ้นเสมอสำหรับการดักฟังความลับ แต่ข้อมูลข่าวสารบนเน็ตเวิร์กนั้นเป็นข้อมูลที่สำคัญและไม่สำคัญ มีข้อมูลหลายชนิดที่ผู้ใช้ไม่ประสงค์จะเปิดเผยต่อผู้ใดเลยแต่จำเป็นต้องสื่อสารข้อมูลเหล่านั้นไปบนเน็ตเวิร์ก เช่น รหัสผ่าน

2.8.1 องค์ประกอบของ Sniffer

คำว่า Sniffer นั้นเป็นเครื่องหมายทางการค้าซึ่งจดทะเบียนไว้โดยบริษัท Network Associates Inc. ในสหรัฐฯ เพื่อใช้ในผลิตภัณฑ์ของตนเองชื่อ Sniffer Netword Analyzer ซึ่งเป็นโปรแกรมวิเคราะห์เน็ตเวิร์กโดยอาศัยการดักอ่านข้อมูลทั้งหมดบนเน็ตเวิร์กมาทำการวิเคราะห์แยกแยะการใช้งานเน็ตเวิร์กออกไปตามโปรโตคอลที่ใช้งานกันอยู่ เพื่อช่วยในการวางแผนตรวจสอบ และแก้ไขข้อบกพร่องที่อาจมีขึ้นในเน็ตเวิร์ก แต่เนื่องจากคำนี้เป็นที่เรียกขานกันแพร่หลายจนเป็นที่เข้าใจกันว่า Sniffer เป็นเครื่องมือที่ดักอ่านข้อมูลบนเน็ตเวิร์ก ซึ่งหากจะเรียกให้ถูกต้องแล้วอุปกรณ์ประเภทนี้ควรจะเรียกว่า Network Wire Tapping Device แต่เมื่อเป็นที่เข้าใจกันและเรียกขานกันแพร่หลายแล้ว ปัจจุบันจึงทำการเรียกอุปกรณ์ชนิดนี้ว่า Sniffer

Sniffer ที่จะสามารถทำงานได้นั้นจะต้องมีองค์ประกอบพื้นฐาน 4 ส่วนคือ

- Hardware : หมายถึงอุปกรณ์อิเล็กทรอนิคส์ต่าง ๆ ที่สามารถดักอ่านสัญญาณจากเน็ตเวิร์กเข้ามาได้ และสามารถนำสัญญาณที่ได้ส่งต่อไปประมวลผลออกมาเป็นข้อมูลทางคอมพิวเตอร์ได้ มีหน้าที่หลักคือจัดการกับการรับข้อมูลในระดับฟิสิกคัล เช่นระดับแรงดัน สัญญาณรบกวน การแก้ไขข้อผิดพลาดของสัญญาณ อุปกรณ์นี้โดยทั่วไปก็คือเน็ตเวิร์กอะแดปเตอร์นั่นเอง
- Driver : เป็นโปรแกรมระดับล่างที่ควบคุมการดักข้อมูลของฮาร์ดแวร์ตามข้อ 1 และนำสัญญาณที่ได้จากฮาร์ดแวร์ไปเก็บเป็นข้อมูลดิบรอการประมวลผลในลำดับถัดไป
- Buffer : เป็นหน่วยความจำที่ใช้พักข้อมูลจากการดักมาได้ของ Driver โดยจะทำการจัดเก็บเพียงชั่วคราว และหมุนเวียนข้อมูลใหม่เข้ามาเสมอเมื่อมีข้อมูลใดปรากฏขึ้นบนเน็ตเวิร์กกลไกการนำข้อมูลจากไดรเวอร์มาเก็บยังบัฟเฟอร์นี้จะเป็นตัวบ่งบอกสมรรถนะของการดักข้อมูลของ Sniffer นั้นว่าจะสามารถดักข้อมูลได้ความเร็วสูงสุดเท่าใด หากกระบวนการนำข้อมูลไปเก็บเป็นไปอย่างล่าช้า ย่อมทำให้ Sniffer ไม่สามารถดักข้อมูลที่อยู่เน็ตเวิร์กได้ทันและต้องปล่อยข้อมูลนั้นทิ้งไป
- Software : เพื่อทำหน้าที่จัดการข้อมูลที่ได้รับเข้ามาโดยการประมวลผลตามวัตถุประสงค์ของการดักอ่านข้อมูลนั้น เนื่องจากข้อมูลดิบที่ดักอ่าน

มาได้นั้นจะเป็นข้อมูลในระดับต่ำ คือ Data Lind Layer ซึ่งจะมีข้อมูลที่ยังไม่ได้ผ่านการดีมัลติเพล็กซ์และจัดรูปแบบให้เข้าใจได้ สิ่งที่ได้จะเป็นข้อมูลเลขฐานสอง 0 กับ 1 จำนวนมหาศาลที่ต้องมาแปลความหมายกันอีก อีกประการหนึ่งคือข้อมูลที่ดับอ่านมาได้นั้นเป็นข้อมูลจากการสื่อสารของทุก ๆ โฮสต์ที่ใช้เน็ตเวิร์กนั้นร่วมกันอยู่ ผสมกันอย่างไม่มีระเบียบและไม่มีการแยกแยะว่าเป็นเป็นสื่อสารเรื่องอะไร ระหว่าโฮสต์ใดกับโฮสต์ใด การที่จะแปลความหมายของข้อมูลเหล่านี้ได้ก็จำเป็นอย่างยิ่งที่จะต้องมีโปรแกรมสำหรับทำหน้าที่จัดการกับกองข้อมูลขนาดใหญ่นี้ให้อยู่ในรูปแบบที่สามารถเข้าใจได้มากขึ้น นั่นคือทำหน้าที่คล้ายคลึงกับการดีมัลติเพล็กซ์ของโปรโตคอลปกติ แต่จะมีข้อแตกต่างคือจะเป็นการดีมัลติเพล็กซ์ของข้อมูลทุก ๆ โฮสต์โดยไม่สนใจว่าเป็นข้อมูลของโฮสต์ใด

หลังจากข้อมูลผ่านการดีมัลติเพล็กซ์แล้วก็จะอยู่ในรูปที่สามารถเข้าใจได้ง่ายขึ้น แต่จะเข้าใจได้มากน้อยขนาดไหนขึ้นอยู่กับความสามารถในการดีมัลติเพล็กซ์ของโปรแกรมนั้น ๆ หากสามารถดีมัลติเพล็กซ์ได้จนถึงโปรโตคอลเลเยอร์ที่สูงเช่น HTTP หรือ SMTP ก็จะสามารถเข้าใจได้ง่ายขึ้นรวมทั้งหากมีการจัดการแยกแยะหมวดหมู่ของการสื่อสารของแต่ละโฮสต์ก็จะเห็นความต่อเนื่องของการสื่อสารได้ดีขึ้น

2.8.2 การทำงานของ Sniffer

การที่ Sniffer สามารถดักอ่านข้อมูลที่อยู่บนเน็ตเวิร์กได้นั้นมีสาเหตุที่สำคัญคือ ด้วยลักษณะของโปรโตคอลอีเธอร์เน็ตที่ใช้หลักการกระจายของข้อมูลไปยังทุกโฮสต์ที่อยู่ในเน็ตเวิร์กและอาศัยโฮสต์แต่ละตัวทำหน้าที่จำแนกการสื่อสารของตนเอง นั่นหมายความว่าข้อมูลทุกแพ็กเก็ตที่ใช้สื่อสารกันนั้นได้ถูกส่งไปยังทุกโฮสต์ ซึ่งจะได้รับพร้อมกันและเหมือนกัน เพียงแต่การที่จะสื่อสารกันได้อย่างถูกต้องนั้นโฮสต์แต่ละตัวจะต้องมีกระบวนการที่สามารถรู้ได้ว่าข้อมูลแพ็กเก็ตใดเป็นของตนเอง และข้อมูลแพ็กเก็ตใดมิใช่ของตนเอง ทุก ๆ แพ็กเก็ตที่กระจายบนเน็ตเวิร์กนั้นจะมีหมายเลขระบุชัดเจนคือ MAC Address หรือเรียกอีกอย่างหนึ่งว่า Ethernet Address ซึ่งจะเป็นสิ่งที่บอกว่าแพ็กเก็ตมาจากฮาร์ดแวร์ใดในเน็ตเวิร์กทำให้สามารถระบุได้ว่าแพ็กเก็ตนั้นส่งมาจากโฮสต์ใด และต้องการส่งให้โฮสต์ใด

MAC Address จะเป็นหมายเลขเฉพาะตามฮาร์ดแวร์ทุกชนิดที่ใช้การสื่อสารโดยโปรโตคอลอีเธอร์เน็ต และในทางทฤษฎีแล้วฮาร์ดแวร์ทุกชนิดจะไม่มี MAC Address ที่ซ้ำกัน โดยทั่วไป MAC Address จะถูกกำหนดตายตัวอยู่ใน ROM ของฮาร์ดแวร์และไม่สามารถเปลี่ยนแปลงได้โดยซอฟต์แวร์ แต่ในที่สุดก็ไม่มีสิ่งใดเหนือความสามารถของมนุษย์สามารถกำหนด MAC Address ได้ มนุษย์ก็ย่อมจะสามารถเปลี่ยนแปลงได้เช่นกัน

การใช้งานของฮาร์ดแวร์จะต้องควบคู่กับไดรเวอร์ของฮาร์ดแวร์นั้น ๆ โดยปกติแล้วไดรเวอร์จะถูกกำหนดให้ปฏิบัติตามโปรโตคอลอย่างเข้มงวดคือ

- ให้รับข้อมูลที่มี MAC Address เป็นของตนเองเท่านั้น (ห้ามอ่านข้อมูลผู้อื่น)
- ให้ส่งข้อมูลโดยใช้ MAC Address ของตนเองเท่านั้น (ห้ามปลอมเป็นผู้อื่น)

ดังนั้นหากให้ไดรเวอร์ตามปกติที่มากับฮาร์ดแวร์แล้วเครื่องคอมพิวเตอร์ก็จะทำงานอยู่ในระเบียบเรียบร้อย และไม่สามารถวุ่นวายกับข้อมูลของผู้อื่นได้ แต่อย่างไรก็ตามไดรเวอร์ก็เป็นเพียงโปรแกรมประเภทหนึ่งเท่านั้น ที่ถูกเขียนขึ้นและถูกกำกับโปรโตคอล แต่ถ้ามีคนเขียนไดรเวอร์ที่ไม่จำกับทางโปรโตคอลก็จะรับข้อมูลของคนอื่นเข้ามาได้โดยทันที มีโหมดที่โดยอนุญาตให้ฮาร์ดแวร์รับข้อมูลของผู้อื่นเข้ามาได้โดยไม่มีการปิดกั้นเรียกว่า โพรมิสคูอัสโหมด (Promiscuous Mode) Sniffer จะอาศัยการทำงานในโหมดนี้เพื่อวิเคราะห์แพ็กเก็ตต่าง ๆ ที่เข้ามาในระบบเน็ตเวิร์ก

2.8.3 การป้องกันการถูกดักอ่านข้อมูลโดย Sniffer

เนื่องจาก Sniffer เมื่อถูกติดตั้งลงบนระบบเน็ตเวิร์ก จะไม่ส่งผลกระทบอะไรกับระบบเน็ตเวิร์ก จึงทำการตรวจสอบภายนอกได้ยาก เว้นแต่ว่าเข้าไปตรวจสอบการทำงานโหมดการทำงานของเน็ตเวิร์กอะแดปเตอร์ใน ทุก ๆ โฮสต์ที่อยู่บนเน็ตเวิร์กเดียวกัน ขอให้จำไว้ว่าการทำงานปกติของเน็ตเวิร์กอะแดปเตอร์จะไม่ทำงานอยู่ในโพรมิสคูอัสโหมด หากพบว่ามีโฮสต์ใดทำงานอยู่ในโหมดนี้ก็ให้สันนิษฐานได้ทันทีว่าโฮสต์นั้น ๆ ได้ทำงานเป็น Sniffer คอยแอบอ่านข้อมูลผู้อื่น

สิ่งที่ผู้ใช้ควรจะคำนึงถึงอยู่เสมอเมื่อใช้งานเน็ตเวิร์กคือ

- หลีกเลี่ยงการใช้งานแอพพลิเคชั่นที่รับส่งข้อมูลที่รับส่งข้อมูลโดยไม่ได้เข้ารหัส (Plain Text) เช่นโปรแกรมเทลเน็ต, FTP, ฟิงเกอร์, IRC เพราะข้อมูลสามารถูดักอ่านและเข้าใจได้อย่างรวดเร็ว
- ให้ระลึกอยู่เสมอว่าข้อมูลที่ส่งผ่านเน็ตเวิร์กนั้นสามารถถูกดักอ่านได้ จงประเมินความเสียหายมากที่สุดที่ยอมรับได้หากข้อมูลนั้นถูกเปิดเผย หากข้อมูลมีความสำคัญมาก ควรจะหามาตรการรักษาความปลอดภัยชนิดอื่นมาเสริมก่อนที่จะทำการส่งข้อมูล อย่างคำนึงแต่ความสะดวกอย่างเดียว
- หากมีการใช้บริการกับเว็บ server ที่เกี่ยวข้องกับการเงิน หรือข้อมูลรหัสผ่าน ให้เลือกใช้ผู้บริการที่เข้ารหัสข้อมูลด้วย SSL
- เป็นไปได้ให้ลดการใช้สื่อสัญญาณร่วมกันให้น้อยที่สุด ปัจจุบัน Switched Hub (หรือ ที่เรียกกันว่า Switch) มีราคาไม่สูงมาก หากนำมาเปลี่ยนกับ Shared Hub (คือ Hub ธรรมดา) ที่มีใช้งานอยู่ จะช่วยลดความเสี่ยงของการถูกดักอ่านข้อมูลได้มาก เพราะใน Switched Hub นั้นข้อมูลจะไม่กระจาย คือจะวิ่งจากต้นทางไปยังปลายทางที่ระบุเท่านั้น

- หากสามารถเพิ่มความปลอดภัยของข้อมูลได้ด้วยการเข้ารหัสข้อมูลก่อนส่งก็ควรจะนำมาใช้งาน เพื่อทำให้ถึงแม้ว่ามีผู้สามารถดักอ่านข้อมูลได้ แต่ก็จะไม่สามารถถอดรหัสของข้อมูลให้เข้าใจได้โดยง่าย
- หากมีการสื่อสารข้อมูลภายในองค์กรโดยผ่านอินเตอร์เน็ต การนำเทคโนโลยีของ VPN (Virtual Private Network) มาใช้งานจะช่วยเพิ่มระดับความปลอดภัยให้สูงขึ้นได้

2.8.4 การใช้ประโยชน์จาก Sniffer

หากไม่พูดถึงการไปแอบอ่านข้อมูลของผู้อื่นแล้ว sniffer ก็ไม่ใช้สิ่งเลวร้ายนัก อีกทั้งยังสามารถนำโปรแกรมที่มีรากฐานมาจาก Sniffer ไปใช้ประโยชน์ได้หลายด้านอาทิเช่น

- Network Analyzer การที่ Sniffer สามารถดักข้อมูลทั้งหมดที่มีอยู่เน็ตเวิร์กได้ ทำให้สามารถนำข้อมูลดังกล่าวมาทำการประมวลผลวิเคราะห์คุณสมบัติต่าง ๆ ของเน็ตเวิร์กได้หลายแง่มุม ไม่ว่าการกระจายการใช้งานเน็ตเวิร์กของโปรโตคอลต่าง ๆ เช่น HTTP, SMTP, FTP เพื่อจะได้ทราบว่ามีการใช้งานเน็ตเวิร์กไปเพื่อการใด มากน้อยเพียงไร จะทำให้สามารถจัดการเน็ตเวิร์กได้อย่างเหมาะสม หรือในด้านของการนำใช้งานแบนด์วิดธ์ไปในทางที่เป็นประโยชน์หรือไม่หรือแม้กระทั่งการวิเคราะห์เพื่อตรวจสอบว่าเน็ตเวิร์กยังคงทำงานได้ตามปกติหรือไม่
- NetWork Debugging Tools ในบางครั้งการหาสาเหตุความผิดปกติของแอพพลิเคชั่นต่าง ๆ จำเป็นที่จะต้องสืบค้นลึกลงไปถึงเนื้อข้อมูลที่รับส่งกันจริง ๆ บนระบบเน็ตเวิร์กเพื่อหาว่าเหตุใดการทำงานของแอพพลิเคชั่นจึงไม่สามารถทำงานได้ตามปกติ การได้เห็นข้อมูลดิบที่รับส่งกันจริงบนเน็ตเวิร์กจะช่วยให้การวิเคราะห์และแก้ไขปัญหาสามารถดำเนินการได้อย่างรวดเร็วและตรงต่อปัญหาได้มากขึ้น โดยเฉพาะกรณีที่มีการใช้เครื่องมือในระดับเน็ตเวิร์กมาเกี่ยวข้อง เช่น มีการส่งข้อมูลผ่านไฟร์วอลล์แล้วมีปัญหา หรือการทดสอบ ACL (Access Control List) ของ Router เป็นต้น ซึ่งปัญหาในระดับนี้หากไม่มี Sniffer เพื่อจับข้อมูลดิบขึ้นมาวิเคราะห์ดูแล้ว อาจจะต้องใช้เวลานานมากกว่าที่จะแก้ไขปัญหาได้
- Packet Monitoring การศึกษาเทคนิคของโปรโตคอลในระดับเน็ตเวิร์กจำเป็นต้องเห็นข้อมูลที่สื่อสารกันอยู่จึงจะสามารถเข้าใจได้และเห็นภาพจริง แพ็กเก็ตมอนิเตอร์ริ่งจะเป็นเครื่องมือที่นำแพ็กเก็ตมาแสดงให้เห็นเทคนิคการสแกนต่าง ๆ ของแฮกเกอร์ล้วนแล้วแต่ได้มาจากการทำงาน

ของแพ็กเก็ตมอนิเตอร์ริ่ง หากขาดโปรแกรมประเภทนี้แล้ว ก็จะไม่มีเครื่องมือที่ใช้ในการศึกษาโปรโตคอลได้ดี

- Intrusion Detection System ระบบตรวจจับการบุกรุกก็มีพื้นฐานมาจากการดักอ่านข้อมูล เพียงแต่เป็นการประยุกต์การวิเคราะห์รูปแบบการบุกรุกเข้ากับการดับอ่านข้อมูลบนเน็ตเวิร์กแบบเรียลไทม์ ทำให้สามารถรู้ได้ว่ามีกิจกรรมใดบนเน็ตเวิร์กที่มีแนวโน้มว่าจะเป็นการบุกรุกหรือทำอันตรายต่อผู้อื่น
- Other Security Tools นอกจากนี้ Sniffer ยังสามารถนำไปประยุกต์ใช้ไปในงานรักษาความปลอดภัยอย่างกว้างขวาง เช่น เพื่อบันทึกร่องรอยของแฮกเกอร์ จับตาดูการใช้งานทรัพยากรที่มีความสำคัญ ทดสอบความแข็งแรงของไฟร์วอลล์ เป็นต้น

2.9 เครื่องมือที่ใช้ในการบุกรุกและการโจมตีของแฮกเกอร์[8]

2.9.1 พื้นฐานทั่วไปที่แฮกเกอร์ใช้ในการบุกรุก

2.9.1.1 การกระตุ้นและการตอบรับ(Stimulus And Response)

การกระตุ้น (Stimulus) และการตอบรับ (Response) เป็นกลไกส่วนหนึ่งที่สำคัญในการเจาะเข้าไปยังระบบต่างๆ ในกรณีการใช้งานคอมพิวเตอร์ปกติสามารถจะทำการกระตุ้นและตอบรับระหว่างกันตลอดเวลา และการกระทำดังกล่าวไม่ได้ก่อให้เกิดความเสียหายแต่อย่างใด ตัวอย่างเช่น คำสั่ง Ping เป็นการตรวจสอบสถานะคอมพิวเตอร์ โดยการส่งคำสั่งไป และรอการตอบรับ (Reply) การทำงานใน ICMP มีกระบวนการดังนี้

- ส่ง ICMP echo ไปยังคอมพิวเตอร์ B (เป็นการกระตุ้น)
- เมื่อคอมพิวเตอร์ B ได้รับ ICMP echo ก็จะตอบรับกลับไปด้วย ICMP Reply ไปยังคอมพิวเตอร์ A
- เมื่อคอมพิวเตอร์ A ก็จะทราบทันทีว่าคอมพิวเตอร์ B ทำงานอยู่และสามารถสื่อสารผ่านเครือข่ายได้ตามปกติ

หรือกรณีการส่งข้อมูลผ่าน TCP มีขั้นตอนดังนี้

- เครื่องคอมพิวเตอร์ A กระตุ้นขอทำการสื่อสาร โดยส่ง SYN , ISN ไปยังคอมพิวเตอร์ B
- คอมพิวเตอร์ B เมื่อได้รับ ก็จะตอบรับด้วย SYN + ACK ตามข้อตกลงของ TCP ไปยังคอมพิวเตอร์ A

- และเช่นกันเมื่อคอมพิวเตอร์ A ก็จะทราบทันทีว่าคอมพิวเตอร์ B ทำงานอยู่และสามารถสื่อสารผ่านเครือข่ายได้ตามปกติ

ด้วยข้อบกพร่องดังกล่าวการร้องขอหรือกระตุ้นไปที่โฮสต์ โฮสต์จะตรวจสอบว่าถูกต้องตามโปรโตคอลหรือไม่เท่านั้น ไม่สามารถกำหนดเรื่องอื่นได้เช่น มีการร้องขอมากไหม กี่ครั้ง หรือมีความบ่อยมากแค่ไหน และการตอบรับนี้ก็เหมือนเป็นดาบสองคม กล่าวคือทำให้แฮกเกอร์สามารถวิเคราะห์เป้าหมายได้ ดังในตัวอย่างต่อไปนี้

ตารางที่ 2.1 ตัวอย่างการวิเคราะห์เป้าหมาย

วิธีการกระตุ้น	การตอบรับที่คาดว่าจะได้รับ	ผลการวิเคราะห์
ส่ง TCP SYN พอร์ต 80	TCP SYN +ACK	1.โฮสต์เป้าหมายทำงานอยู่ 2.เป้าหมายมีการให้บริการโดยใช้พอร์ต 80 3.สันนิษฐานว่าเป็นเว็บเซิร์ฟเวอร์(เนื่องจากเว็บเซิร์ฟเวอร์ใช้พอร์ต 80 ในการให้บริการ
	RST	1.โฮสต์เป้าหมายทำงานอยู่ 2.เป้าหมายไม่มีการให้บริการที่ใช้พอร์ต 80
	ICMP HOST Unreachable	1.โฮสต์เป้าหมายไม่ทำงาน ปิดเครื่องอยู่

ดังตัวอย่างที่เห็นในตารางนั้น สามารถที่จะหาได้มาง่ายๆโดยใช้วิธีการสแกนพอร์ตหรือที่เรียกว่าพอร์ตสแกน หรือเน็ตเวิร์กสแกน การสแกนนี้ก็คือการส่งคำสั่ง TCP SYN หรือ ICMP Echo ไปยังทุกๆ พอร์ตของเป้าหมายนั่นเอง ก็จะได้การตอบรับกลับมาทำให้นำมาวิเคราะห์ได้

2.9.1.2 ความสำคัญของพอร์ต

พอร์ตเป็นช่องทางการสื่อสารของ TCP/IP และทำหน้าที่แยกไปตามแต่ละจุดประสงค์ของข้อมูลของแต่ละแอพพลิเคชั่นเพื่อไม่ให้ข้อมูลปะปนกัน

พอร์ตของ TCP/IP พอร์ตนี้หมายถึงทั้ง TCP Port และ UDP Port ทำหน้าที่เกี่ยวข้องกับการสื่อสารต่างๆนั่นเอง โดยพอร์ตทั้งสองนั้นมีอย่างละ 65,534 พอร์ต โดยทั่วไปพอร์ตมีสองสถานะ คือเปิด และปิด พอร์ตเปิด หมายถึงพอร์ตที่ใช้งานได้ตามปกติสามารถสื่อสารกันได้ และพอร์ตปิด คือพอร์ตที่จะไม่สามารถที่จะสื่อสารกันได้นั่นเองและต้องมีฝ่ายใดฝ่ายหนึ่งเปิดพอร์ตขึ้น มาก่อนเรียกว่าเซิร์ฟเวอร์พอร์ต(server port) เพื่อรอรับการติดต่อ และฝ่ายที่ส่งสัญญาณ

มาขอติดต่อเรียกว่าไคลเอนต์พอร์ต(client port) และหากไม่มีเซิร์ฟเวอร์พอร์ตการสื่อสาร TCP /IP ก็จะไม่สามารถเริ่มต้นได้

การเปิดพอร์ตนั้นต้องมีแอพพลิเคชั่นที่มีหน้าที่ตอบรับการ และจัดการการสื่อสารต่าง ๆ ที่ร้องขอมายังพอร์ตนั้นๆรองรับ โดยมาตรฐานของ TCP/IP ก็จะมีแอพพลิเคชั่นที่ทำหน้าที่คือ FTP, SMTP, HTTP, POP, TIME และ TELNET เป็นต้น ซึ่งสามารถศึกษาได้จากตารางแสดงหมายเลขพอร์ต [ภาคผนวก ก.] ที่นิยมใช้งานทั่วๆไป ในปัจจุบัน

พอร์ตของระบบปฏิบัติการ พอร์ตนี้เกี่ยวข้องกับระบบปฏิบัติการถ้าไม่เปิดก็หมายความว่าจะไม่สามารถใช้ได้ เช่น Microsoft Windows NT จะต้องใช้พอร์ตหมายเลข 135-139 ของ TCP ในการทำงาน ซึ่งไม่สามารถปิดได้เนื่องจากพอร์ตนี้เป็นส่วนหนึ่งของระบบปฏิบัติการ

พอร์ตอันตราย กล่าวคือเป็นโปรแกรมที่ถูกเขียนมาเพื่อจุดประสงค์ในการโจมตีนั่นเอง ได้แก่พอร์ตต่างๆ ดังต่อไปนี้

- ม้าโทรจัน (Trojan Horse) โปรแกรมประเภทโทรจันเป็นโปรแกรมที่แอบแฝงมาในคราบของโปรแกรมปกติธรรมดาที่ใช้งานทั่วๆไป และแอบทำหน้าที่อื่นๆ เช่น ดักจับรหัสผ่าน เก็บข้อมูลการกดแป้นพิมพ์ เป็นต้น รวมถึงที่เกี่ยวข้องกับการเปิดพอร์ตเพื่อทำการส่งข้อมูลต่างๆ ไปยังที่ใดที่หนึ่งบนอินเตอร์เน็ต
- แบ็คออริฟิซ(Back Orifice) หรือ แบ็คดอร์(Back Door) หน้าที่ของโปรแกรม นี้คือ เมื่อทำงานอยู่บนเครื่องใดแล้ว ก็จะแอบเปิดพอร์ตให้ไคลเอนต์จากภายนอกสามารถควบคุมเครื่องได้

2.9.1.3 การใช้ข้อมูลจากพอร์ตเพื่อเจาะระบบ

โดยสัญญาณเริ่มต้นก็คือการสแกนพอร์ต โดยแฮกเกอร์จะทราบรายละเอียดต่างๆ มากมาย เช่น โฮสต์ที่ถูกสแกนใช้ระบบปฏิบัติการอะไร มีแอพพลิเคชั่นได้ทำงานอยู่บ้าง เป็นต้น และแฮกเกอร์ก็จะอาศัยช่องว่างของระบบต่างๆเพื่อเข้ามาบุกรุกเป้าหมายนั่นเอง

2.9.1.4 การทำแผนที่เป้าหมายอย่างละเอียด

- Individual Host Ping (Ping Sweep) การ ping ลักษณะนี้เป็นการ ping แบบกวาดไปทั่วทั้งระบบ เพื่อสำรวจพื้นฐานเพื่อตรวจสอบดูว่าระบบเน็ตเวิร์กเป็นปกติหรือไม่โดยการส่ง ICMP Echo Request ไปและได้รับการตอบกลับด้วย ICMP Echo Reply กลับมายังโฮสต์ ซึ่งสามารถจะจัดกลุ่มทำแผนที่ได้

- Broadcast Ping วิธีนี้คือ การ ping ไล่ไปทีละโฮสต์จนหมดเน็ตเวิร์ก หากการสำรวจเป้าหมายที่มีขนาดใหญ่ก็จะใช้เวลานาน กระบวนการของ Broadcast Ping ก็ยังคงเหมือนเดิมโดยส่ง ICMP Echo Request ไปยัง Broadcast Addressและได้รับการตอบกลับด้วย ICMP Echo Reply กลับมายังโฮสต์โดยพร้อมกัน

- Subnet Broadcast Ping การกระทำก็เช่นเดียวกัน แต่บอกได้ว่าเป้าหมายใช้ Subnet ด้วย ดังในตัวอย่างเป็นการส่ง ICMP Echo Request จากโฮสต์ Scanner.comไปยัง Broadcast Address ย่อยที่ละเน็ตเวิร์กของ 192.168.21.0

จากตัวอย่างในข้างต้นจะเห็นได้ว่าเป็นการง่ายมากที่จะทำการออกแบบแผนที่เครือข่าย และวิเคราะห์เป้าหมาย โดยใช้การทำงานพื้นฐานของการตรวจสอบการทำงานของเน็ตเวิร์กนั่นเองซึ่งการกระทำดังกล่าวจะเป็นผลดีเพราะช่วยให้ผู้ดูมระบบสามารถทราบได้ว่าระบบเครือข่ายนั้นทำงานได้ปกติหรือไม่ แต่ในทางกลับกันก็เป็นเครื่องมือที่มีประสิทธิภาพและสำคัญของพวกที่มุ่งร้ายต่อระบบ หรือเหล่าแฮกเกอร์นั่นเอง

2.9.2 สแกนพอร์ต TCP/IP

2.9.2.1 ความสำคัญของการสแกนพอร์ต

การสแกนพอร์ตนี้จะเป็นการสำรวจว่าแต่ละโฮสต์ โดยมีขอบเขตเฉพาะโฮสต์เดียว และเป็นขั้นตอนถัดมาจากการสำรวจเน็ตเวิร์กแล้ว เพราะการสำรวจเน็ตเวิร์กเพียงอย่างเดียวนั้นมีข้อมูลไม่เพียงพอจะทำให้การเจาะระบบได้แค่เพียงสามารถทราบพิกัดตำแหน่งเท่านั้น โดยข้อมูลในการสแกนพอร์ตนั้นจะช่วยระบุเป้าหมายได้ชัดเจนขึ้น

แต่เนื่องจากการสแกนพอร์ตนั้นเป็นการกระทำที่มีเจตนามุ่งร้ายอย่างชัดเจน ดังนั้นเทคนิคในการสแกนพอร์ตจึงมีวิธีที่ซับซ้อนขึ้นเรื่อย ๆ เพื่อมิให้เป้าหมายนั้นสามารถตรวจจับได้ โดยจะอธิบายตั้งแต่เทคนิคที่เป็นพื้นฐานไปจนซับซ้อนต่อไปดังนี้

2.9.2.1.1 วิธี Connect Request

วิธีนี้เป็นวิธีพื้นฐานที่สุดก็คือ การทำทีเสมือนว่าต้องการติดต่อไปยังแอพพลิเคชั่นที่ทำงานอยู่บนเซิร์ฟเวอร์ โดยส่งสัญญาณไปขอเริ่มสื่อสารกับพอร์ตเป้าหมายบนเซิร์ฟเวอร์ แล้วรอผลตอบกลับจากพอร์ตนั้น ๆ ว่า จะตอบรับคำขอหรือไม่ ซึ่งเป็นการกระทำที่มิได้เป็นการพยายามที่จะสแกนพอร์ตแต่อย่างไร เพราะเป็นการติดต่อตามปกติ ทำให้ต้องพิจารณาหลาย ๆ แพ็กเก็ตประกอบกันจึงพอประเมินความมุ่งหมายของผู้ส่งแพ็กเก็ตได้ว่ากำลังทำการสแกนพอร์ตอยู่

ข้อสังเกตที่จะสรุปว่าเป็นการสแกนพอร์ตหรือไม่คือ

- มีความพยายามในการเริ่มต้นติดต่อกับพอร์ตที่ไม่ให้บริการมาตรฐาน
- แพ็กเก็ตที่ติดต่อมามีเวลาใกล้เคียงกันและจากที่เดียวกัน
- แพ็กเก็ตที่เข้ามาอย่างต่อเนื่องและจำนวนมาก

2.9.2.1.2 วิธี SYN Scan

การสแกนพอร์ตแบบ SYN Scan หากพิจารณาดูจากแพ็กเก็ตที่ส่งไปยังเซิร์ฟเวอร์โดยผิวเผินจะใกล้เคียงกับวิธี Connect Request แต่สิ่งที่แตกต่างกันคือ วิธีนี้ผู้สแกนจะทำการส่ง SYN แพ็กเก็ตมาเพื่อทำการติดต่อเองโดยตรงกับเป้าหมายโดยไม่ผ่านระบบปฏิบัติการ และรอผลการตอบรับของเป้าหมายกลับมา ซึ่งจะมีอยู่สองแบบคือ หากมีแอพพลิเคชั่นทำงานอยู่ก็จะตอบกลับมาด้วย SYN ACK หรือถ้าไม่มีการทำงานก็จะส่งสัญญาณตอบกลับมาด้วย RST

เมื่อมีการตอบรับด้วย SYN ACK จากเป้าหมายกลับมา ระบบปฏิบัติการของโฮสต์ของผู้สแกนก็จะทำการตอบรับไปอีกครั้งด้วย RST เนื่องจากระบบไม่ได้เป็นผู้ส่งแพ็กเก็ตไปทำการติดต่อ

การสแกนแบบนี้เป็นฝ่ายดีต่อแฮกเกอร์เพราะว่า เวลาที่ทำการติดต่อไปแล้ว ไม่สามารถทำ 3-Way handshake ได้สำเร็จ จึงทำให้ทางโฮสต์นั้นไม่สามารถบันทึก log ได้ จึงไม่สามารถบันทึกการสแกนพอร์ตได้เพราะการเชื่อมต่อไม่สำเร็จ

2.9.2.1.3 วิธี FIN SCAN

โดยปกติ FIN เป็นแพ็กเก็ตของ TCP ที่ส่งเมื่อยุติการติดต่อ นั่นหมายถึงจะต้องมีการสื่อสารกันมาก่อนแล้ว แต่ FIN Scan จะเป็นการส่ง FIN แพ็กเก็ตไปยังเป้าหมายโดยไม่มีการสื่อสารใด ๆ มาก่อนเลย การFIN โดยไม่ SYN มาก่อนนั้นสามารถตรวจสอบได้ไม่ยากแต่ต้องทำบนตัวโฮสต์นั้นเอง เพราะโฮสต์ปลายทางเท่านั้นที่จะรู้ว่าตนเองมีการสื่อสารกับใครบนพอร์ตใดอยู่และมีสถานะอย่างไร โอสต์ทั่วไปจะมีหน่วยความจำที่เก็บสถานะการสื่อสารกับโฮสต์อื่น ๆ อยู่ว่าไปถึงไหนแล้ว เพื่อเวลามีแพ็กเก็ตเข้ามาจะได้ตอบถูกตามโปรโตคอล แน่นอนว่าโฮสต์ก็ต้องทราบว่าไม่เคยติดต่อกับ IP Address นี้ แต่อย่างไรก็ยังคงตอบ FIN แพ็กเก็ตกับอยู่ดี

2.9.2.1.4 วิธี SYN/FIN Scan

เป็นวิธีที่ใช้ทั้ง SYN และ FIN ในการสแกนพอร์ตไปพร้อม ๆ กันโดยที่โฮสต์จะทำการตอบกลับเป็น SYN ACK หรือ FIN ACK ขึ้นอยู่กับว่าโฮสต์ให้ความสำคัญของ Flag ไหนมากกว่า

2.9.2.1.5 วิธี NULL SCAN

วิธีนี้จะไม่ใช้ flag ใดๆ ในการสแกนเลย โดยส่งแพ็กเก็ตที่ไม่มี flag ใดที่ถูกเซตไว้เลยไปยังเป้าหมายโดยทั่วไปแล้วประเภทนี้ไม่ได้อยู่ในโปรโตคอล

จึงไม่มีผู้สนใจ การส่ง Flag แบบนี้โดยทั่วไปแล้วการตอบรับกับมาของแต่ละระบบปฏิบัติจะมีการตอบสนองไม่เหมือนกัน

2.9.3 Denial of Services Attack

Denial of Services เป็นการโจมตีเป้าหมายด้วยวิธีการต่าง ๆ เพื่อมิให้เป้าหมายสามารถให้บริการได้ตามปกติ ส่วนใหญ่จะอาศัยข้อบกพร่องของโปรโตคอลกับระบบปฏิบัติการที่ทำงานอยู่บนเป้าหมาย ร่องรอยที่เกิดขึ้นจากการโจมตีลักษณะนี้ก็จะสามารถตรวจพบได้จากลักษณะต่าง ๆ ของแพ็กเก็ตบนเน็ตเวิร์ก

เนื่องจากการค้นพบการ Dos แต่ละวิธีนั้นเป็นไปอย่างไม่เป็นทางการ และมีที่มาแตกต่างกันหลากหลาย ในที่นี้จึงอ้างอิงกับมาตรฐานที่ยอมรับกันทั่วไป โดยการอ้างอิงกับมาตรฐานของ CERT (Computer Emergency Response Team) แห่งประเทศอเมริกา เป็นศูนย์กลางการแก้ปัญหาภัยคุกคามบนอินเตอร์เน็ตของอเมริกา และมาตรฐาน CVE (Common Vulnerablilties and Exposure) ซึ่งรวบรวมข้อบกพร่องจากแหล่งต่าง ๆ พร้อมจัดหมวดหมู่ให้อยู่ในฐานการอ้างอิงเดียวกัน

2.9.3.1 Anomalous Packet

อะนอมาบัสแพ็กเก็ต (Anomalous Packet) หมายถึงแพ็กเก็ตประหลาดที่ไม่มีโอกาสเกิดขึ้นในสภาวะปกติได้อย่างสิ้นเชิง แพ็กเก็ตประเภทนี้เป็นการจงใจเปลี่ยนข้อมูลสำคัญที่ใช้ควบคุมการสื่อสารข้อมูลให้ผิดปกติ ไม่ว่า IP, UDP หรือ TCP ต่างก็มีข้อกำหนดอยู่ในโปรโตคอลของตนเอง ผู้ที่จะใช้งานโปรโตคอลนี้ต้องปฏิบัติตามจึงจะสามารถสื่อสารข้อมูลกันได้ ในแต่ละโปรโตคอลย่อยมีค่าในเฮดเดอร์ที่จะใช้เป็นกลไกการควบคุมการสื่อสารข้อมูล แต่ละอะนอมาลัสแพ็กเก็ตเหล่านี้จะเป็นแพ็กเก็ตที่ถูกดัดแปลงด้วยเทคนิคของการควบคุมเน็ตเวิร์กเลเยอร์โดยตรงแบบไม่ผ่านโปรโตคอล เปรียบเสมือนการเรียกใช้งานระบบปฏิบัติการให้ทำงานนอกเหนือจากเงื่อนไขที่ได้กำหนดไว้ตามปกติ ดังนั้นหากระบบปฏิบัติการสามารถจัดการกับแพ็กเก็ตประเภทนี้ได้ดีก็อาจจะไม่มีผลกระทบมากนัก แต่หากระบบปฏิบัติการไม่สามารถจัดการกับแพ็กเก็ตเหล่านี้ได้ก็จะส่งผลกระทบอย่างรุนแรง

2.9.3.2 Ping Flood Attack

Ping Flood เป็นการโจมตีที่อาศัยปริมาณแพ็กเก็ตมาก ๆ เพียงอย่างเดียว แต่ถึงกระนั้นก็ตาม การโจมตีวิธีนี้ก็สร้างความเสียหายได้ไม่น้อย

หลักการโจมตีของ Ping Flood คือการส่ง ICMP Echo Request (แบบเดียวกับที่ได้จากคำสั่ง Ping) ปริมาณมาก ๆ ไปยังเป้าหมายอย่างรวดเร็ว ทำให้โฮสที่ถูกโจมตีจะต้องคอยตอบ ICMP Echo Reply ตลอดเวลาจนแทบจะไม่มีเวลาจะทำงานอื่น ความรุนแรงของการโจมตีจะมากหรือน้อยขึ้นอยู่กับความเร็วของการส่ง หากแฮกเกอร์มีแบนด์วิดธ์มาก และมีเครื่องที่มีสมรรถนะสูง

ซึ่งสามารถสร้าง ICMP แพ็กเก็ตได้ในปริมาณมากในเวลาอันสั้น เครื่องเป้าหมายก็จะหยุดทำงานลงได้

นอกจากการสร้างความเสียหายแก่เครื่องคอมพิวเตอร์เป้าหมายแล้ว Ping Flood ยังสร้างความเสียหายให้แก่เน็ตเวิร์กที่เครื่องคอมพิวเตอร์เป้าหมายตั้งอยู่ด้วย เพราะที่เลเยอร์ที่ต่ำลงไป เช่น Ethernet ต่างก็ใช้งานร่วมกันกับเครื่องโฮสต์อื่น ๆ

การป้องกัน Ping Flood ทำได้โดยการไม่อนุญาตให้แพ็กเก็ตของ ICMP Echo เข้าไปยังเน็ตเวิร์กได้โดยการกำหนดที่เราต์เตอร์หรือไฟวอลล์ แต่อย่างไรก็ตาม ICMP Echo ก็ถูกใช้สำหรับโปรแกรม Ping เพื่อตรวจสอบสถานะของเซิร์ฟเวอร์อยู่เป็นปกติ หากปิดไม่ให้ ICMP Echo ผ่านเข้าไป โปรแกรม Ping ก็จะทำงานไม่ได้เช่นกัน ดังนั้นควรคำนึงถึงการป้องกันบางอย่างมีผลต่อการทำงานปกติ ดังเช่นการปิดการ Ping

2.9.3.3 SYN Flood Attack

SYN Flood Attack เป็นการโจมตีในระดับ TCP โดยใช้การ 3-ways handshake โดยเริ่มต้น ส่งสัญญาณ SYN มายังเซิร์ฟเวอร์ และเซิร์ฟเวอร์ก็จะทำตอบสัญญาณ SNY ACK ไปยังผู้ขอ จากนั้นก็รอการตอบรับอีกครั้งหนึ่งจึงจะจบกระบวนการ เนื่องจากเซิร์ฟเวอร์จะต้องจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อรองรับการเชื่อมต่อจนกว่าการทำ 3-ways handshake จะสิ้นสุดลง จึงจะมีเวลาค่าหนึ่งที่จะรอให้ได้สัญญาณ ACK ตอบกลับมา หากถึงเวลาที่กำหนดแล้วไม่มีแพ็กเก็ตของ ACK กลับมาเซิร์ฟเวอร์จะต้องยุติการรอนั้นและคืนหน่วยความจำให้แก่ระบบปฏิบัติการ

สมมุติว่าเซิร์ฟเวอร์ต้องจัดสรรหน่วยความจำ 1 กิโลไบต์สำหรับการเชื่อมต่อหนึ่งครั้งโดยที่มีเวลา (Time Out) 10วินาที ดังนั้นหากภายในเวลา 10 วินาที เซิร์ฟเวอร์ได้รับสัญญาณ SYN 1,000 ครั้ง ก็จะต้องจัดสรรหน่วยความจำอย่างน้อยที่สุด 1 เมกะไบต์เพื่อรองรับการเชื่อมต่อ และในสถานการณ์ของการโจมตีด้วย SYN Flood แฮกเกอร์จะพยายามส่ง SYN มายังเป้าหมายด้วยความเร็วสูงสุดที่แบนด์วิดธ์มีเช่น 5,000 ครั้งต่อวินาที ดังนั้นภายในเวลา 10 วินาทีก่อนที่จะ Time Out เซิร์ฟเวอร์จะต้องใช้หน่อยความจำถึง 50 เมกะไบต์ เพื่อการนี้ ซึ่งแน่นอนว่าเป็นไปได้ยากที่เซิร์ฟเวอร์ทั่วไปจะสามารถจัดสรรให้ได้

เนื่องจาก SYN Flood เป็นการโจมตีตรงไปยังระบบปฏิบัติการ ดังนั้นวิธีป้องกันที่ดีที่สุดก็คือการปรับปรุงระบบปฏิบัติการให้ทันสมัยที่สุด สำหรับเราต์เตอร์จะไม่สามารถตรวจสอบและป้องกันการโจมตีชนิดนี้ได้ (สำหรับเน็ตเวิร์กที่มีไฟร์วอลล์ป้องกันอยู่หากโดนโจมตีด้วย SYN Flood บางครั้งไฟร์วอลล์เองก็ไม่สามารถต้านทานได้)

2.9.3.4 Land Attack

ลักษณะการโจมตี

- หมายเลข IP ต้นทางเท่ากับ IP ปลายทาง
- หมายเลขพอร์ตต้นทางเท่ากับหมายเลขพอร์ตปลายทาง

- SYN Flag ถูก Set เสมือนขอเริ่มต้นการเชื่อมต่อ
- แพ็กเก็ตจะส่งไปยัง TCP พอร์ตที่เปิดอยู่

จะทำให้มีการตอบกลับไปมาของ TCP วนรอบอยู่ในตัวเองด้วยความเร็วสูง ทำให้คอมพิวเตอร์ต้องใช้ทรัพยากรที่มีอยู่ทั้งหมด ไม่ว่า CPU, หน่วยความจำและอื่นๆ เพื่อคอยจัดการกับ TCP ที่ตอบกลับไปมาดังกล่าวจนไม่อาจจะไปทำงานอื่น ๆ ได้อีก จนดูเหมือนว่าเครื่องคอมพิวเตอร์หยุดทำงานและไม่ตอบสนองต่อการกระตุ้นใด ๆ แม้แต่คีย์บอร์ด ดังนั้นเหลือวิธีเดียวคือต้องรีเซ็ตเครื่องหรือปิดเครื่องจึงจะสามารถหยุดการวนรอบของ TCP ได้

ดังนั้นการป้องกันการปลอม IP จะป้องกันการโจมตีประเภทนี้ได้ แต่การป้องกันการปลอม IP และเมื่อป้องกันการปลอม IP Address ไม่ได้สิ่งที่ควรทำก็คือการปรับปรุงระบบปฏิบัติการของโฮสต์ให้มีความทันสมัยที่สุดเพื่อให้ถึงแม้ว่าจะมีการโจมตีเกิดขึ้นแต่โฮสต์ที่ได้ทำการปรับปรุงก็น่าจะสามารถรับมือกับแพ็กเก็ตประเภทนี้ได้เป็นอย่างดี

2.9.3.5 Teardrop Attack

Teardrop เป็นการโจมตีโดยใช้บกพร่องของแฟรกเมนต์รีแอสเซมเบิลของ IP เพื่อทำให้ระบบปฏิบัติการปลายทางของเป้าหมายทำงานผิดพลาด ไม่อยู่ในเงื่อนไขที่กำหนดไว้และหยุดทำงาน

ในการประกอบรวมแฟรกเมนต์แพ็กเก็ตกลับมาอยู่ใน 1 ดาต้าแกรมนั้น IP มีลักษณะการทำงานโดยพิจารณาจากข้อมูลของ 3 ฟิลด์คือ

- Data length ขนาดความยาวของข้อมูลในแพ็กเก็ตนั้น
- Offset ตำแหน่งของแพ็กเก็ตที่จะนำไปประกอบรวมกลับใน IP ดาต้าแกรม
- Flag แฟลกซึ่งระบุว่าไม่มีแพ็กเก็ตต่อจากแพ็กเก็ตนี้อีก หมายถึงแพ็กเก็ตนี้เป็นส่วนสุดท้ายของดาต้าแกรม

การโจมตีของ Teardrop จะให้การเหลื่อมกันของแพ็กเก็ตในระหว่างที่มีการรวมแฟรกเมนต์แพ็กเก็ตเข้าด้วยกัน เนื่องจากในการแฟรกเมนต์ตามปกติแล้วแพ็กเก็ตจะถูกแบ่งออกเป็นส่วนย่อยแต่สามารถนำมารวมกันใหม่ได้พอดี และตำแหน่งจะถูกต้องสอดคล้องกันเสมอ แพ็กเก็ตที่ใช้ในการโจมตีของ Teardrop จะเป็นแพ็กเก็ตที่ถูกสร้างขึ้นมาโดยเฉพาะมิได้ผ่านกลไกการแฟรกเมนต์ตามปกติของ IP ในระบบปฏิบัติการนั้นเมื่อได้รับแพ็กเก็ตเข้ามา 1 แพ็กเก็ต โดยเฉพาะแพ็กเก็ตที่เป็นแฟรกเมนต์ ระบบปฏิบัติการจะต้องจัดสรรหน่วยความจำให้เพียงพอเพื่อรองรับการรวมกันแฟรกเมนต์จนครบดาต้าแกรม การรีแอสเซมเบิลก็คือการนำข้อมูลไปใส่ยังแฟรกเมนต์ไม่ควรจะเกิดขึ้นได้ไม่ว่ากรณีใด ดังนั้นเมื่อระบบปฏิบัติการเชื่อเช่นนี้จึงมิได้ตรวจสอบตำแหน่งของแฟรกเมนต์อย่างถี่ถ้วน ทำให้การนำข้อมูลจากแฟรกเมนต์ไปใส่ยังหน่วยความจำนั้นแฟรกเมนต์นั้นไปเก็บในส่วนที่จัดสรรไว้สำหรับเก็บคำสั่ง ทำให้การทำงานของระบบปฏิบัติการนั้นผิดพลาดและหยุดทำงานในที่สุด

2.9.3.6 Smurf Attack

ลักษณะการโจมตี

- ใช้ ICMP Echo Reply เป็นกลไกในการโจมตี
- ใช้ร่วมกับเทคนิคของการปลอมหมายเลย IP
- ส่ง ICMP Echo Request ไปยัง Broadcast Address

Smurf Attack เป็นการปรับปรุงเทคนิคการ Flood เน็ตเวิร์กให้ฉลาดกว่าเดิม โดยการใช้คุณสมบัติของบรอดคาสต์ โดยการส่งแพ็กเก็ตไปยัง address ของบรอดคาสต์จะทำให้ทุกโฮสต์ในเน็ตเวิร์กได้รับแพ็กเก็ตนั้นอย่างทั่วถึง ดังนั้นหากมีโฮสต์ใดที่ส่ง ICMP Echo Request มายังบรอดคาสต์ก็จะทำให้ทุก ๆ โฮสต์ทั้งหมดที่อยู่ในเน็ตเวิร์กนั้นจะได้รับ ICMP Echo และจะต้องตอบกลับด้วย ICMP Echo Reply กลับไปยังผู้ส่งเสมอ

การโจมตีนั้นทำได้โดย ในมุมมองแฮกเกอร์แล้วเปรียบเสมือนตนเองมีอาวุธพิเศษที่จะช่วยเพิ่มกำลังการโจมตี เช่นหากในเน็ตเวิร์กมีโฮสต์ 200 โฮสต์ แฮกเกอร์ส่งเพียงแพ็กเก็ตเดียวก็จะทำให้เป้าหมายถูกรุมถล่มด้วยแพ็กเก็ต 200 แพ็กเก็ตในทันทีไม่ว่าเป้าหมายจะมีความทนทานและสมรรถนะอย่างไรย่อมยากที่จะยืนหยัดกับการโจมตีแบบนี้ได้

2.9.3.7 Ping Of Death Attack

การโจมตีแบบนี้เป็นอีกวิธีหนึ่งที่ใช้ข้อบกพร่องของการแฟรกเมนต์มาเป็นช่องทางในการโจมตี ตามปกติแล้ว IP ดาต้าแกรมจะมีขนาดสูงสุดได้ไม่เกิน 65535 ไบต์ การส่ง IP ดาต้าแกรมภายในแพ็กเก็ตเดียวทำอย่างไรก็ไม่เกินนี้ การโจมตีของ Ping Of Death เห็นข้อบกพร่องในส่วนนี้และพุ่งเป้าไปยังระบบปฏิบัติการที่ไม่ได้จัดการแฟรกเมนต์อย่างรัดกุมเพียงพอ ด้วยตรรกะพื้นฐานว่าหากมีการจัดสรรหน่วยความจำไว้สูงสุด 64 K เพื่อรองรับดาต้าแกรม 1 ดาต้าแกรม ระบบปฏิบัติการไม่มีกลไกการตรวจสอบที่รอบคอบแล้ว การรีแอสเซมเบิลของแฟรกเมนต์ก็เป็นเพียงการนำข้อมูลไปใส่ในหน่วยความจำตามตำแหน่งที่ระบุในตัวแฟรกเมนต์นั่นเอง ในกรณีของ Ping Of Death สิ่งที่จะเกิดขึ้นก็คือ โปรแกรมนั้นก็จะทำงานผิดเพี้ยนไปหมด และกรณีที่เลวร้ายที่สุดคือข้อมูลส่วนเกินได้ไปทับหน่วยความจำของระบบปฏิบัติการเสียแล้ว ระบบปฏิบัติการก็ไม่อยู่ในสถานะที่จะทำงานได้อย่างถูกต้องอีกต่อไป โดยทั่วไปแล้วโฮสต์นั้นจะหยุดทำงานไปในทันที

2.9.3.8 Tribe Flood Network

Tribe Flood Network (TFN) เป็นการเปลี่ยนแนวทางการโจมตีใหม่ให้รุนแรงและซับซ้อนกว่าเดิม โดยการโจมตีเป้าหมายพร้อม ๆ กันด้วยหลาย ๆ โฮสต์ ดังนั้นแฮกเกอร์จึงเปลี่ยนวิธีโดยการใช้เทคนิค TNF โดยจุดประสงค์การโจมตีคือทำให้เน็ตเวิร์กเต็ม (Flood) แต่เปลี่ยนจากใช้เครื่องเดียวเป็นใช้หลายเครื่องในการโจมตีแทน

2.9.3.9 Diagnostic Port Attack

TCP/IP ในสมัยแรกได้มีการใส่คุณสมบัติและบริการหลายอย่างใน เพื่อใช้ในการตรวจสอบสถานะ การเชื่อมต่อ, บริการตรวจสอบเวลา, บริการเหล่านี้เรียกว่า "Small Service" มีรายละเอียดดังนี้

- Echo ใช้สำหรับตรวจสอบสถานะของเซิร์ฟเวอร์ด้วยการส่งแพ็กเก็ตไปที่พอร์ต echo หากมีข้อมูลตอบกลับมาแสดงว่าเซิร์ฟเวอร์ยังทำงานอยู่ เป็นเสมือนการ Ping แต่จะใช้ TCP หรือ UDP แทนที่จะใช้ ICMP
- Discard ใช้สำหรับการทดสอบบางชนิด โดยทั่วไป TCP หรือ UDP หากไม่ทำการเปิดพอร์ตให้บริการจะมีการส่ง RST สำหรับ TCP และ ICMP Port Unreachable กลับไปยังไคลเอนต์ แต่สำหรับของ discard นี้จะไม่มี Error Message ดังกล่าวแต่เซิร์ฟเวอร์ก็ไม่ตอบรับใด ๆ
- Daytime, Time ใช้สำหรับสอบเวลาของเซิร์ฟเวอร์ เพียงแต่ข้อมูลจะอยู่คนละรูปแบบนั่นเอง
- Chargen ใช้ในการตรวจสอบการเชื่อมต่อเหมือนกัน โดยเฉพาะกรณีที่ต้องการ monitor เซิร์ฟเวอร์ตลอดเวลา chargen จะมีประโยชน์เพราะตราบใดที่ยังมีข้อมูลมาจากเซิร์ฟเวอร์เซ็ตว่าเซิร์ฟเวอร์ยังคงทำงานตามปกติ

การโจมตีลักษณะนี้สามารถทำได้โดย

- ส่งแพ็กเก็ตจากพอร์ต echo ไปยัง พอร์ต chargen โดยที่ต้นทางและปลายทางเป็นโฮสต์เป้าหมาย
- เมื่อบริการ chargen ได้รับแพ็กเก็ตใดก็จะเริ่มทำงานโดยการส่ง ASCII character กลับไปยังผู้ที่ส่งแพ็กเก็ตเข้ามาตามหมายเลขพอร์ตที่ระบุไว้ ซึ่งก็คือตอบกลับมายังเครื่องตนเองที่พอร์ต echo นั้นเอง
- เมื่อแพ็กเก็ตจาก chargen มาถึงพอร์ต echo บริการ echo ก็จะทำงานโดยการตอบข้อมูลที่ได้รับกลับมาไปยังพอร์ต chargen อีก
- การทำงานของ chargen จะกลับไปยังขั้นตอนที่ 2 และวนเวียนเช่นนี้ไปไม่รู้จบที่โฮสต์เองก็ส่งรับข้อมูลกันระหว่างพอร์ต echo กับ chargen จนในที่สุดก็จะหยุดทำงาน

จะเห็นว่าเป็นกระบวนการทำลายตนเองที่รวดเร็ว และมีผลรุนแรงมาก

2.9.3.10 UDP Bomb

โดยปกติแล้ว UDP จะมีส่วน UDP Header 8 ไบต์ แล้วส่วนข้อมูลอีกต่างหาก การโจมตีก็ทำโดยการส่ง UDP Length เป็น 7 ซึ่งน้อยกว่าปกติ ถ้าระบบปฏิบัติการไม่คำนึงถึงเรื่องนี้ก็อาจทำให้ทำงานผิดพลาด แล้วการส่ง UDP ดาต้าแกรมมากถึง 4,294,967,295 ไบต์ ก็อาจทำให้เครื่องหยุดทำงาน

2.9.3.11 ICMP Source Quench Attack

ICMP Source Quench เป็นการควบคุมความเร็วในการรับส่งข้อมูลทั้งสองฝั่งให้มีขนาดเท่ากัน ทำให้แฮกเกอร์ใช้แพ็กเก็ตนี้ส่งไปยังเป้าหมายโดยแจ้งให้โฮสต์เป้าหมายทำการเราต์แพ็กเก็ตทั้งหมดไปที่แอดเดรสของลูปแบค ก็คือการส่งแพ็กเก็ตทั้งหมดวนกลับเข้าหาตัวเองพร้อมทั้งแจ้งให้โฮสต์ลดความเร็วของการส่งข้อมูลไปยังลูปแบคด้วย เมื่อเป้าหมายถูกโจมตีก็จะทำให้กระบวนการรับส่งข้อมูลของโฮสต์เป้าหมายทำงานผิดปกติและหยุดทำงานได้

2.9.3.12 Winfreeze

Winfreeze ใช้ ICMP redirect Message เป็นกลไกในการโจมตี ด้วยการส่ง ICMP redirect ไปยังเซิร์ฟเวอร์เป้าหมายเป็นจำนวนมากอย่างต่อเนื่องเมื่อเซิร์ฟเวอร์เป้าหมายได้รับ ICMP ดังกล่าวก็จะพยายามนำข้อมูลที่ส่งมาด้วยเข้าไปปรับปรุงในตารางเราต์ติ้งเทเบิลของตนเอง แต่เนื่องจากปริมาณของ ICMP ที่เข้ามามากและเร็วเกินกว่าที่จะนำไปเพิ่มใน routing table ได้ทันทำให้การปรับปรุง routing table ผิดพลาดได้

2.9.3.15 Jolt

Jolt เป็นการโจมตีโดยอาศัยแฟรกเมนต์ทั่วไป แฮกเกอร์จะอาศัยการส่งแฟรกเมนต์แพ็กเก็ตซ้ำ ๆ จำนวนมากอย่างต่อเนื่องมายังเซิร์ฟเวอร์เป้าหมาย หากข้อมูลมีความเร็วและปริมาณไม่มากพอก็จะเพียงแต่ทำงานช้าลงกว่าปกติ หากแฮกเกอร์สามารถเพิ่มความเร็วในการส่งข้อมูลเครื่องก็จะหยุดทำงานไม่ตอบสนองต่อผู้ใช้อีกเลย หากสังเกตโดยการทำการตรวจสอบค่า CPU Utilization จะพบว่าในขณะที่เครื่องถูกโจมตีนั้น CPU Utilization จะขึ้นสูงถึง 100% และถูกใช้ไปโดยเคอร์เนลของระบบ ปฏิบัติการ นั่นเป็นเหตุผลที่ทำให้ไม่มีการตอบสนองใด ๆ ต่อผู้ใช้ เนื่องจากกำลังของ CPU ถูกใช้ไปในการจัดการกับ Fragment หมดนั่นเอง เป็นวิธีการโจมตีสำหรับโจมตี Window เป็นส่วนใหญ่

บทที่ 3
วิธีการดำเนินโครงงาน

3.1 อุปกรณ์และเครื่องมือในการพัฒนา

3.1.1 อุปกรณ์ Hardware ที่ใช้ในการพัฒนา

3.2.1.1 เครื่องคอมพิวเตอร์จำนวน 3 เครื่อง

3.2.1.2 อุปกรณ์ระบบเน็ตเวิร์ค

3.1.2 อุปกรณ์ Software ที่ใช้ในการพัฒนา

3.2.2.1 โปรแกรม Snort ที่เป็นแนวทางในการพัฒนา โดยใช้เวอร์ชัน 0.99

3.2.2.2 ระบบปฏิบัติการ Microsoft Windows XP Service Pack 2

3.2.2.3 ระบบปฏิบัติการ Linux Fedora core 4

3.2.2.4 โปรแกรม VMware เวอร์ชัน 5.0.0 build 13124

3.2.2.5 โปรแกรม VI 6.3

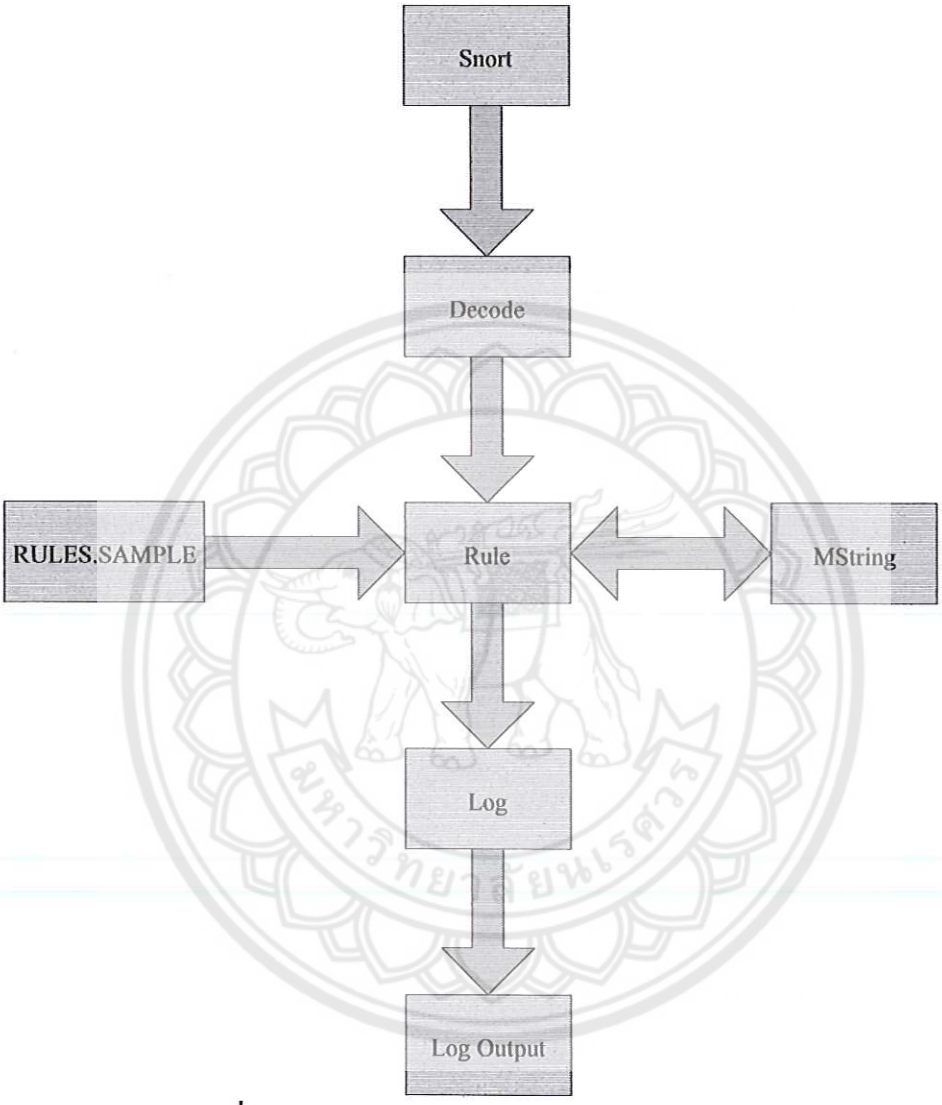
3.2.2.6 โปรแกรม Microsoft Visual Studio 6

3.2.2.7 โปรแกรม Panther

3.2.2.8 โปรแกรม wine เวอร์ชัน 0.9.2

3.2 หลักการออกแบบโปรแกรมการตรวจจับการบุกรุก

ในการตรวจสอบการบุกรุกนั้น ได้ออกแบบโปรแกรมดังแผนผังการทำงานโดยรวมของโปรแกรม (Flowchart) ดังรูปที่ 3.1

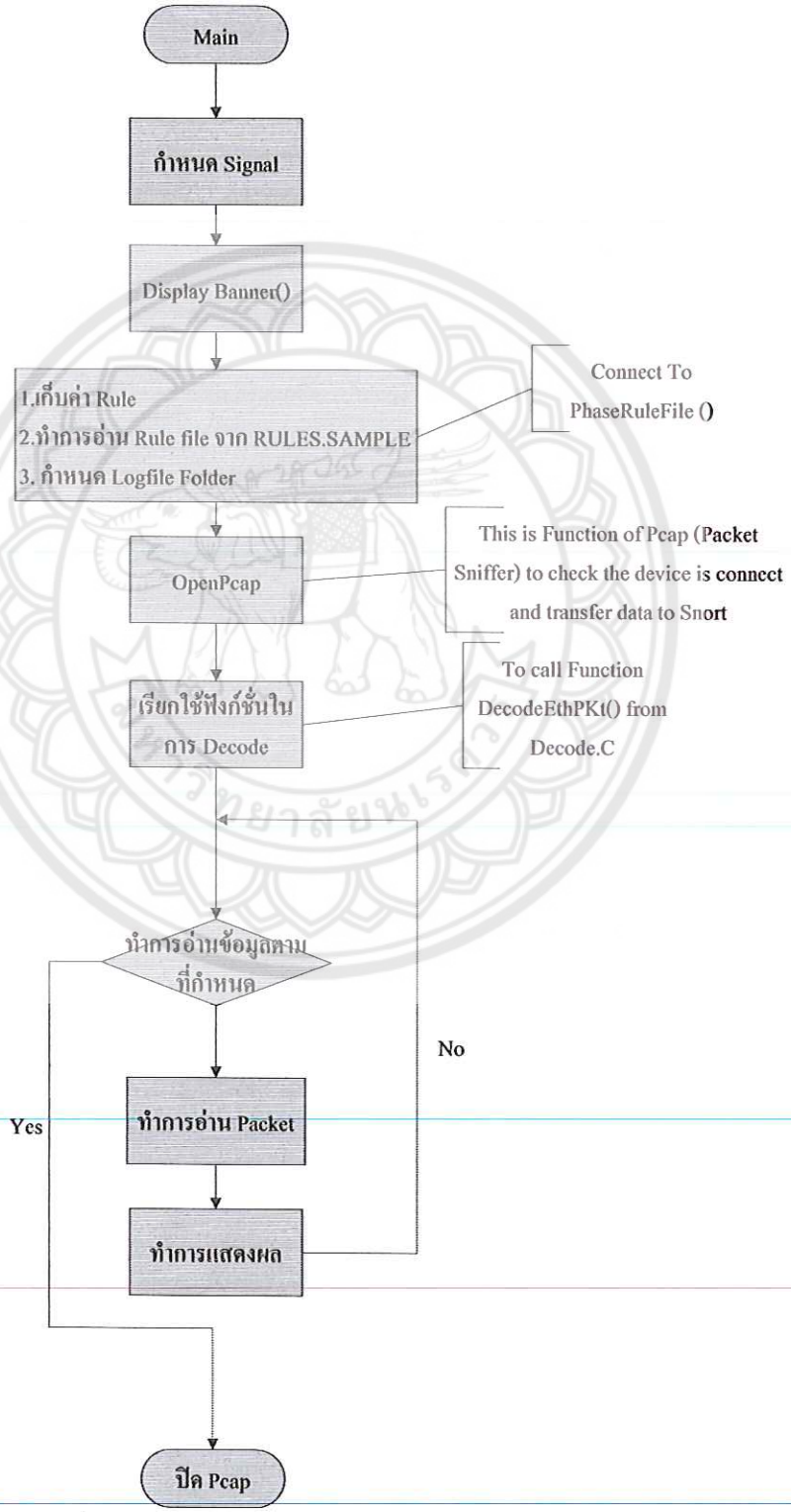


รูปที่ 3.1 แผนผังการทำงานโดยรวมของโปรแกรม

จากรูปที่ 3.1 เป็นรูปการทำงานโดยรวมของโปรแกรมโดยที่ Snort ทำการเริ่มเก็บข้อมูลและทำการติดต่อกับตัวโปรแกรม Pcap เพื่อทำการติดต่อขอรับข้อมูลจากระบบ ที่เป็นข้อมูลที่เป็นแพ็กเก็ตต่าง ๆ ที่ผ่านระบบเครือข่าย แล้วทำการส่งข้อมูลให้กับทางด้าน Decode เพื่อทำการประมวลข้อมูลว่าเป็นข้อมูลที่ได้เป็นประเภทใด (TCP, UDP หรือ ICMP) เพื่อให้ทางฟังก์ชัน Rule นั้น ทำการประมวลตรวจสอบว่าข้อมูลที่เข้ามานั้นเป็นไปตามกฎที่กำหนดไว้ว่าเป็นการบุกรุกหรือไม่ โดยที่ Rule นั้นจะไปรับกฎทั้งหมดที่สร้างขึ้นที่ RULES.SAMPLE แล้วนำกฎทั้งหมดมาประมวลด้วยตัว MString ที่เป็นฟังก์ชันในการนำกฎมาแปลงให้อยู่ในรูปแบบที่ Rule ได้กำหนดไว้

แล้วส่งไปให้ในส่วนของ Log เพื่อทำการแยกประเภทข้อมูลเพื่อทำการเขียนบันทึกไฟล์ตามประเภทที่กำหนดไว้ (Logfile, Alertfile) และทำการแสดงผลออกมา โดยสามารถแสดงการทำงานโดยละเอียดดังนี้

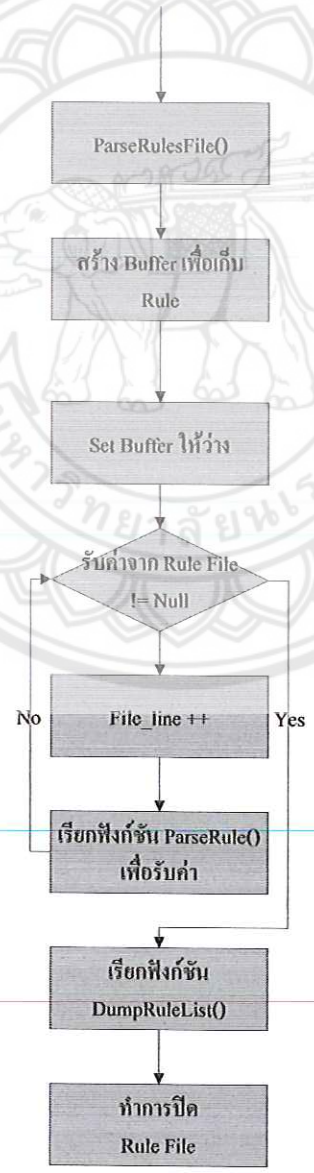
3.2.1 ในส่วนของโปรแกรมหลัก



รูปที่ 3.2 รูปของการทำงานหลักของโปรแกรม

จากรูปที่ 3.2 เป็นรูปการทำงานของโปรแกรมหลักที่อยู่ใน Snort โดยเริ่มจากการกำหนดค่า Signal ต่อมาก็เรียกฟังก์ชันที่ทำหน้าที่ในการแสดงผลที่เก็บข้อมูลของคณะผู้จัดทำและรายละเอียดของเวอร์ชัน จากนั้นเริ่มเก็บค่าต่าง ๆ และทำการสั่งให้เริ่มทำการเก็บ Rule ไฟล์มาไว้ใน Buffer และกำหนดที่เก็บของ Logfile เมื่อทำการเสร็จแล้วก็สั่งให้ไปติดต่อกับ Pcap ซึ่งเป็นตัวที่ทำงานในการรับข้อมูลแพ็กเก็ตต่าง ๆ จากระบบปฏิบัติการเพื่อขอข้อมูลนำมาประมวลผล แล้วไปเรียกฟังก์ชัน Decode เพื่อนำข้อมูลไปประมวลผลต่อ แล้วทำการรับส่งไปจนกว่าจะมีคำสั่งให้หยุดหรือ ตามจำนวนข้อมูลที่ได้กำหนดไว้แล้ว แล้วทำการปิดการติดต่อกับ Pcap แล้วทำการปิดโปรแกรมหลัก

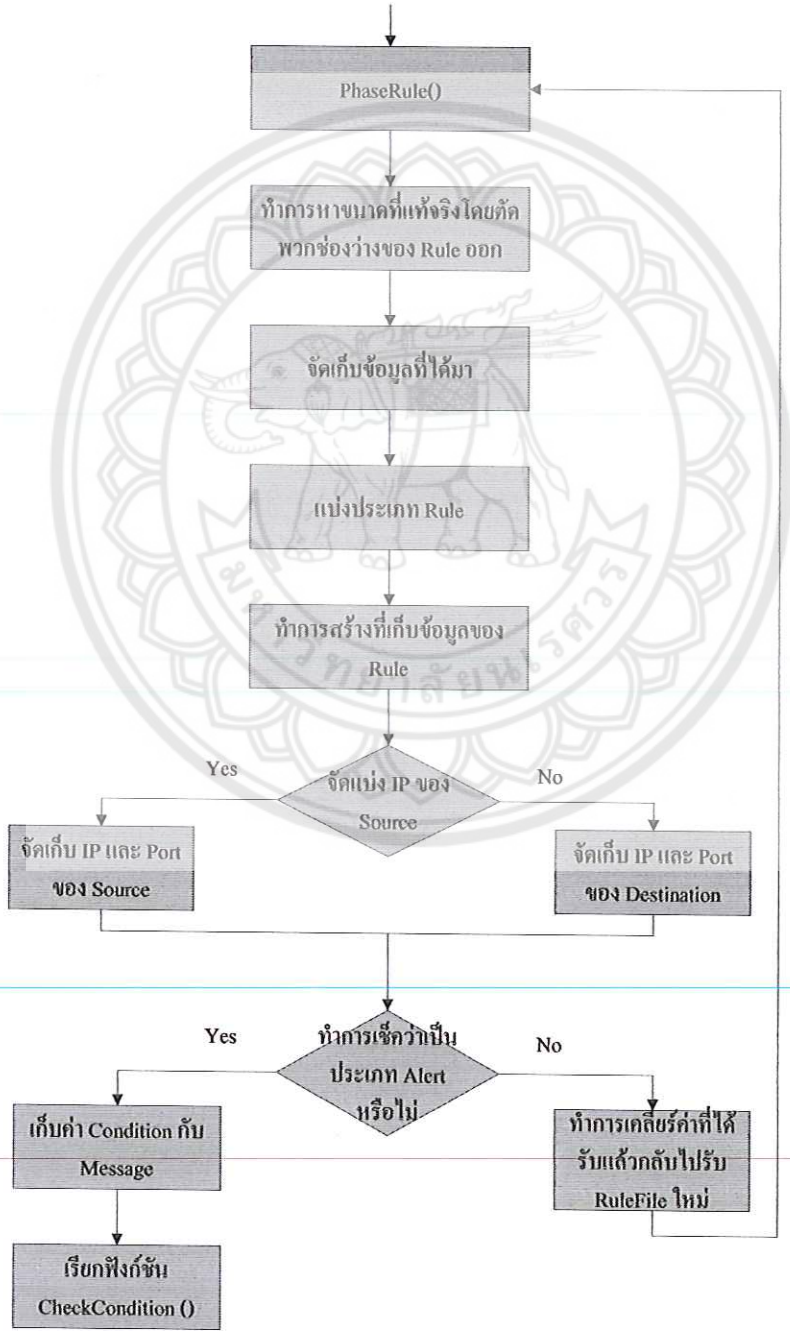
3.2.2 หลักการทำงานของการเรียกข้อมูลใน RULES.SAMPLE เข้ามาเก็บ



รูปที่ 3.3 การทำงานของการเรียกข้อมูลของกฎใน RULES.SAMPLE

จากรูปที่ 3.3 นั้นเริ่มต้นด้วยการกำหนดให้ทำการสร้าง Buffer เพื่อมาลองรับข้อมูลที่จะนำเข้ามาจาก RULES.SAMPLE แล้วทำการสั่งให้กำหนดค่า Buffer ให้เป็นค่าเท่ากับ 0 เพื่อรอรับข้อมูล จากนั้นทำการเปิดไฟล์แล้วเริ่มตรวจว่ามีข้อมูลใน RULES.SAMPLE และทำการรับข้อมูลเข้ามาเก็บไว้ใน Buffer จนครบเรียบร้อยแล้วทำการเรียกฟังก์ชัน DumpRuleList () เพื่อดึงข้อมูลมาเพื่อตรวจสอบหาข้อผิดพลาด เมื่อตรวจพบจะทำการแสดง Error message แจ้งเตือน จากนั้นทำการปิดไฟล์

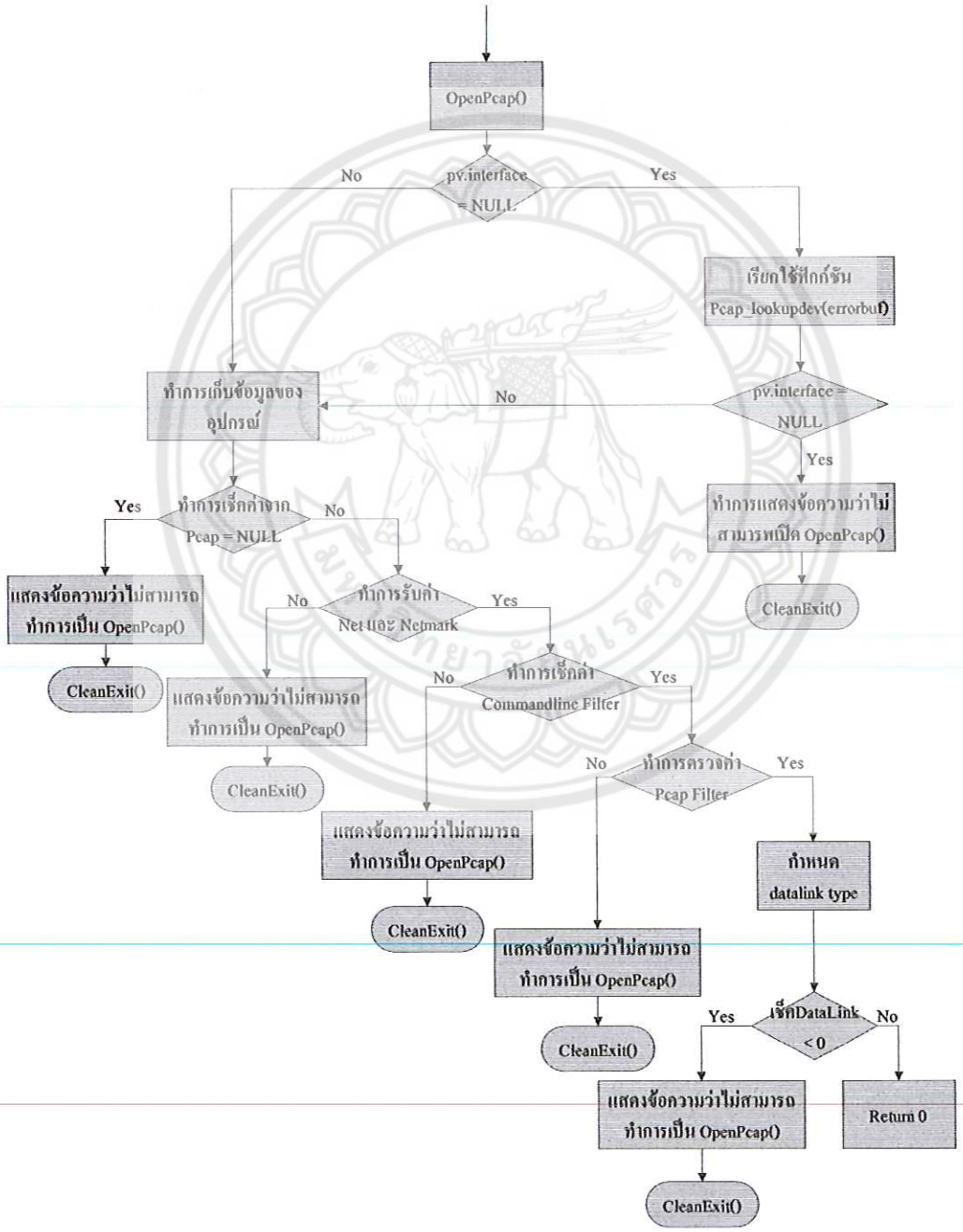
3.2.3 การเก็บข้อมูลที่รับมา PhaseRuleFile () เข้ามาเพื่อเก็บไว้ทำการตรวจสอบ



รูปที่ 3.4 การเก็บข้อมูลที่รับมา PhaseRuleFile () เข้ามาเพื่อเก็บไว้ทำการตรวจสอบ

จากรูปที่ 3.4 เริ่มจากรับค่ามาจาก PhaseRuleFile () แล้วทำการหาขนาดที่แท้จริงโดยที่ตัดค่าต่าง ๆ เช่น ช่องว่าง แล้วจัดเก็บข้อมูลที่ได้ไว้ใน Buffer จัดการแบ่งประเภทกฎโดยจัดแบ่งค่า IP และ Port ของต้นทาง หรือ IP และ Port ของต้นปลายทาง แล้วทำการเช็คว่ากฎที่รับเข้ามานั้นเป็นกฎที่เป็นข้อมูลธรรมดาจะทำการจัดเก็บเป็น Logfile หรือเป็นการบุกรุกถ้าเป็นการบุกรุกก็จะเก็บค่าตัวแปรต่าง ๆ แล้วทำการส่งไปให้กับฟังก์ชัน CheckCondition () เพื่อทำการเลือกว่าเป็นการบุกรุกประเภทใด

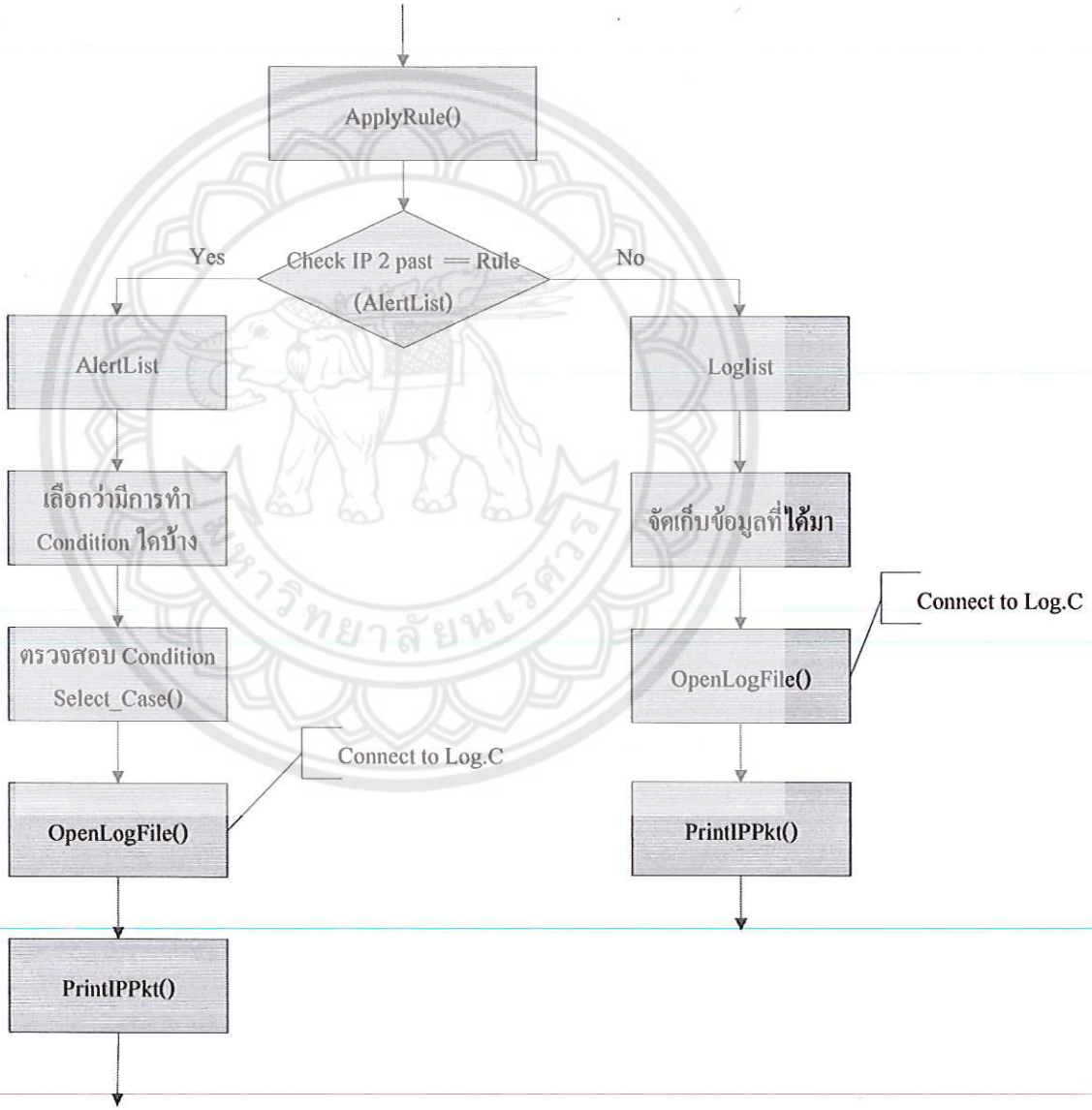
3.2.4 การติดต่อรับค่าของข้อมูลจาก Pcap



รูปที่ 3.5 รูปของการติดต่อเพื่อรับข้อมูลกับ Pcap

จากรูปที่ 3.6 เมื่อได้รับข้อมูลที่ได้จาก Pcap เรียบร้อยแล้วจะทำการเรียกฟังก์ชัน DecodeEthPkt () เพื่อทำการถอดค่าของแพ็กเก็ตที่ได้รับ แล้วทำการส่งไปให้ฟังก์ชัน DecodeIP () เพื่อทำการถอดรหัสเพื่อหาค่า IP Network Layer แล้วทำการแยกประเภทเพื่อทำการถอดรหัสโดยแบ่งตามที่ศึกษาคือ TCP, UDP และ ICMP และทำการส่งไปให้ ฟังก์ชัน ApplyRule () เพื่อทำการตรวจเช็คว่าเป็นข้อมูลที่ตรงกับกฎแบบใดที่กำหนดไว้ แต่ถ้าไม่เป็นไปตาม Protocol ที่กำหนดก็จะทำการละทิ้งแพ็กเก็ตนั้นแล้วทำการรับค่าของแพ็กเก็ตใหม่

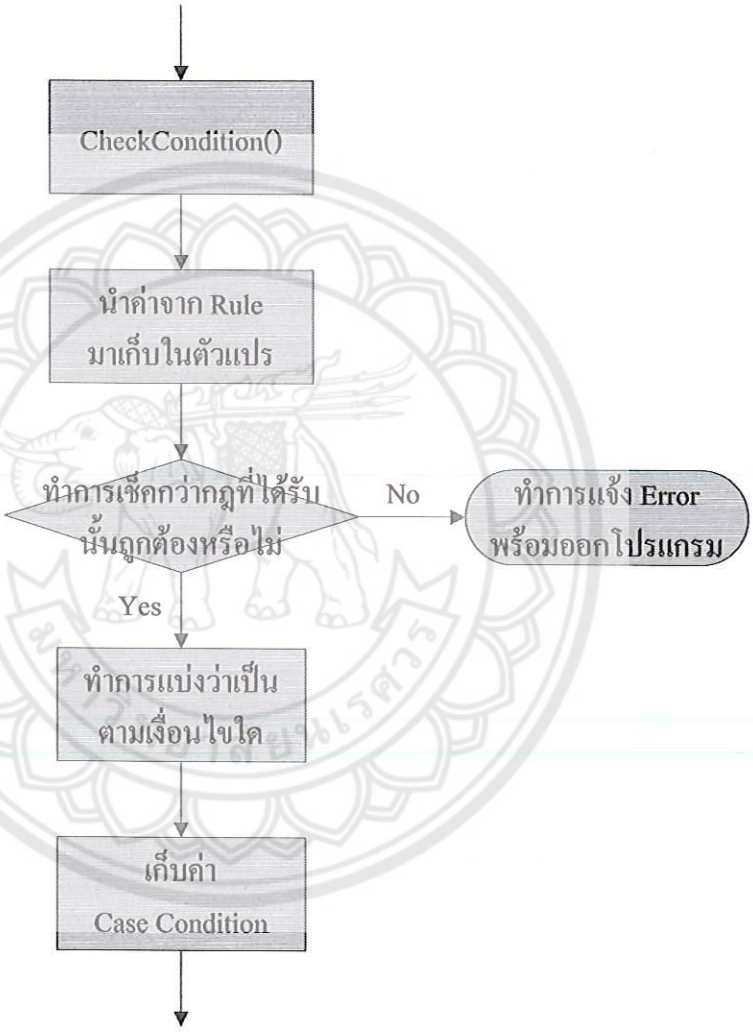
3.2.6 การตรวจสอบว่าข้อมูลที่ได้รับเป็นไปตามกฎใด



รูปที่ 3.7 การตรวจสอบว่าข้อมูลที่ได้รับเป็นไปตามกฎใด

จากรูปที่ 3.7 เป็นการทำงานของการตรวจสอบว่าข้อมูลที่ได้รับนั้นเป็นไปตามกฎที่กำหนดไว้ว่าเป็นการบุกรุกหรือว่าเป็นข้อมูลธรรมดา ถ้าเป็นการบุกรุกแล้วก็ทำการตรวจสอบว่าตรงกับเงื่อนไขใดบ้างแล้วทำการเก็บค่าแล้วไปฟังก์ชัน OpenLogFile () และ PrintIPPKT () เพื่อทำการจัดเก็บเป็น Logfile ไว้ในที่ที่กำหนดเพื่อไว้ใช้ในการตรวจสอบข้อมูลในการบุกรุก หรือว่าเป็นข้อมูลธรรมดา เพื่อจะได้เพิ่มหรือลดกฎในการป้องกัน

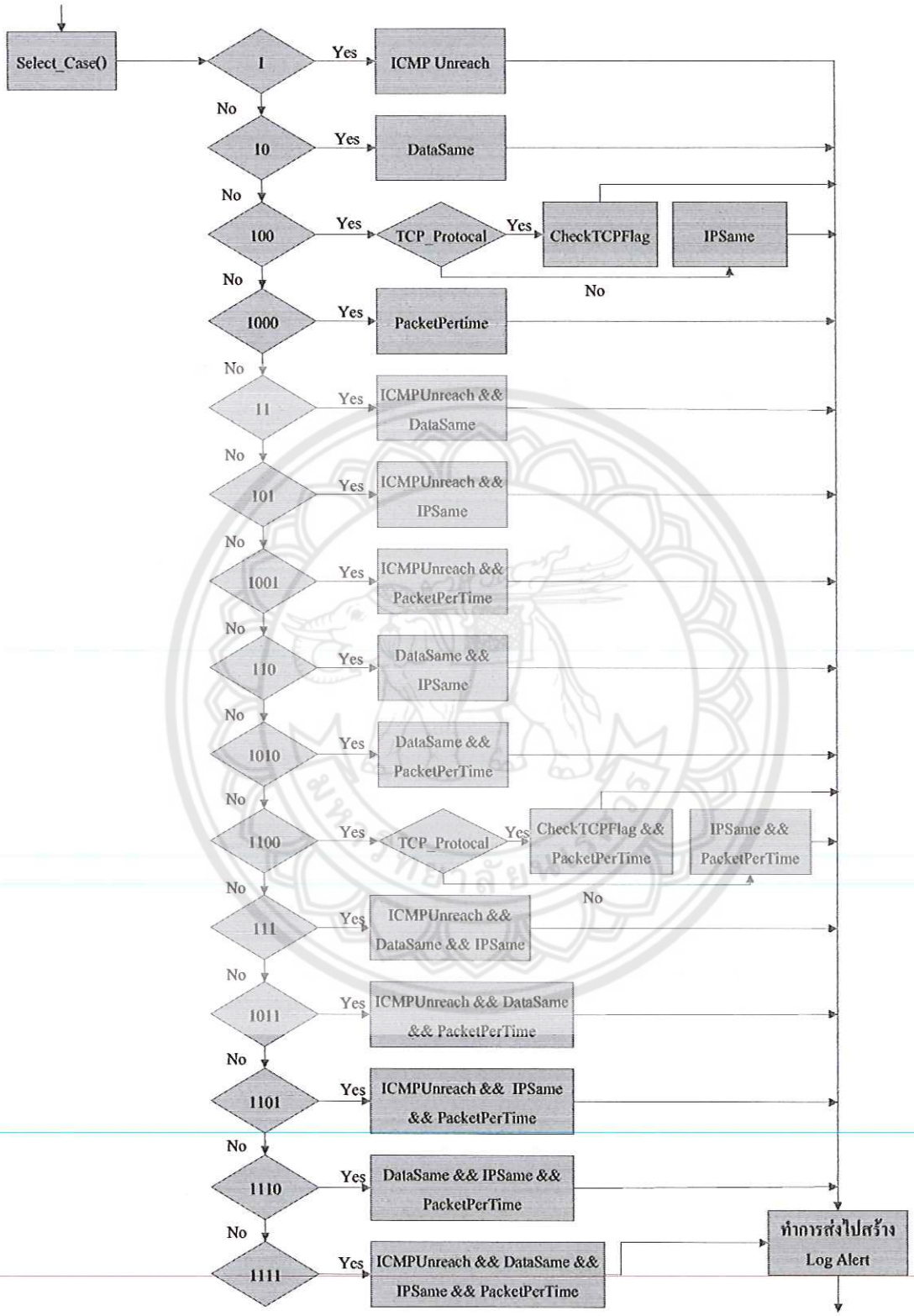
3.2.7 การตรวจสอบว่า Rulefile ที่ได้รับมานั้นมีค่าถูกต้องตามกฎหรือไม่



รูปที่ 3.8 การตรวจสอบว่า Rulefile ที่ได้รับมานั้นมีค่าถูกต้องตามกฎหรือไม่

รูปที่ 3.8 เป็นการนำค่าของกฎที่ได้มานั้นมาเก็บในตัวแปรแล้วทำการตรวจสอบว่าที่ได้รับมาเป็นไปตามกฎและเงื่อนไขที่กำหนดหรือไม่ ถ้าเป็นไปตามที่กำหนด จะทำการแบ่งว่าเป็นตามเงื่อนไขใด และเก็บค่าของเงื่อนไข เพื่อนำข้อมูลที่แยกแล้วมาทำการตรวจสอบว่าเป็นไปตามเงื่อนการบุกรุกประเภทใด แต่ถ้าไม่เป็นไปตามที่กำหนดนั้นจะทำการปิดการติดต่อ แล้วทำการแจ้งเตือนว่ากฎที่ได้รับนั้นมีการผิดพลาดแล้วทำการปิดโปรแกรม

3.2.8 การนำข้อมูลที่น่าสงสัยมาทำการตรวจสอบว่าเป็นการบุกรุกประเภทใด

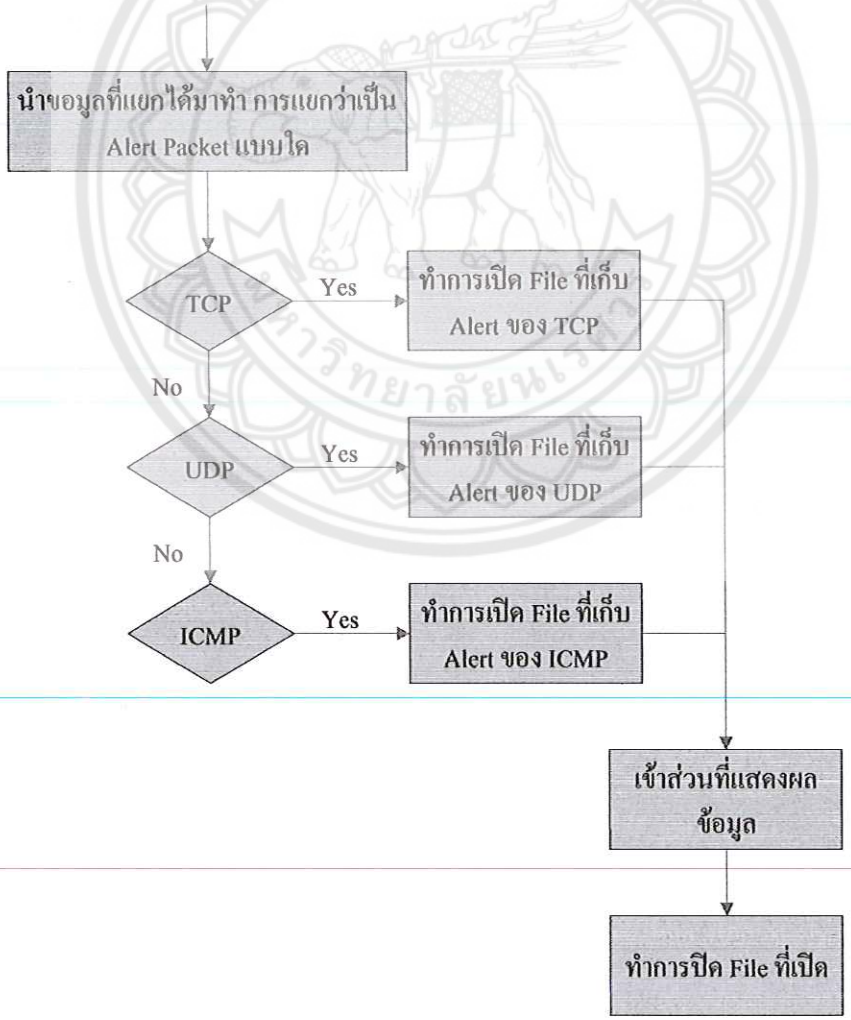


รูปที่ 3.9 การนำข้อมูลที่สงสัยมาทำการเลือกว่าเป็นการบุกรุกแบบใด

จากรูปที่ 3.9 เป็นการตรวจสอบว่าข้อมูลที่น่าสงสัยที่ได้รับจาก CheckCodition มาทำการตรวจสอบว่าเป็นการบุกรุกประเภทใดโดยที่จะตั้งเงื่อนไขไว้ว่า

- ICMP UnReach เป็นการตรวจสอบว่า ICMP ที่ส่งมีการผิดพลาดหรือถูกเปลี่ยนแปลง
- DataSame เป็นการตรวจสอบว่ามีข้อมูลข้างในนั้นเป็นข้อมูลชนิดเดียวกัน
- ถ้าเป็น TCP Protocol สามารถแบ่งได้
    - CheckTCPFlag เป็นการตรวจสอบว่าของ TCPFlag
- PacketPerTime เป็นการตรวจสอบเวลาแต่ละการส่งข้อมูล
- IPSame เป็นการตรวจสอบว่าค่า IP Address ที่ส่งมาติดต่อนั้นเป็น IP ใด

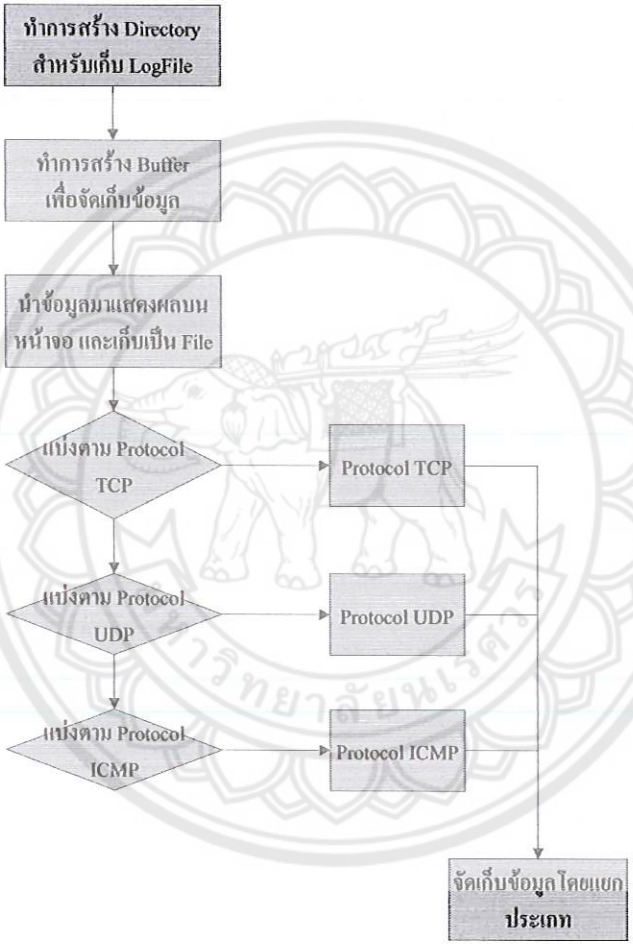
3.2.9 การจัดเก็บ Logfile ที่เป็นการเตือนภัย



รูปที่ 3.10 การจัดเก็บ Logfile ที่เป็นการเตือนภัย

รูปที่ 3.10 เริ่มจากข้อมูลที่นำมาเลือกว่าเป็นการบุกรุกประเภทใดแล้ว นำข้อมูลนั้นมาทำการบันทึกเป็น Logfile เพื่อทำการบันทึกเป็นไฟล์โดยทำการแบ่งเป็นประเภทข้อมูลตามโพรโตคอล ที่กำหนดไว้แล้วทำการเปิดไฟล์ที่เป็นการแจ้งเตือนแล้วทำการบันทึกเพื่อให้ผู้ควบคุมรู้ตามชนิดของการโจมตีตามโปรโตคอลนั้น ๆ แล้วทำการแสดงผลลัพธ์ออกมา เมื่อดำเนินการเรียบร้อยแล้วทำการปิดไฟล์การเตือนนั้น

3.2.10 การจัดเก็บ Logfile ที่เป็นข้อมูลธรรมดา



รูปที่ 3.11 การจัดเก็บ Logfile ที่เป็นข้อมูลธรรมดา

จากรูปที่ 3.11 เป็นการทำการจัดเก็บข้อมูลที่ได้รับมาที่เป็นข้อมูลธรรมดา ไม่ใช้การบุกรุกเพื่อทำการวิเคราะห์ เพราะการบุกรุกนั้นมีการปรับเปลี่ยนกลยุทธอยู่ตลอดเวลา ดังนั้นจึงมีส่วนนี้เพื่อเก็บข้อมูลที่ได้รับมาที่ไม่ได้เป็นการแจ้งเตือน เริ่มจากการที่ทำการสร้าง Directory สำหรับเก็บ Logfile แล้วทำการสร้าง Buffer ในการเก็บข้อมูล เมื่อมีข้อมูลเข้ามาแล้วจัดการแสดงผลแล้วทำการจัดเก็บโดยแบ่งตามประเภทเช่นเดียวกับการแจ้งเตือน

บทที่ 4
ผลการทดลองและการวิเคราะห์

4.1 จัดเตรียมก่อนการทดลอง

โปรแกรมดักจับ Packet ถูกสร้างขึ้นโดยการใช้ Libpcap ดังนั้นก่อนทำการใช้งานโปรแกรมจำเป็นต้องทำการลง Libpcap ก่อน แล้วโปรแกรมจะสามารถทำงานได้ ซึ่งสามารถหา Download ได้ที่ www.tcpdump.org ในขั้นตอนการลง Libpcap นั้นให้ทำการเข้าไปใน Folder ของ Libpcap จากนั้นใช้คำสั่ง

- ./configure
- make

จากนั้นจึงสามารถใช้งานโปรแกรมดักจับ Packet ได้ แต่เมื่อนำโปรแกรมมาใช้งานในครั้งแรก จำเป็นต้องมีการ Configure เพื่อทำการตวจสอบและทำการปรับให้สามารถทำงานกับระบบปฏิบัติการได้ และทำการ Make โปรแกรมขึ้นมาก่อนจึงสามารถทำงานได้

เมื่อใช้งานสองคำสั่งจะปรากฏ File Snort ขึ้นมาและพร้อมนำมาใช้งานได้

4.2 ขั้นตอนการทดลอง

4.2.1 การกำหนด RULE ในการใช้งาน

RULE เป็นการกำหนดการทำงานของโปรแกรม เพื่อให้สามารถจับ Packet ที่ต้องการได้ ค่าของ RULE ต่างๆ อยู่ในไฟล์ RULES.SAMPLE ในการกำหนด RULE ของโปรแกรมนี้ มีอยู่สองวิธี คือ การ Log และ Alert โดยการทำงานของทั้งสองนั้นจะแตกต่างกันดังนี้

- Log เป็นการดักจับ Packet ใน Protocol ทุก Packet ที่อยู่ในระบบ Network
- Alert เป็นการดักจับ Packet ที่อยู่ใน Protocol โดยจับเฉพาะ Packet ที่อยู่ในเงื่อนไขที่กำหนดเอา โดยสามารถกำหนดเงื่อนไขได้ด้วยการกำหนดค่าได้

4.2.1.1 การกำหนด RULE LOG

RULE LOG มี format คือ

log protocol IP_S PORT_S -> IP_D PORT_D

ความหมายของตัวแปรในแต่ละค่ามีดังนี้

- Protocol โดยมีแบ่งได้ 3 Protocol คือ TCP, UDP หรือ ICMP
- IP_S หมายถึง ค่า IP ที่ทำการส่ง packet ถ้ากำหนดค่าด้วย 192.168.1.0/24 หมายถึงการตรวจสอบเฉพาะ IP Address ของภายในระบบ Network เท่านั้น

ไม่สามารถตรวจสอบภายนอกได้ แต่ถ้าใช้ any จะสามารถตรวจสอบ IP Address ภายนอกที่เข้าในระบบ Network ได้

- PORT_S หมายถึงค่า port ของ IP ต้นทางที่ทำการส่ง สามารถกำหนดเป็น port ที่ต้องการได้ หรือถ้าต้องการตรวจสอบ port ทั้งหมดให้ใส่ค่าเป็น any (แต่ในส่วนของ ICMP ส่วนมากเป็นการตรวจสอบสถานะเครื่องเป้าหมาย ดังนั้นในส่วนของ ICMP ค่าของ Port จึงเป็น any)
- IP_D หมายถึง ค่า IP ปลายทางที่ทำการรับ Packet ถ้ากำหนดค่าด้วย 192.168.1.0/24 หมายถึงการตรวจสอบเฉพาะ IP Address ของภายในระบบ Network เท่านั้น ไม่สามารถตรวจสอบภายนอกได้ แต่ถ้าใช้ any จะสามารถ ตรวจสอบ IP Address ภายนอกที่เข้าในระบบ Network ได้
- PORT_D หมายถึง ค่า Port ของ IP ปลายทางที่ทำการรับ Packet สามารถ กำหนดค่าได้ หรือถ้าต้องการตรวจสอบ Port ทั้งหมดให้ใส่ค่าเป็น any (แต่ในส่วนของ ICMP ส่วนมากเป็นการตรวจสอบสถานะของเครื่องเป้าหมายดังนั้น ในส่วนของ ICMP ค่าของ Port จึงเป็น any)

ตัวอย่างเช่นถ้าต้องการตรวจสอบการส่ง Packet TCP ที่ Port 80 ให้เขียนเป็น

- log tcp any 80 -> any any

หรือถ้าต้องการตรวจสอบการรับ packet ICMP ที่ Port 5500 ให้เขียนเป็น

- log icmp any any -> any 5500

4.2.1.2 การกำหนด RULE ALERT

RULE ALERT มี format คือ

alert protocol IP_S PORT_S -> IP_D PORT_D [condition] <message>

ส่วนของค่า Protocol, IP_S, PORT_S, IP_D และ PORT_D มีค่าเหมือนกับ RULE LOG ส่วนในค่า Message คือข้อความเพื่อบอกว่าเป็น Alert ชนิดใด ในส่วนของ Condition เป็นการกำหนดเงื่อนไขในการจับ Packet ซึ่งในแต่ละ Protocol จะไม่เหมือนกันดังนี้

4.2.1.2.1 PROTOCOL ICMP

มี format : [p0000/000:I:D:U]

- p0000/000 = จำนวน Packet ที่กำหนด / เวลาที่กำหนด (n/t) หมายถึงการจับ Packet เมื่อมีการรับส่ง Packet จำนวน n ในเวลา t เช่น กำหนดค่า p0005/001 จะทำการจับ Packet เมื่อมีการรับส่ง packet ในจำนวน 5 packet ต่อ 1 วินาที และเมื่อการกำหนดค่าเป็น p0050/000 หมายถึงจับการรับส่ง packet จำนวน 50 packet ตั้งแต่เริ่มการใช้งานโปรแกรม แต่ถ้าเขียนเป็น p0000/020 นั้นหมายถึงการจับ Packet ในเวลา 20 วินาที

นับตั้งแต่เริ่มโปรแกรม จนถึงการจับ Packet ตัวสุดท้าย ถ้าไม่ต้องการใช้การตรวจสอบด้วยวิธีนี้ให้ใส่ข้อความเป็น p0000/000

- I = ทำการจับ Packet เมื่อ IP_S (IP ส่ง) กับ IP_D (IP รับ) เหมือนกับ Packet ที่ได้ทำการส่งมาก่อนหน้านี้ ถ้าไม่ต้องการตรวจสอบด้วยวิธีดังกล่าว ให้ใส่ค่า 0
- D = ทำการจับ packet เมื่อข้อมูลใน Packet ที่ตรวจได้เหมือนกับข้อมูลของ Packet ก่อนหน้า ถ้าไม่ต้องการตรวจสอบด้วยวิธีนี้ให้ทำการใส่ 0
- U = ทำการจับ Packet เมื่อ Message type ใน ICMP เป็นชนิด Error ถ้าไม่ต้องการตรวจสอบด้วยวิธีนี้ให้ใส่ค่า 0

4.2.1.2.2 PROTOCOL UDP

มี Format : [p0000/000:I:D] ค่า p0000/000, I, D ค่าในส่วนนี้จะเหมือนกับของ PROTOCOL ICMP

4.2.1.2.3 PROTOCOL TCP

มี format : [p0000/000:F&F]

- p0000/000 มีค่าเหมือน PROTOCOL ICMP
- F&F ตรวจจับเฉพาะ Packet ที่มี TCP flags ที่กำหนด เมื่อทำการใส่

  - F&0 หรือ 0&F หมายถึงตรวจจับ Packet ที่มีแต่ Flags TCP ที่กำหนดเพียง Flags เพียง Flags เดียวถ้ามี Flags อื่นด้วยจะไม่ทำการตรวจจับ
  - F&N หรือ N&F หมายถึงตรวจจับ Packet ทุก Packet ที่มี Flag TCP ที่กำหนด ทุก Packet โดยที่ใน Packet นั้นอาจมี Flags อื่นด้วย
  - N&0, 0&N หรือ N&N เป็นการตรวจจับ Packet ที่ไม่ Flags อยู่แม้แต่ Flags เดียว
  - F&F หมายถึงการตรวจจับ Packet ที่มี Flags 2 Flags ที่กำหนดเกิดขึ้นพร้อมกัน
  - 0&0 หมายถึงไม่ทำการตรวจจับ Packet ด้วยวิธีนี้

การกำหนดค่า flags

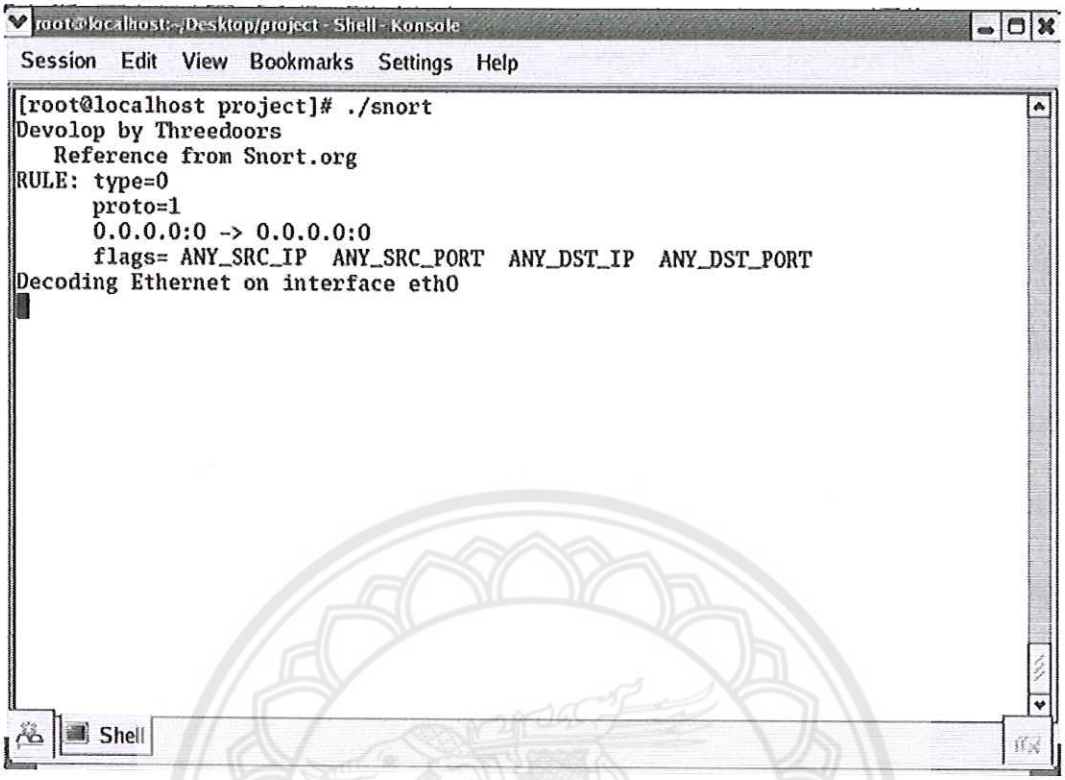
- A หรือ a : ตรวจจับ Packet ที่มี flag ACK
- S หรือ s : ตรวจจับ Packet ที่มี flag SYN
- F หรือ f : ตรวจจับ Packet ที่มี flag FIN
- U หรือ u : ตรวจจับ Packet ที่มี flag URG
- P หรือ p : ตรวจจับ Packet ที่มี flag PUSH
- R หรือ r : ตรวจจับ Packet ที่มี flag RST

รูปที่ 4.1 ภาพแสดงไฟล์ RULES.SAMPLE

4.2.2 การใช้งานโปรแกรม

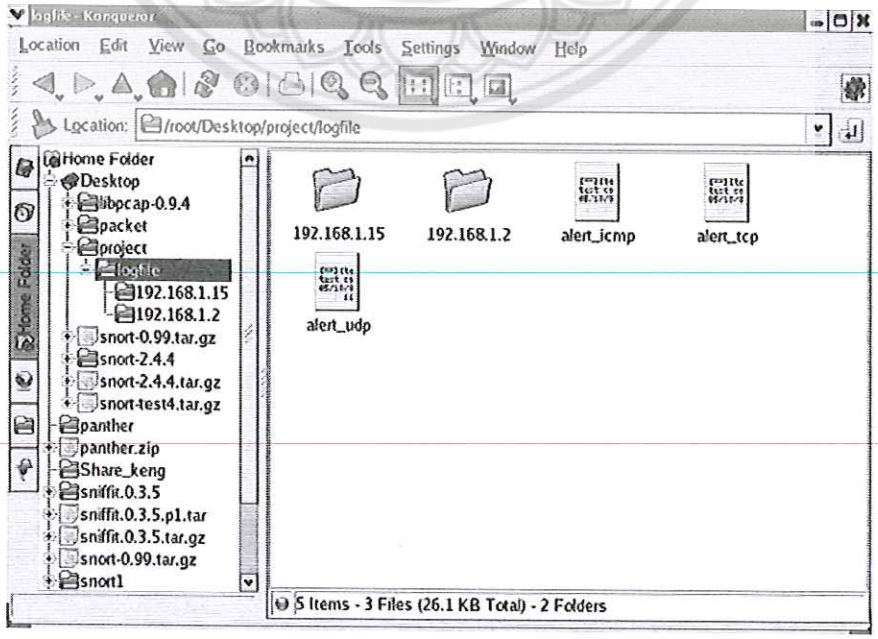
เมื่อทำการกำหนด RULE ของโปรแกรมแล้ว ในส่วนของการของรันโปรแกรม นั้น ให้เข้าไปที่ Folder ของโปรแกรม เมื่อต้องการทำรันโปรแกรมให้ใช้คำสั่ง

./snort



รูปที่ 4.2 ภาพการทำงานของโปรแกรม

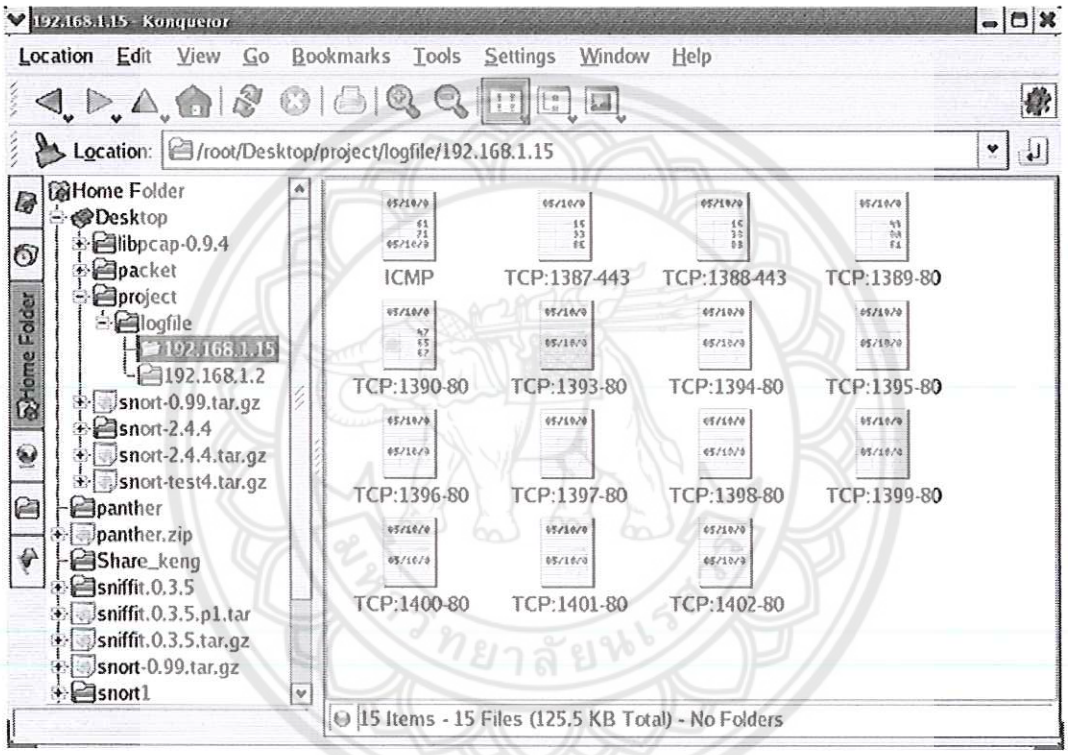
ก่อนการทำรันโปรแกรมนั้นใน Folder ที่ทำการเก็บโปรแกรมต้องมี Folder ชื่อ Logfile ก่อน เพราะ Packet ที่ถูกตรวจจับได้นั้น จะถูกนำไปเก็บไว้ที่ Folder นั้น และเมื่อต้องการหยุดโปรแกรม ให้กดปุ่ม Ctrl + C จะสามารถออกจากโปรแกรมได้



รูปที่ 4.3 ภาพไฟล์ที่เก็บข้อมูล Packet

จากที่เห็นไฟล์ที่ชื่อ alert_icmp, alert_tcp, หรือ alert_udp เป็น File ที่เก็บค่า RULE ALERT ต่างๆ โดยแยก เป็น Protocol ตามชื่อ ส่วน Folder ที่เป็นชื่อหมายเลข IP นั้นคือ การเก็บค่าของ RULE LOG โดยในการส่วนของการ Log นั้นจะเก็บค่าเป็น Folder หมายเลข IP ว่าหมายเลข IP นี้มีการรับส่ง Packet ใดบ้าง

ส่วนภายใน Folder หมายเลข IP ต่างๆ นั้นจะเก็บค่าเป็น file PORT_S:PORT_D ในส่วนของ Protocol TCP และ UDP ถ้าเป็น Protocol ICMP นั้นจะเก็บเป็นไฟล์เดียวนั้นคือ File ICMP



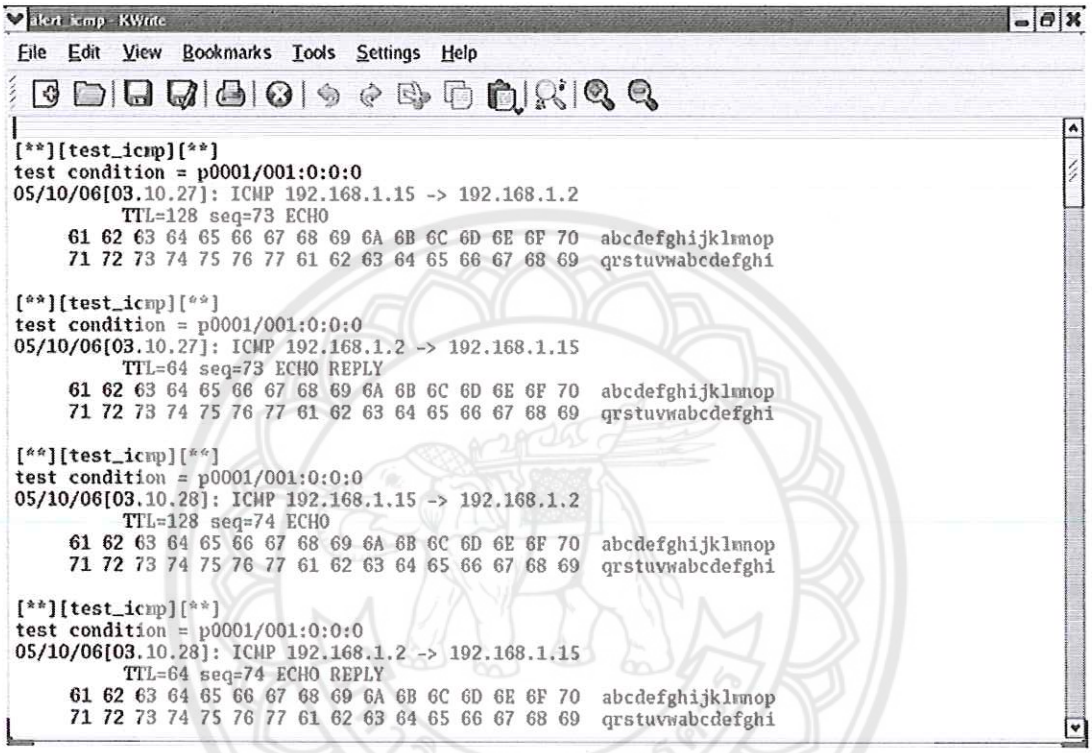
รูปที่ 4.4 file ของการทำ RULE LOG

4.3 การวิเคราะห์ข้อมูล

4.3.1 วิเคราะห์ Packet

เมื่อทำการรันโปรแกรมแล้วจะได้ไฟล์ที่ เก็บ Packet ต่างๆ ซึ่ง ในแต่ละ Packet นั้นมีรายละเอียดของข้อมูลที่ไม่เหมือนกัน จึงต้องทำความเข้าใจในส่วนของข้อมูลภายใน Packet

4.3.1.1 PACKET ICMP



รูปที่ 4.5 ภาพแสดงข้อมูลใน Packet ICMP

ICMP เป็น Protocol ที่ใช้ในการตรวจสอบสถานะของเครื่องต่างๆ ภายในระบบ Network หรืออุปกรณ์บางอย่างในระบบ Network ได้

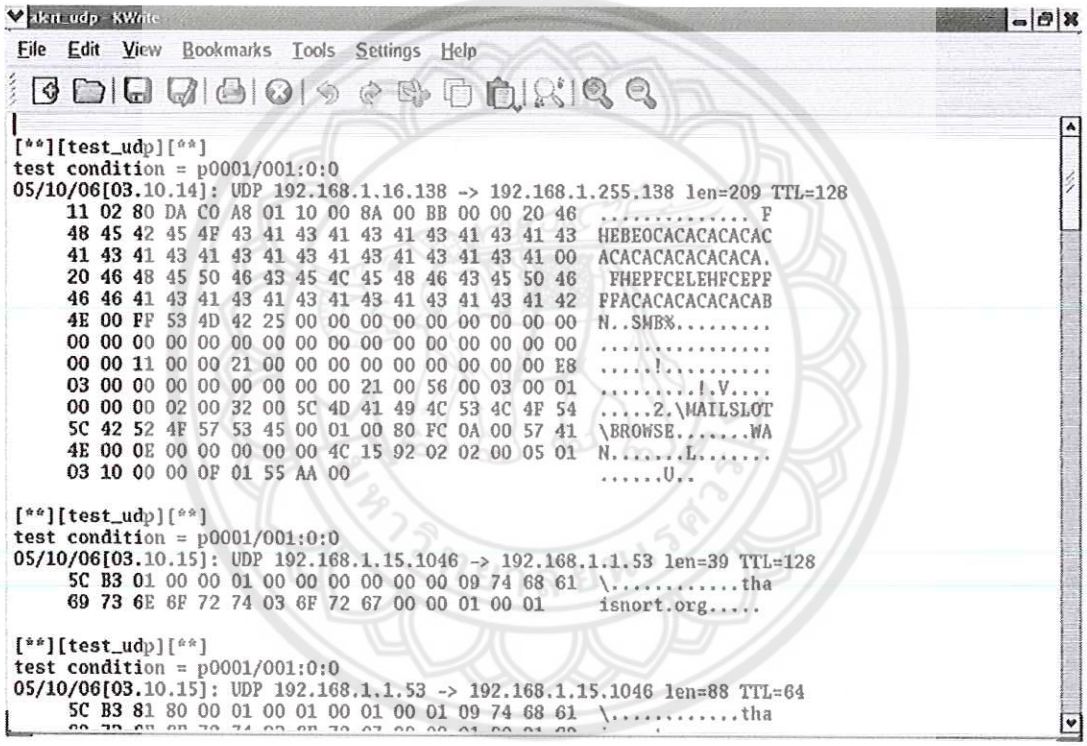
จากรูปที่ 4.5 เป็นรูปที่ได้จากการจับ Packet แบบ RULE ALERT สังเกตได้จาก ทุก Packet จะมีข้อความ test_icmp และ ในส่วนของ IP Address จะเห็นได้ว่า IP ที่ส่งข้อมูลจะเปลี่ยนไป ถ้าเป็นการ Log file จะเก็บค่า IP Address ของผู้ส่งเหมือนกัน

ความหมายต่างๆ ของ packet icmp

- test_icmp : เป็น Message ของ RULE ALERT (เฉพาะ RULE ALERT)
- test condition = p0001/001:0:0:0 : เงื่อนไขที่ถูกกำหนดในการตรวจจับ (เฉพาะ RULE ALERT)
- 05/10/06[03.10.28] : เดือน/วัน/ปี[ชั่วโมง:นาที:วินาที] เวลาที่จับ packet
- ICMP : ชื่อ Protocol

- 192.168.1.2 -> 192.168.1.15 : IPAddress ส่ง -> IP Address รับ
- TTL : Time To Live เป็นค่าที่กำหนดให้ Packet อยู่ในระบบ Network โดยในการ route 1 ครั้ง จะทำให้ค่า TTL ลดลง และเมื่อ TTL ลดลงเหลือ 0 Packet นั้นจะหายไป
- seq = 73 : เป็นลำดับการส่ง Packet
- ECHO : เป็น Message ของ Packet ICMP โดยแบ่งเป็น 2 ชนิดคือ Message query และ Message error
- ในส่วนที่เหลือเป็นค่าข้อมูลของ Packet

4.3.1.2 PACKET UDP



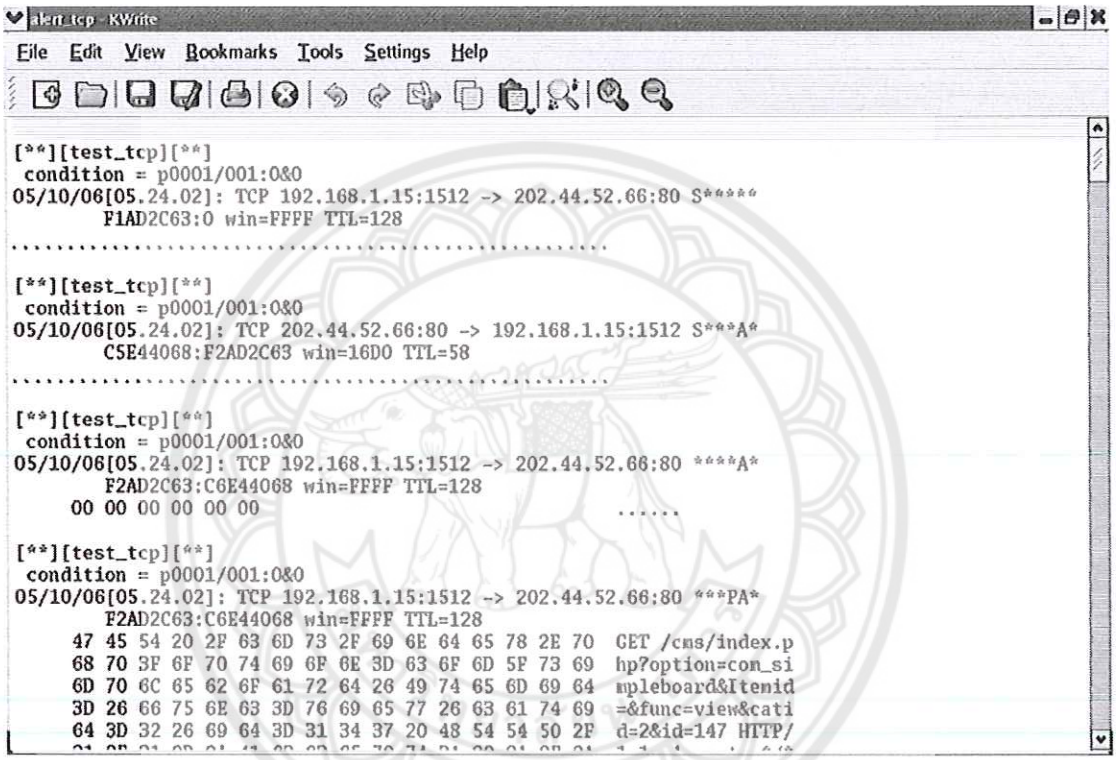
รูปที่ 4.6 ภาพแสดงข้อมูลภายใน Packet UDP

UDP เป็น Protocol ที่ใช้งานแบบ Connectionless (การเชื่อมต่อแบบไม่ต่อเนื่อง) โดยการนำข้อมูลใส่ใน Packet แล้วก็ส่งออกไปโดยไม่ทำการเชื่อมต่อ จากรูปที่ 4.6 สามารถอธิบายค่าต่าง ๆ ของ Packet UDP ได้ดังนี้

- test_udp : เป็น Message ของ RUEL ALERT (เฉพาะ RULE ALERT)
- test condition = p0001/001:0:0 : เป็น Condition ของ RULE ALERT (เฉพาะ RULE ALERT)
- 05/10/06[03.10.15] : เดือน/วัน/ปี[ชั่วโมง:นาที:วินาที] เวลาในกาจับ Packet

- UDP : ชื่อ Protocol
- 192.168.1.16:80->192.168.1.255:138 : IP Address.port ส่ง -> IP Address.port รับ
- len=209 : ขนาดของ packet ที่ตรวจจับได้
- TTL : Time To Live

4.3.1.3 PACKET TCP



รูปที่ 4.7 ภาพแสดงข้อมูลภายใน Packet TCP

TCP เป็น Protocol ที่ใช้ในการติดต่อแบบต่อเนื่อง (Connection-Oriented) เป็นโปรโตคอลที่ถูกแบบออกมาเพื่อใช้ในการสื่อสารได้อย่างมีประสิทธิภาพ จากรูป 4.7 สามารถอธิบายค่าต่างๆ ของ Packet TCP ได้ดังนี้

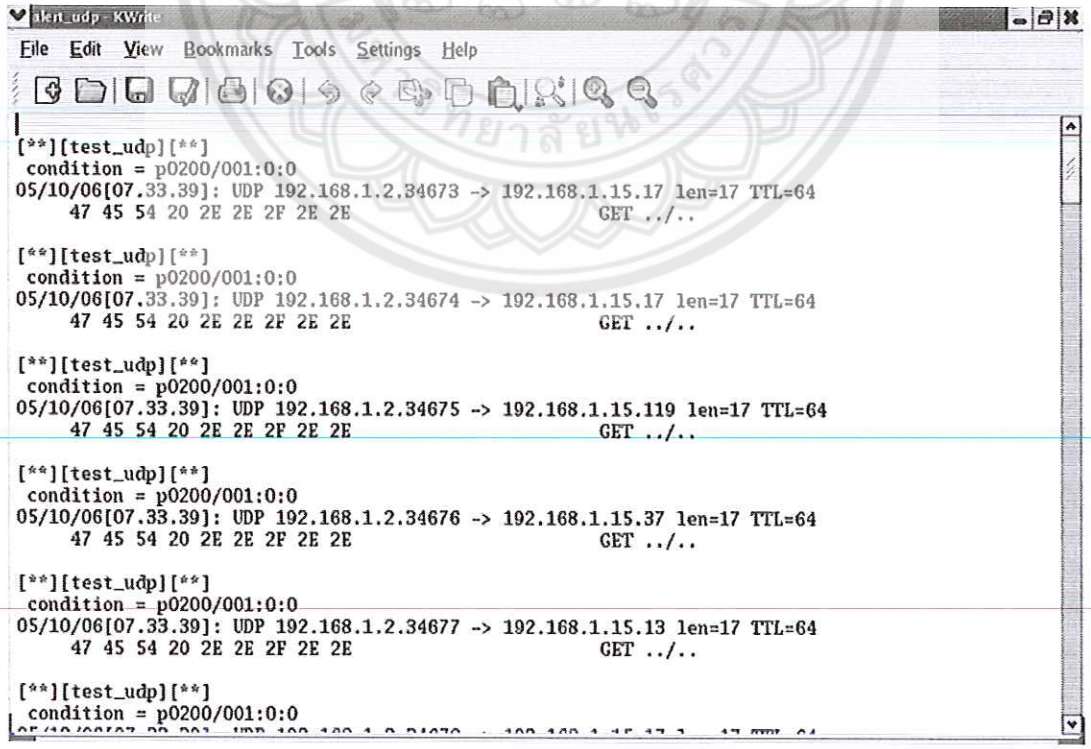
- test_tcp : เป็น Message ของ RULE ALERT (เฉพาะ RULE ALERT)
- condition = p0001/001:0&0 : เป็น Condition ของ RULE ALERT (เฉพาะ RULE ALERT)
- 05/10/06[05.24.02] : เดือน/วัน/ปี[ชั่วโมง:นาที:วินาที] เวลาในกาจับ Packet
- TCP : ชื่อ Protocol

- 192.168.1.15:80 -> 202.44.52.66:80 : IP Address:port ส่ง -> IP Address:port รับ
- S***** : Flags ของ Protocol TCP มีค่าเรียงกันคือ SFRPAU โดยแต่ละตัวอักษรมีความหมาย S=SYN, F=FIN, R=RST, P=PUSH, A=ACK และ U=URG
- C5E44068:F2AD2C63 : เป็นค่า Sequence Number:Acknowledgement Number คือเลขลำดับของ Packet และหมายการเลขลำดับ Packet ต่อไป
- Win = FFFF คือขนาดของ Window size เพื่อบอกขนาดของข้อมูล
- TTL = Time To Live

4.3.2 การวิเคราะห์การโจมตี

4.3.2.1 วิเคราะห์การโจมตีด้วยเวลา

ในการทำงานของโปรแกรมจะมีส่วนของการคำนวณจำนวน Packet ต่อ เวลา ทำให้สามารถสังเกตเห็นการโจมตีได้ โดยปกติแล้วการรับส่งข้อมูลนั้นจะทำงานในเวลาที่รวดเร็ว โดยเฉพาะกับเครื่อง Server ที่ต้องรองรับข้อมูลมหาศาลในการร้องขอ และส่งข้อมูลไป แต่เนื่องจากเครื่องทดสอบเป็นเครื่อง PC client ทั่วไป ในการรับส่งข้อมูลจึงไม่มากนึก แต่ถ้ามี Packet รับส่งข้อมูลกันอย่างรวดเร็วบนเครื่องแล้วนั้นเป็นสิ่งควรสังเกตว่าเครื่องกำลังโดนโจมตี



รูปที่ 4.8 ภาพตัวอย่างการโจมตี 1

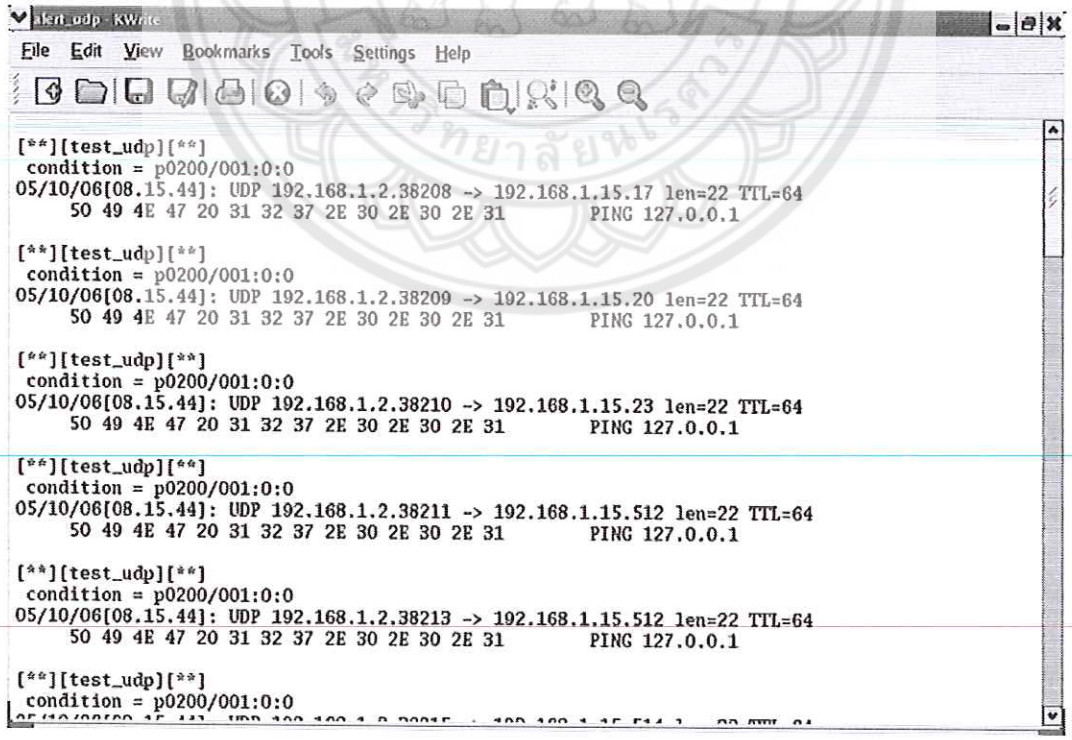
จากรูปที่ 4.8 จะเห็นได้ว่าได้ตั้งค่า Condition Packet/time ไว้ที่ 200 Packet ต่อ 1วินาที นั้นหมายถึงถ้ามีการรับส่งข้อมูล 200 Packet ใน 1 วินาที ถึงตรวจจับข้อมูลดังกล่าว ในเครื่อง PC เท่าไปจะไม่มีการรับส่งข้อมูลที่มากถึง 200 Packet/1 วินาที (ยกเว้นเครื่อง Server) นั้นแสดงถึงการโดนโจมตีได้เช่นกัน

4.3.2.2 การโจมตีของ Protocol UDP

UDP เป็น Protocol ที่มีใช้สำหรับการส่งข้อมูลโดยไม่มีการสร้าง Connection ขึ้นมาเมื่อนำข้อมูลใส่เข้าไปใน Packet จากนั้นก็ทำการส่งไปข้อมูลยังปลายทาง ซึ่งปกติจะเป็นการขอบริการให้ผู้ใช้ส่งไปหนึ่ง Packet และมีผลลัพธ์จากบริการนั้นๆ ส่งกลับมาหนึ่ง Packet

จากรูปที่ 4.8 จะเห็นได้ว่า เป็น Protocol UDP ในข้อมูลนั้น มีค่าเป็น GET ../.. คำสั่ง GET เป็นการร้องขอใช้บริการ จากรูปที่ 4.8 ไม่ได้ระบุการขอบริการเอาไว้ และยังทำการส่งไปเรื่อยๆ ทำให้เครื่องรับไม่สามารถตอบสนองได้ทัน

ในการโจมตีได้ Protocol UDP นั้นบางครั้งอาจส่ง Packet ที่มีข้อมูล ping 127.0.0.1 ซึ่ง127.0.0.1 เป็น Localhost ของเครื่องทุกเครื่องนั้นก็หมายถึงการ ping เข้าหาตัวเองนั้นเอง เมื่อเครื่องได้รับ Packet นี้ก็จะทำการ ping 127.0.0.1 แล้วด้วยจำนวนการส่งข้อมูลที่มาก อีกทั้งต้องทำการ ping เข้าเครื่องตัวเอง อาจทำให้เครื่องเกิด hang ขึ้นมาได้ รูปที่ 4.9 คือตัวอย่างการส่ง Packet ping 127.0.0.1



รูปที่ 4.9 ตัวอย่างการโจมตี 2

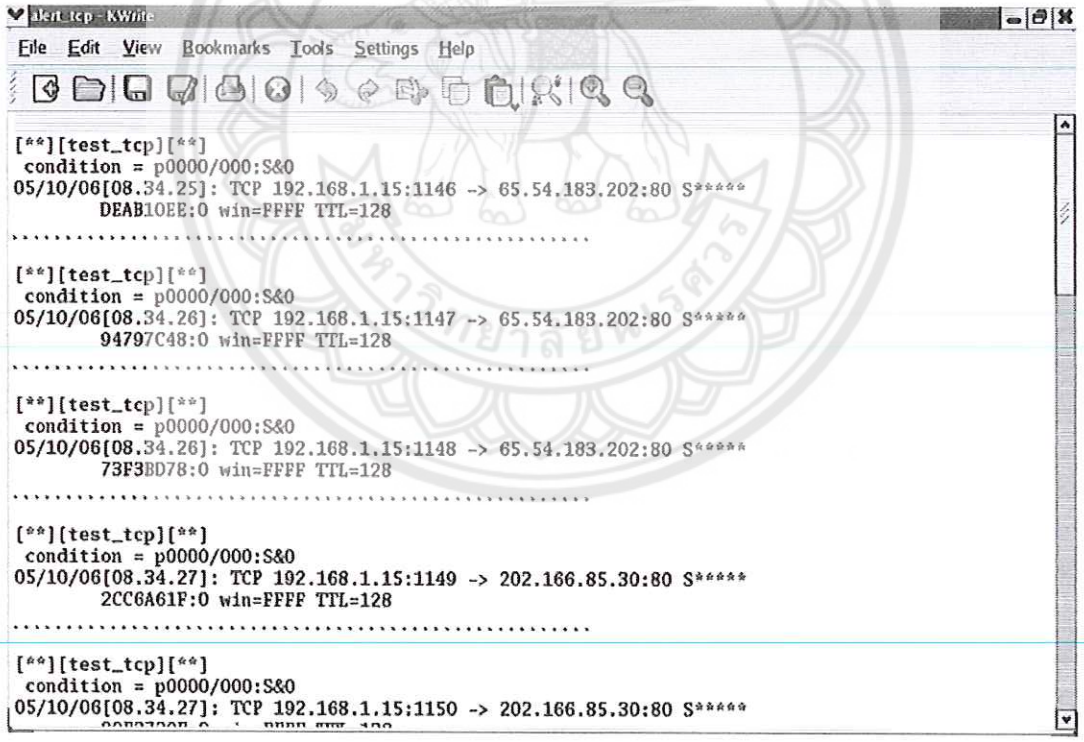
4.3.2.3 การโจมตีของ Protocol ICMP

ในส่วนของ Protocol ICMP นี้ การโจมตีส่วนมากก็ใช้วิธีการส่ง Packet จำนวนมากไปยังปลายทางให้เครื่องปลายทางตอบรับการส่งข้อมูลไม่ไหว โดยทั่วไปมักใช้การ ping ไปยังเครื่องปลายทาง

4.3.2.4 การโจมตีของ Protocol TCP

การโจมตีด้วยของ Protocol TCP นั้นมีหลากหลาย วิธีแต่เนื่องด้วยโปรแกรมที่ได้สร้างขึ้นนั้นมีความสามารถในการตรวจสอบ Flags TCP ได้ โดยการโจมตีด้วย Flags TCP นั้นมีดังนี้

- การส่ง Packet ค่า Flag SYN จำนวนมาก เนื่องจากสัญญาณ SYN เป็นสัญญาณเริ่มต้นการติดต่อ เมื่อส่งไปที่เครื่อง Server แล้วก็เครื่อง Server จะเตรียม Process ไว้รองรับการติดต่อนั้น แต่ถ้าการส่งสัญญาณ SYN นั้นเป็นเข้ามาอย่างรวดเร็วมาก อาจทำให้ Process ทั้งหมดต้องหยุดทำงาน จากรูปที่ 4.10 เป็นการตรวจสอบสัญญาณ SYN flags แต่นั้นไม่ใช้การถูกโจมตี เพราะเมื่อตรวจดูเวลาแล้วจะเห็นว่าไม่ได้การส่ง Packet จำนวนมากไปในเวลาเดียวกัน

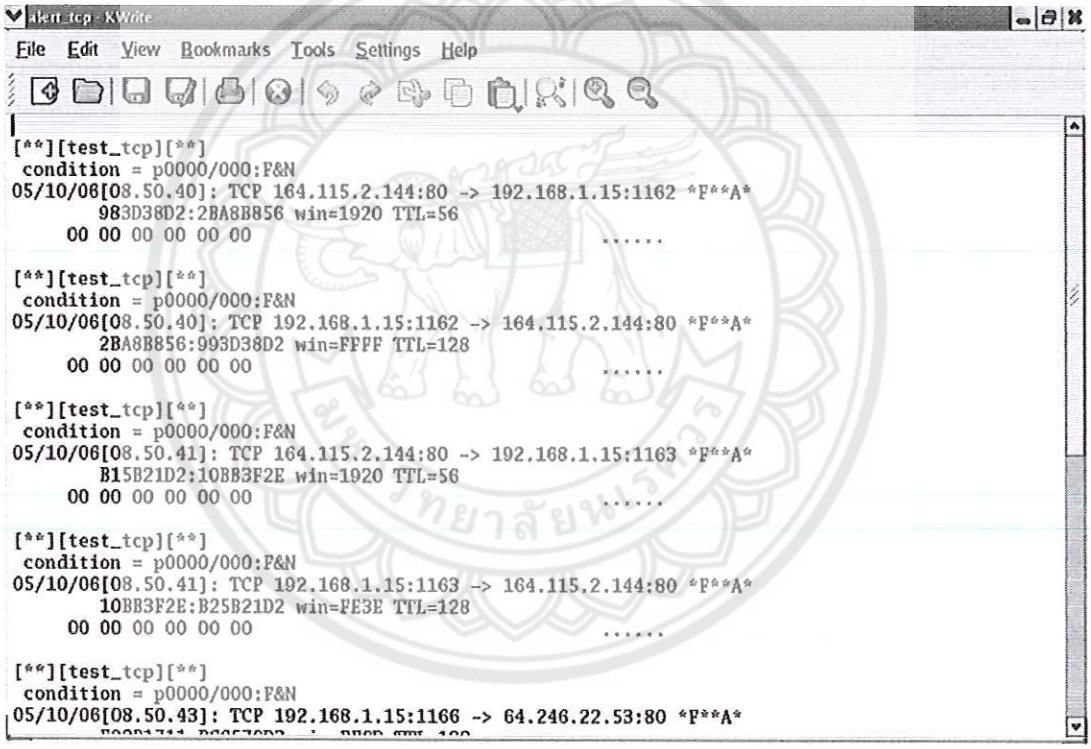


รูปที่ 4.10 ตัวอย่างการจับสัญญาณ SYN flags

- การส่งสัญญาณ ACK แต่ไม่มี Sequence Number โดยปกติเมื่อมีการส่งสัญญาณ ACK ทุกครั้งจะมีค่า Sequence Number ถ้าไม่มีนั้นคือการโจมตีอย่างนึง

- การส่งสัญญาณ SYN และ FIN มาพร้อมกันใน 1 Packet นี้เป็น Packet ที่เป็นไปไม่ได้เพราะ SYN เป็นสัญญาณเริ่มต้นการติดต่อ แต่ FIN เป็นสัญญาณขอหยุดการติดต่อ ถ้ามี Packet แบบนี้เข้ามาในระบบนั้นแสดงว่ากำลังถูกโจมตี

- การส่งสัญญาณ FIN แต่ไม่มี ACK ทุกครั้งที่มี สัญญาณ FIN จะต้องมีสัญญาณ ACK กลับไปด้วย แล้วจึงมีสัญญาณ ACK จากปลายทางเพื่อบอกว่ารับรู้การติดต่อ แต่ถ้ามีเพียงสัญญาณ FIN เพียงอย่างเดียวนั้นหมายถึงการถูกโจมตี จากรูป 4.11 ได้ให้โปรแกรมตรวจจับ Packet TCP โดยตรวจจับ Packet ที่มี Flags FIN ทุก Packet จะเห็นได้ว่าทุกครั้งที่มีการเกิด Flag FIN จะมีสัญญาณ ACK ด้วยตลอดเวลา



รูปที่ 4.11 ตัวอย่างการจับสัญญาณ FIN

จากที่ได้กล่าวถึงการโจมตีทั้งหมดนั้นเป็นแค่พื้นฐานในการศึกษาเพราะว่าในการทำการโจมตีนั้นในปัจจุบันหลายวิธี บางวิธีก็สามารถป้องกันได้ และที่สำคัญ "ไม่มีการป้องกันที่สมบูรณ์แบบ"

บทที่ 5
ปัญหาและข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงงาน

ในการศึกษาการตรวจจับ Packet ด้วยโปรแกรมที่ได้พัฒนาขึ้นนั้นช่วยให้สามารถเข้าใจในส่วนของการวิเคราะห์ข้อมูลใน Packet ของ Protocol ICMP,UDP และ TCP สามารถตรวจสอบการทำงานของระบบ Network ว่ามีการรับส่งข้อมูลแบบใด อีกทั้งยังสามารถตรวจหาการโจมตีบนระบบ Network ได้ โดยโปรแกรมดักจับแพ็กเก็ตสามารถทำงานได้ดังนี้

- สามารถตรวจจับ Protocol ICMP, UDP และ TCP
- สามารถเลือกจำนวน และเวลาในการจับแพ็กเก็ตได้ อีกทั้งสามารถตรวจจับจำนวนแพ็กเก็ตต่อเวลาที่ต้องการได้
- Protocol ICMP สามารถตรวจจับ แพ็กเก็ตที่มีข้อมูลเหมือนๆ กันที่ส่งมา ตรวจสอบ IP ที่เหมือนๆ กันที่ส่งมา และตรวจสอบ message การแจ้ง error ต่างๆ ได้
- Protocol UDP สามารถตรวจจับแพ็กเก็ตที่มี IP และข้อมูลที่เหมือนๆ กันได้
- Protocol TCP สามารถเลือกตรวจจับแพ็กเก็ตที่มี flag TCP ที่ต้องการได้

5.2 ปัญหาที่พบขณะดำเนินงาน

ในช่วงเริ่มต้นนั้นไม่สามารถทำการสร้าง Function ที่เพื่อทำการจับ Packet ในระบบ Network ได้ เนื่องจากยังไม่มีความรู้เบื้องต้นในส่วนนี้ จึงได้นำโปรแกรมถูกสร้างขึ้นแล้วมาใช้ในการพัฒนาต่อ แต่เป็นเพราะไม่เคยได้ใช้โปรแกรมตัวนั้น และภาษาที่ใช้ในการเขียนโปรแกรมนั้นซับซ้อนจึงต้องใช้เวลานานในการพัฒนาขึ้นมา

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

1. ควรมีการตรวจสอบศึกษาการทำงานของระบบ Protocol TCP/IP อย่างละเอียดเพื่อให้เข้าใจในลักษณะส่งรับข้อมูลบนระบบ Network ได้มากขึ้น

2. ควรมีการปรับปรุงโปรแกรมให้มีความสามารถมากขึ้นเพราะโปรแกรมยังมีข้อที่ต้องทำการแก้ไขคือ

- ไม่สามารถตรวจจับ packet ที่ทำการรับส่งในระบบได้ทั้งหมดเพราะมี Packet จำนวนมากที่อยู่ในระบบ Network แต่ถึงอย่างนั้นก็สามารถจับได้มากถึง 98% ของ Packet ทั้งหมด
- ไม่สามารถทำการกำหนด Rule ที่มี Protocol เดียวกันได้

- โปรแกรมนี้สามารถตรวจจับได้ เฉพาะ Protocol ICMP, UDP และ TCP ซึ่งเป็น Protocol หลักที่ใช้ในระบบ Network เท่านั้น

3. โปรแกรมนี้เป็นโปรแกรมที่เหมาะกับการนำไปศึกษาการทำงานของระบบ Protocol TCP/IP และการรับส่งข้อมูลในระบบ Network

เอกสารอ้างอิง

[1] http://www.thaiadmin.org

[2] http://www.thaisnort.org

[3] http://www.elhacker.net

[4] http://www.se-ed.net/hacking/index.html

[5] http://www.snort.org

[6] http://www.tcpdump.org

[7] http://www.cplusplus.com

[8] เรืองไกร รังสิพล, "เจาะระบบ TCP/IP จุดอ่อนของโปรโตคอลและวิธีป้องกัน

[9] Anddrew S. Tanenbaum, สัลยุทธ์ สว่างวรรณ (เรียบเรียง), "Computer Networks"

[10] ประภาพร ช่างไม้, "คู่มือการเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น

ภาคผนวก

ภาคผนวก ก.

ตาราง1 แสดงพอร์ตมาตรฐาน ของ TCP

TCP	
Port number	**Services**
1	Tcpmux
7	Echo
13	Time
17	Qotd(Quote of the day)
19	Chargen
21	Ftp
22	Ssh (Secure Shell)
23	Telnet
43	Whois
53	DNS
70	Gopher
79	Finger
80	Http
87	Link
95	Supdup
109-110	POP
111	Portmap
135	Epmap
139	NetBIOS
143	IMAP
144	News Windows Sys
443	Https
512	Remote Exec
513	Remote login
514	Remote Shell
515	Priter
540	UUCP

TCP(ต่อ)	
Port number	**Services**
749-751	Kerberos
1080	Socks
5632	PC Anywhere
6667	IRC

ตาราง2 แสดงพอร์ตมาตรฐาน ของ UDP

UDP	
Port Number	**Services**
7	Echo
13	Time
17	Qotd
19	Chargen
53	DNS
67-68	BOOTP
69	TFTP
88	Keberos
111	Portmap
137-138	NetBIOS
161-162	SNMP
177	X11 login
513	Who
514	Syslog
517	Talk
518	Ntalk
2049	NFS
5631	PC Anywhere