ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Network Intrusion Detection System (NIDS)

นายธนัญชัย ตรีภาค

นางสาววจี เทศวานิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Network Intrusion Detection System (NIDS)

โดย

นายธนัญชัย ตรีภาค

นางสาววจี เทศวานิช

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Network Intrusion Detection System (NIDS)

ผู้จัดทำ

1. นายธนัญชัย ตรีภาค รหัสประจำตัว 39014211
2. นางสาววจี เทศวานิช รหัสประจำตัว 39014443

_____ อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

นายธนัญชัย ตรีภาค 39014211
นางสาววจี เทศวานิช 39014443
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ความปลอดภัยนับเป็นปัจจัยสำคัญอย่างหนึ่งในการใช้งานคอมพิวเตอร์ที่มีการเชื่อมต่อเป็นเครือข่ายอย่างในปัจจุบัน ซึ่งมีผู้ต้องการโจมตีเครื่องคอมพิวเตอร์ผ่านทางเครือข่ายมากขึ้น ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้น จึงมุ่งเน้นการตรวจจับการโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี ที่มีแนวโน้มเพิ่มขึ้นเรื่อยๆ ในปัจจุบัน ซึ่งการโจมตีในลักษณะนี้สามารถแบ่งออกได้เป็นการส่งแพ็กเก็ตปริมาณมาก การทำแฟร็กเมนเตชันที่ผิดปกติ และแบบที่ผสมกันระหว่างสองแบบข้างต้น ซึ่งมีผลทำให้เครื่องเป้าหมายไม่สามารถให้บริการได้ ดังนั้นจึงออกแบบและพัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ขึ้นบนระบบปฏิบัติการลีนุกซ์ ซึ่งเป็นระบบที่สามารถตรวจจับการโจมตีในลักษณะที่ได้กล่าวมาแล้วข้างต้นได้ครบทุกประเภท ทำให้ผู้ดูแลระบบสามารถตรวจสอบความผิดปกติที่เกิดขึ้นกับระบบของตนได้

Network Intrusion Detection System (NIDS)

Mr. Thanunchai Threepak
Miss. Wajee Teswanich
Mr. Thana Hongsuwan Advisor
Mr. Akkradach Watcharapupong Advisor

ABSTACT

Security is one of the most important topics when using computers on the network because there are many attacks via computer network nowadays. Many of Network Intrusion Detection System (NIDS) developments often aim to emphasise on Denial of Services (DoS) attack, which trends to increase everyday. This kind of network attack can be categorized into three types. The first type is sending considerable amount of packets using TCP/IP protocol stack. The second is called abnormal fragmentation, and the last one is the combination of the two types mentioned above. The consequence from these kinds of attack will cause the targeted host to give up their current services. Thus, this thesis provides a Network Intrusion Detection System development on Linux operating system in order to monitor and analyze the attacks.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี เนื่องจากได้รับการแนะนำ สนับสนุน และให้คำปรึกษาเป็นอย่างดีจาก อาจารย์ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง รวมทั้งอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ให้การอบรมสั่งสอนวิชาความรู้แก่คณะผู้จัดทำมาโดยตลอด

และขอขอบพระคุณเป็นอย่างสูงสำหรับบุคคลที่สำคัญที่สุดที่ทำให้คณะผู้จัดทำมีวันนี้ คือ บิดามารดา ผู้เป็นที่เคารพรักยิ่งของคณะผู้จัดทำ ซึ่งท่านให้การอบรมสั่งสอน เลี้ยงดู และให้โอกาสในการศึกษาอย่างเต็มที่ จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ขอขอบพระคุณผู้ดูแลระบบคอมพิวเตอร์ภาควิชาวิศวกรรมศาสตร์ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่อำนวยความสะดวกในการใช้งานเครือข่าย และขอขอบคุณเพื่อนๆ ที่ให้ข้อคิดเป็น และเป็นกำลังใจให้เสมอมา

คณะผู้จัดทำ

สารบัญ

สารบัญ (ต่อ)

หน้าที่

สารบัญภาพประกอบ

หน้าที่

สารบัญตาราง

หน้าที่

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันเมื่อเครือข่ายคอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ความต้องการในการนำคอมพิวเตอร์ไปใช้ในการติดต่อสื่อสารหรือเชื่อมต่อเพื่อให้บริการแก่กันก็ย่อมมากขึ้นตามไปด้วย ในขณะเดียวกันก็ย่อมต้องมีผู้ไม่ประสงค์ดีต้องการที่บุกรุกทำลายการให้บริการดังกล่าว

นับวันผู้บุกรุกทางเครือข่ายคอมพิวเตอร์จะมีจำนวนมากขึ้น และมีรูปแบบของการโจมตีที่หลากหลายขึ้น การโจมตีรูปแบบหนึ่งที่มีแนวโน้มเพิ่มขึ้น คือ การโจมตีเพื่อให้ปิดบริการ (Denial of Services หรือ DoS) ซึ่งเป็นการโจมตีที่พอร์ตของทีซีพี/ไอพี โดยทั่วไปพอร์ตของทีซีพี/ไอพีเชื่อมต่อกับบริการ (Services) ที่รองรับพอร์ตนั้นๆ ดังนั้นการโจมตีพอร์ตจึงเท่ากับการโจมตีบริการนั่นเอง อาจทำให้ผู้ให้บริการไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการใดๆ ได้เลย

ดังนั้นจึงต้องมีการตรวจจับการโจมตีจากผู้บุกรุกเหล่านี้ จึงเกิดแนวคิดของการทำระบบป้องกันผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Network Intrusion Detection System หรือ NIDS) ขึ้น เพื่อตรวจสอบการบุกรุกทางเครือข่ายคอมพิวเตอร์ที่เกิดขึ้น รวมทั้งแจ้งเตือนไปยังผู้ดูแลระบบ และเก็บข้อมูลไว้ในล็อกไฟล์เพื่อใช้ในการตรวจสอบได้ภายหลัง

ปริญญานิพนธ์นี้จึงมุ่งเน้นการศึกษาประเภทต่างๆ ของการโจมตีแบบ DoS โดยจัดแบ่งประเภทของการโจมตี รวมถึงการศึกษาแนวทางป้องกันการโจมตีแต่ละประเภท เพื่อพัฒนาระบบตรวจจับบนระบบปฏิบัติการลีนุกซ์ให้สามารถตรวจจับการโจมตีดังกล่าวได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์ที่จัดทำขึ้นนี้ จัดทำภายใต้วัตถุประสงค์หลัก 4 ประการ ได้แก่

(1) เพื่อศึกษารายละเอียดและการทำงานของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(2) เพื่อแบ่งประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(3) เพื่อศึกษาแนวทางการป้องกันการโจมตีเพื่อให้ปิดบริการสำหรับสแต็ทีซีพี/ไอพีประเภทต่างๆ

(4) เพื่อสร้างระบบต้นแบบในการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

1.3 ขอบเขตของปริญญานิพนธ์

ขอบเขตการทำงานของปริญญานิพนธ์นี้ ได้แก่

(1) จัดแบ่งประเภทของการโจมตีเพื่อให้เพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(2) ออกแบบ และพัฒนาระบบต้นแบบการตรวจจับการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี แต่ละประเภทบนระบบปฏิบัติการลีนุกซ์

(3) ระบบที่สร้างขึ้นต้องสามารถตรวจจับ แจ้งเตือน และเก็บข้อมูลที่เกิดการโจมตีได้อย่างถูกต้อง

1.4 ขั้นตอนการดำเนินงาน

(1) ศึกษารายละเอียดเกี่ยวกับทีซีพี/ไอพีเบื้องต้น

(2) ศึกษาเกี่ยวกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(3) ศึกษารายละเอียดและการทำงานของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(4) จัดแบ่งประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(5) กำหนดวิธีการตรวจจับการโจมตีแต่ละประเภท

(6) ออกแบบโครงสร้างของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(7) ศึกษาการเขียนโปรแกรมผ่านเครือข่าย

(8) ออกแบบขั้นตอนการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(9) พัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(10) ทดสอบและปรับปรุงระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

บทที่ 2
โพรโตคอลทีซีพี/ไอพี

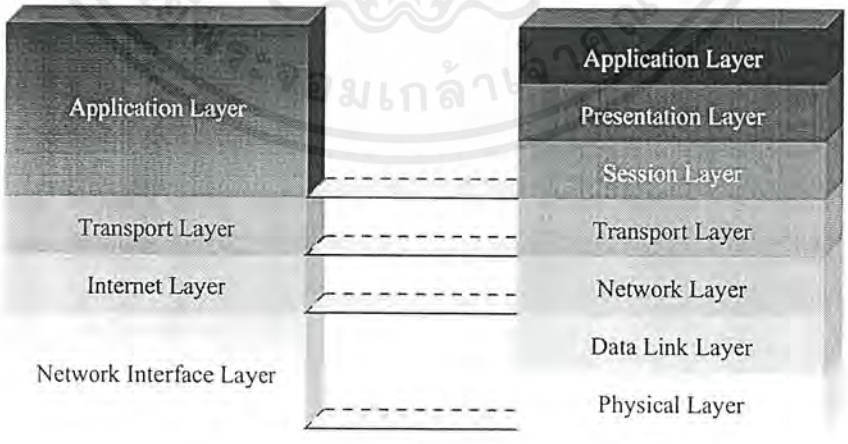
2.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

เป็นโพรโตคอลมาตรฐานที่ใช้กันอยู่ในระบบปฏิบัติการแบบยูนิกซ์ เริ่มพัฒนาโดยกระทรวงกลาโหมของสหรัฐใน ค.ศ. 1969 เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์หลายชนิดที่อยู่ห่างไกลกัน เครือข่ายที่จัดตั้งในระยะแรกชื่อว่า อาร์พาเน็ต (ARPANET)

ต่อมาได้พัฒนาเป็นเครือข่ายอินเทอร์เน็ต โพรโตคอลนี้เหมาะสำหรับเชื่อมต่อคอมพิวเตอร์ทั้งใกล้และไกลเข้าด้วยกัน และมีมาตรฐานรองรับทำให้ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ สามารถสร้างอุปกรณ์และโปรแกรมที่จะรองรับการทำงานของโพรโตคอลนี้ ทำให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ไม่ว่าจะเป็นเครื่องขนาดเล็กหรือขนาดใหญ่ หรือใช้ระบบปฏิบัติการอะไรก็ตาม ทีซีพี/ไอพี (TCP/IP) เป็นชุดโพรโตคอลที่ประกอบด้วยโพรโตคอลต่างๆ หลายโพรโตคอล แต่ละโพรโตคอลมีคุณลักษณะ และมีความสามารถในการทำงานแตกต่างกัน โดยที่ในบทนี้ได้กล่าวถึงรายละเอียดและคุณสมบัติของโพรโตคอลที่สำคัญบางโพรโตคอล

2.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลในการสื่อสารในระบบอินเทอร์เน็ตและอินทราเน็ต การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2-1

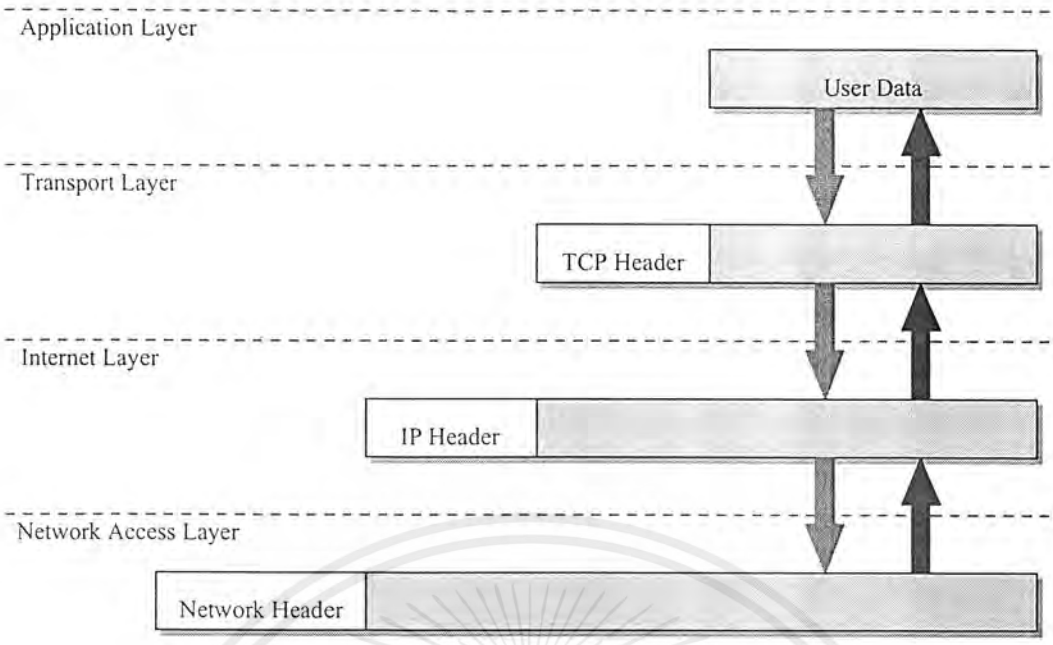


รูปที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังตารางที่ 2-1

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอพพลิเคชัน (Application Layer)	ชั้นนี้รองรับการทำงานของแอพพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่างๆมีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอพพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง
2. ชั้นทรานสปอร์ต (Transport Layer)	มีการสร้างการเชื่อมต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ชั้นนี้มีหน้านี้ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ต คือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP)
4. ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer)	ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย

ตารางที่ 2-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี

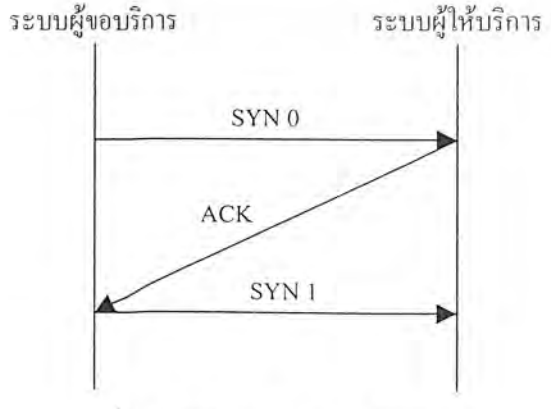


รูปที่ 2-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

ในชุดโพรโตคอลทีซีพี/ไอพีนี้ มีโพรโตคอลหลักที่ขอกล่าวถึง 3 โพรโตคอล ได้แก่ โพรโตคอลทีซีพี โพรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโตคอลไอพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

2.3 โพรโตคอลทีซีพี (TCP: Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งของโพรโตคอลทีซีพี คือ การทำ "3-way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมา จึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2-3



รูปที่ 2-3 แสดงการทำ 3-way Handshake

การเชื่อมต่อแบบ 3-way handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่งและฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

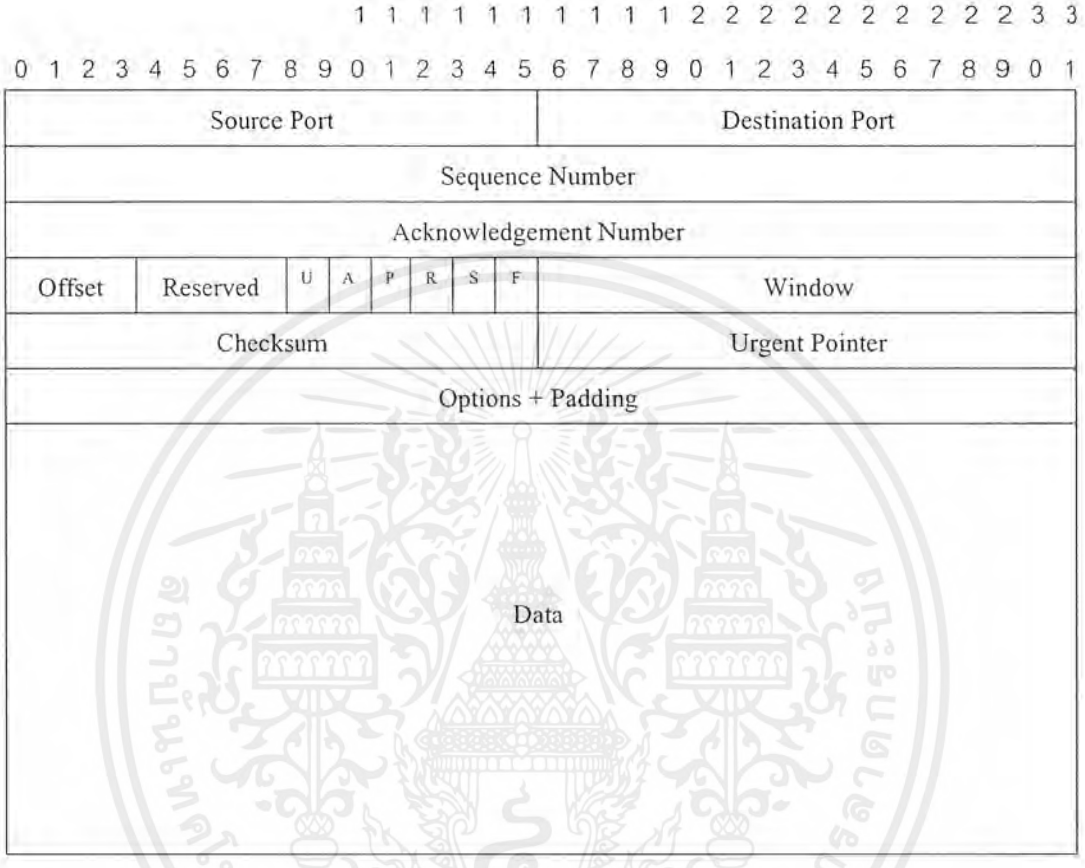
ดังนั้นโพรโตคอลทีซีพีจึงเป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบ "Connection Oriented" ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อ คือ

1. ควบคุมการรับส่งข้อมูล (Basic Data Transfer)
2. ความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
3. ควบคุมการไหลของข้อมูล (Flow Control)
4. การทำมัลติเพล็ก (Multiplexing)
5. ควบคุมการเชื่อมต่อ (Connection)
6. ความปลอดภัยในการรับส่งข้อมูล (Security)

<u>ส่วนประกอบของทีซีพีเฮดเดอร์</u>

1. *Source Port* : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
3. *Sequence Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
4. *Acknowledgement Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูล ปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกฝั่งหนึ่ง) + 1 เสมอ
5. *Data Offset* : เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก
6. *Flag* : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่
    - URG : Urgent Pointer Field Significant - แสดง Urgent Pointer
    - ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement
    - PSH : Push Function
    - RST : Reset The Connection - แสดงเมื่อรีเซ็ทการเชื่อมต่อ
    - SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครนัส
    - FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว
7. *Window* : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control
8. *Checksum* : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล
9. *Urgent Pointer* : เป็นตัวชี้ตำแหน่งของ Urgent Data

10. *Option and Padding* : เป็นตัวบอกออปชันของโพรเซสที่ใช้ทีซีพี
11. *Data* : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)



รูปที่ 2-4 แสดงแพ็กเก็ตทีซีพี

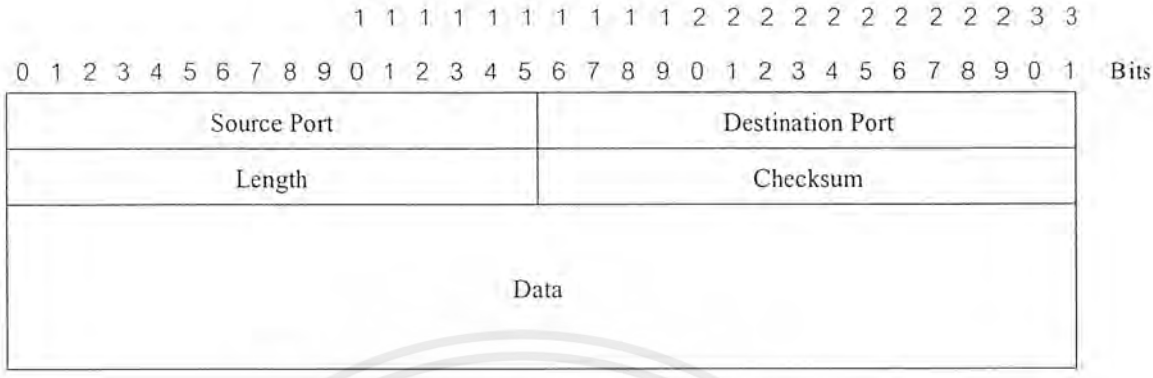
2.4 โพรโตคอลยูดีพี (UDP: User Datagram Protocol)

โพรโตคอลยูดีพีเป็นโพรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือ จัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือ ทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโตคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

<u>ส่วนประกอบของ UDP Frame</u>

1. *Source Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องปลายทาง

3. *Length* : เป็นค่าตัวเลข 16 บิต บอกความยาวของข้อมูล
4. *Checksum* : เป็นค่าตัวเลข 16 บิต ตรวจสอบความถูกต้องของข้อมูลที่ส่ง

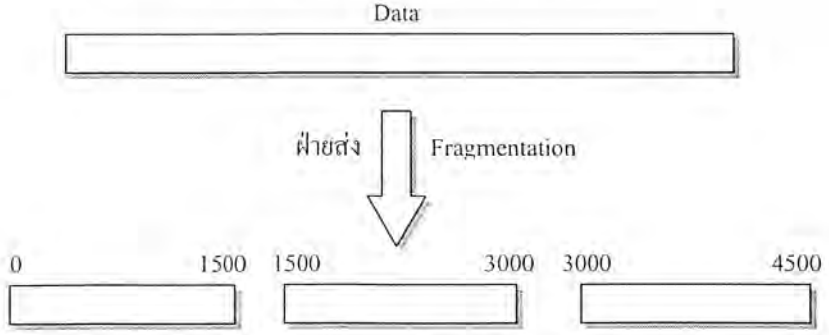


รูปที่ 2-5 แสดงแพ็กเก็ตยูดีพี

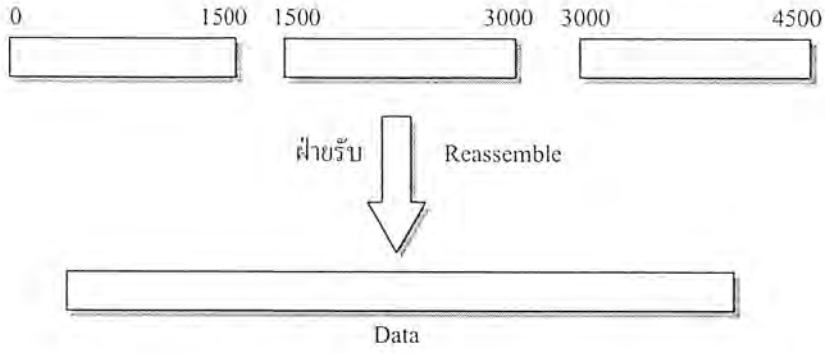
2.5 โพรโตคอลไอพี (IP: Internet Protocol)

โพรโตคอลไอพีเป็นโพรโตคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่

เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ ในระหว่างการส่ง เรียกว่า การทำแฟร็กเมนเตชัน (Fragmentation) เช่น แพ็กเก็ตของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุดเพียง 1,500 ไบต์ ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพ็กเก็ตย่อย และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่า การรีแอสเซมเบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง



รูปที่ 2-6 แสดงการทำแฟร็กเมนเตชัน



รูปที่ 2-7 แสดงการรีแอสเซมเบิ้ล

ส่วนประกอบของแพ็กเก็ตไอพี

1. *version* : เป็นค่าตัวเลข 4 บิต บอกเวอร์ชันของมาตรฐานไอพีที่ใช้ โดยปกติมีค่าเป็น 4 ซึ่งหมายถึง IPv4
2. *Internet Header Length (IHL)* : เป็นตัวบอกความยาวเฮดเดอร์ของไอพี
3. *Type of Service* : เป็นส่วนที่บอกการทำงานของแพ็กเก็ตที่ส่งว่าทำหน้าที่อะไร มีทั้งหมด 8 บิต โดย

   Bit 0-2 : บอกรายละเอียดการทำงานของแพ็กเก็ตนั้นๆ

   111 - Network Control

   110 - Internetwork Control

   101 - CRITIC / ECP

   100 - Flash Override

   011 - Flash

   010 - Immediate

   001 - Priority

   000 - Routine

   Bit 3 : บอกถึงลักษณะของดีเลย์

   0 = Normal Delay - มีดีเลย์ปกติ

   1 = Low Delay - มีดีเลย์ต่ำ

   Bit 4 : บอกถึงประเภทของทรูพุต

   0 = Normal Throughput – มีทรูพุตปกติ

   1= High Throughput – มีทรูพุตสูง

Bit 5 : บอกถึงประเภทของความน่าเชื่อถือ

0 = Normal Reliability - มีความน่าเชื่อถือพอประมาณ

1 = High Reliability - มีความน่าเชื่อถือสูง

Bit 6-7 : กันไว้ใช้ในอนาคต

4. *Total Length* : มีขนาด 16 บิต บอกถึงความยาวในดาต้าแกรมของไอพี
5. *Identification field* : เป็นตัวเลข 16 บิต เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ
6. *Flag* : เป็นตัวเลข 3 bit บอกลักษณะของแพ็กเก็ตว่ามีการแฟร็กเมนต์หรือไม่

Bit 0 : สงวนไว้ ปกติเป็น 0

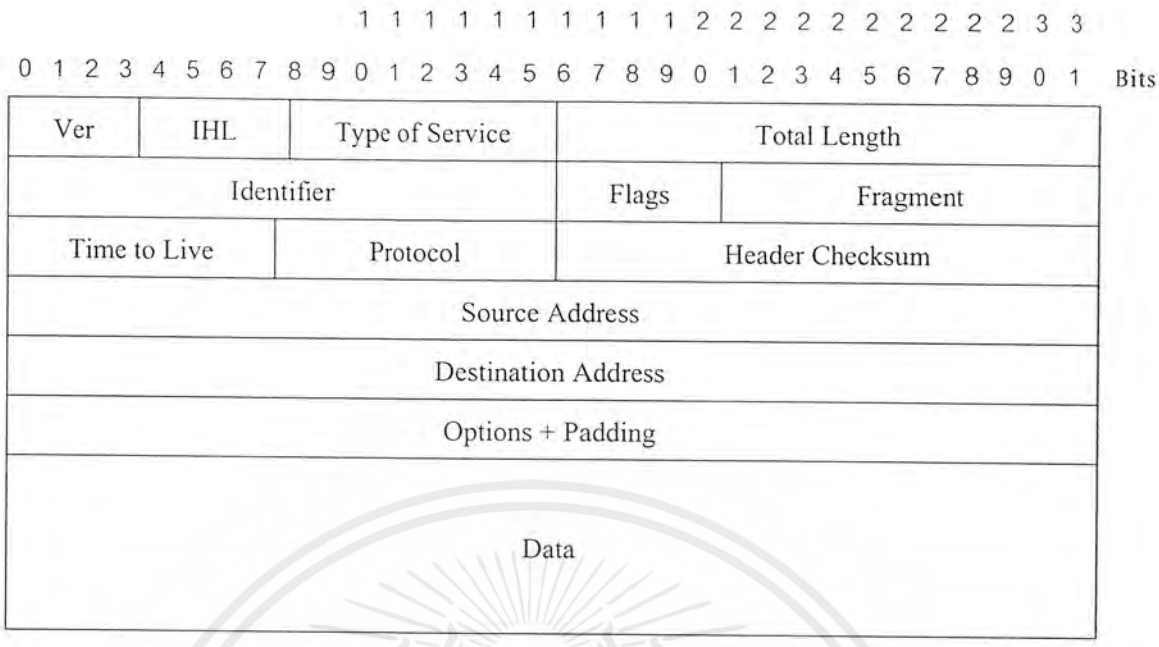
Bit 1 : 0 = บอกว่าแพ็กเก็ตมีการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตไม่มีการแตกแพ็กเก็ตย่อย

Bit 2 : 0 = บอกว่าแพ็กเก็ตนั้นเป็นแพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตนั้นยังไม่ใช่แพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

7. *Fragment Offset* : เป็นค่าตัวเลข 13 บิต บอกออฟเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม
8. *Time To Live (TTL)* : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเราท์เตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปที่ค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเราเตอร์ เพื่อเป็นการป้องกันแพ็กเก็ตล้นเครือข่าย
9. *Protocol* : เป็นตัวเลข 8 bit บอกถึงโพรโตคอลที่อยู่เหนือขึ้นไป ว่าเป็นโพรโตคอลระดับสูงกว่าประเภทใด
10. *Header Checksum* : เป็นค่าตัวเลข 32 บิต ใช้ตรวจสอบความถูกต้องของเฮดเดอร์
11. *Source Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องต้นทาง
12. *Destination Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องปลายทาง



รูปที่ 2-8 แสดงแพ็กเก็ตไอพี

บทที่ 3
การโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี

3.1 ความหมายของการโจมตีเพื่อให้ปิดบริการ

การโจมตีเพื่อให้ปิดบริการ (Denial of Services : DoS) หมายถึง การกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการต่อไปได้อีก โดยทั่วไปโจมตีที่พอร์ตของทีซีพี/ไอพี ซึ่งเชื่อมต่อกับบริการ (Services) ที่รองรับพอร์ตนั้นๆ ดังนั้นการโจมตีพอร์ตจึงเท่ากับการโจมตีบริการของระบบนั่นเอง และอาจมีผลทำให้ระบบนั้นไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการใดๆ ได้เลย

3.2 ประเภทของการโจมตีเพื่อให้ปิดบริการ

ในที่นี้ประเภทของการโจมตีสามารถแบ่งได้ดังต่อไปนี้

3.2.1 ประเภทอยู่ในชั้นทรานสปอร์ต หรือชั้นอินเทอร์เน็ต

การโจมตีในระดับชั้นนี้สามารถแบ่งได้เป็น 2 แบบหลักๆ ได้แก่

3.2.1.1 การส่งแพ็กเก็ตจำนวนมาก (Amount of Packets Sending)

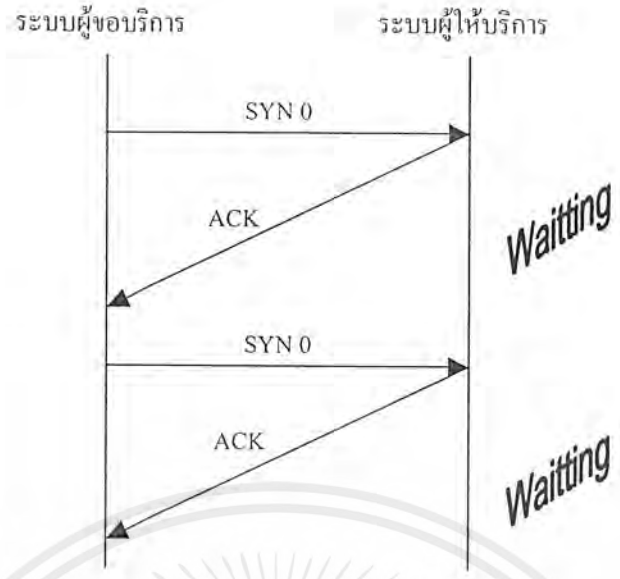
การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตปริมาณมากเข้าไปยังระบบเป้าหมาย อาจทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่าง หรือไม่สามารถทำงานต่อไปได้ ซึ่งแพ็กเก็ตที่ส่งออกไปนี้สามารถแบ่งออกได้เป็น

(1) แพ็กเก็ตข้อมูล (Data Packets)

การโจมตีวิธีนี้ทำได้โดยการส่งแพ็กเก็ตข้อมูลปริมาณมาก เมื่อข้อมูลเข้ามาสู่เครื่องเป้าหมายก็เก็บไว้ในบัฟเฟอร์ก่อนนำมาประมวลผลอีกครั้ง ดังนั้นหากส่งแพ็กเก็ตเข้ามาเป็นปริมาณมาก อาจทำให้บัฟเฟอร์ของเครื่องเป้าหมายไม่เพียงพอที่จะสามารถรองรับแพ็กเก็ตเหล่านั้นได้ทั้งหมด ซึ่งอาจทำให้เครื่องเป้าหมายให้บริการได้ช้าลง หรือต้องหยุดการให้บริการไปเลย

(2) แพ็กเก็ตสำหรับการควบคุม (Control Packets)

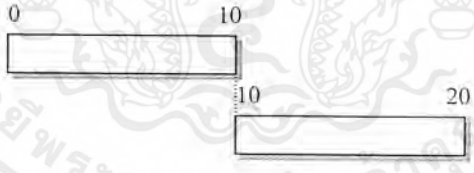
ตัวอย่างของการโจมตีแบบนี้ ได้แก่ การทำ SYN Flooding ปกติการเชื่อมต่อแบบ 3-way handshake เป็นไปตามลักษณะที่ได้อธิบายในหัวข้อ 2.3 แต่ในการโจมตีลักษณะนี้ใช้วิธีทำให้การทำ 3-way handshake ไม่สมบูรณ์ กล่าวคือ เครื่องที่ขอบริการส่งสัญญาณ SYN ไป แต่เมื่อได้รับสัญญาณ ACK จากเครื่องที่ให้บริการแล้ว ไม่ส่งสัญญาณ SYN ตอบกลับไป ทำให้เครื่องที่ให้บริการต้องเปิดการเชื่อมต่อรอการตอบกลับ ดังรูปที่ 3-1 ซึ่งการเปิดการเชื่อมต่อรอเอาไว้นี้ต้องใช้ทรัพยากรของระบบส่วนหนึ่ง และหากมีการส่งสัญญาณในลักษณะนี้มากๆ และทรัพยากรของระบบมีไม่เพียงพอ อาจทำให้ระบบไม่สามารถให้บริการอย่างอื่น หรือให้บริการกับผู้ร้องขอรายอื่นได้



รูปที่ 3-1 แสดงการส่งแพ็กเก็ตแบบ SYN Flood

3.2.1.2 ความผิดปกติของแฟร็กเมนต์ (Abnormal Fragmentation)

การโจมตีวิธีนี้อาศัยหลักการแฟร็กเมนเตชันและรีแอสเซมเบิ้ลที่กล่าวไว้ข้างต้น โดยทำให้แพ็กเก็ตนั้นต้องมีการรีแอสเซมเบิ้ล (กำหนดค่า MF flag = 0) ซึ่งปกติการรีแอสเซมเบิ้ลแพ็กเก็ตทั้งหมดต้องสามารถเชื่อมต่อกันได้สนิท ดังรูปที่ 3-2 แต่แพ็กเก็ตที่ผู้บุกรุกส่งไปมีการแก้ไขข้อมูลในบางฟิลด์ ทำให้เกิดความผิดปกติในกระบวนการรีแอสเซมเบิ้ล ซึ่งการโจมตีในลักษณะนี้ แบ่งได้ดังต่อไปนี้



รูปที่ 3-2 แสดงการรีแอสเซมบลีแบบปกติ

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ (Abnormal Sequences of Packets Sending)

ปกติการส่งแพ็กเก็ตมักเรียงตามลำดับกันไป หากไม่เรียงลำดับก็ต้องรอจนกว่าแพ็กเก็ตก่อนหน้านี้มาถึง เพื่อเรียงลำดับแพ็กเก็ตที่เครื่องรับ แต่การโจมตีแบบนี้กลับส่งเฉพาะแพ็กเก็ตสุดท้าย เพื่อให้ระบบเป้าหมายรอแพ็กเก็ตก่อนหน้า และส่งไปเป็นปริมาณมากๆ เพื่อให้ระบบเป้าหมายไม่สามารถให้บริการอย่างอื่นได้

รูปที่ 3-3 แสดงแพ็กเก็ตสุดท้ายที่ต้องรอแพ็กเก็ตก่อนหน้า

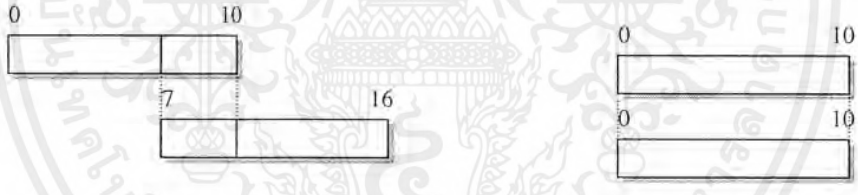
โดยปกติแล้วการโจมตีในรูปแบบนี้ผู้โจมตีจะแก้ไขข้อมูลในฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพี ซึ่งเป็นส่วนที่แสดงลำดับของข้อมูลหลังจากกระบวนการแฟร็กเมนต์เต-ชัน โดยแก้ให้ส่งแพ็กเก็ตสุดท้ายหรือแพ็กเก็ตหลังๆ เพียงแพ็กเก็ตเดียวเลย ทำให้ระบบเป้าหมายต้องรอแพ็กเก็ตก่อนหน้านี้

(2) การส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน (Overlapped Packets' Size Sending)

ปกติแพ็กเก็ตที่ส่งมาต้องนำมาต่อกันที่ระบบเป้าหมายได้พอดี แต่การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน หรือซ้อนทับกัน ทำให้ข้อมูลเมื่อมาต่อกันแล้วเกิดความผิดพลาด หรือไม่สามารถเชื่อมต่อกันได้

โดยปกติแล้วการโจมตีแบบนี้ ผู้บุกรุกสามารถแก้ไขข้อมูลได้ 2 แห่งใหญ่ๆ ได้แก่

- การแก้ไขข้อมูลที่ฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพี หลังจากกระบวนการรีแอสเซมเบิ้ล ซึ่งทำให้ลำดับในการส่งมีความผิดพลาด และอาจเกิดการเหลื่อมล้ำของแพ็กเก็ต กระบวนการรีแอสเซทเบิ้ลอาจเกิดปัญหาได้
- การแก้ไขฟิลด์แสดงความยาวของ (Total Length) ของแพ็กเก็ตไอพี หลังจากกระบวนการรีแอสเซมเบิ้ล ขนาดของแพ็กเก็ตที่มาต่อไม่พอดีกัน ทำให้ไม่สามารถรวมแพ็กเก็ตได้ หรือหากรวมได้ ข้อมูลที่ได้ก็ไม่ถูกต้อง



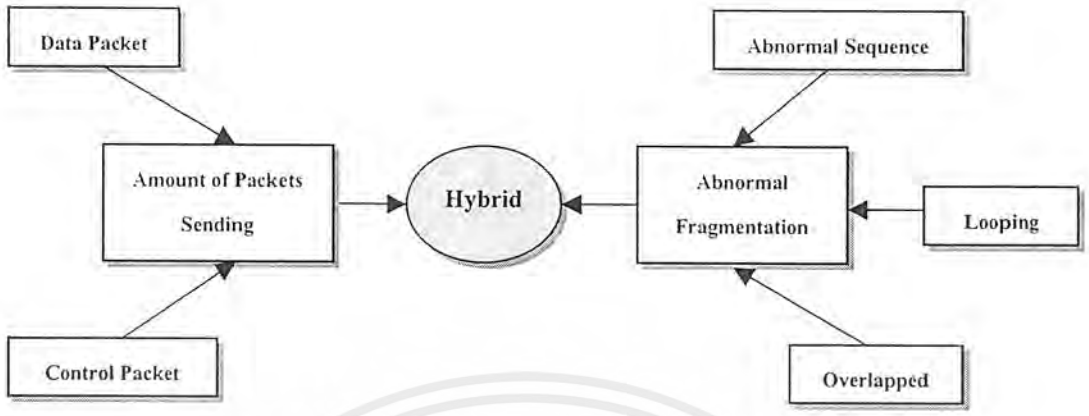
รูปที่ 3-4 แสดงการรีแอสเซมบลีแบบแพ็กเก็ตมีขนาดเหลื่อมกัน

(3) การส่งแพ็กเก็ตแบบวนลูป (Looping)

คือ การส่งโดยกำหนดค่าแอดเดรสต้นทาง (Source Address) และแอดเดรสปลายทาง (Destination Address) ให้เหมือนกันทำให้เกิดการรับส่งวนไปวนมาอยู่ที่เครื่องเป้าหมายเอง เช่น LAND ซึ่งเป็นโปรแกรมโจมตีที่มีการกำหนดแอดเดรสต้นทาง และแอดเดรสปลายทางเป็นค่าเดียวกัน คือ เป็นแอดเดรสของเครื่องเป้าหมายนั่นเอง ทำให้เกิดการส่งวนไปวนมาอยู่ที่เครื่องเป้าหมาย

3.2.1.3 แบบผสม (Hybrid)

คือ การโจมตีที่อาศัยวิธีการผสมกันระหว่างสามแบบแรกที่ได้กล่าวมาแล้ว ดังรูปที่ 3-5



รูปที่ 3-5 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

3.2.2 ประเภทอยู่ในชั้นแอพพลิเคชัน

คือ การโจมตีประเภทอื่นนอกจากที่ได้กล่าวมาแล้วข้างต้น ส่วนใหญ่เกิดจากการใช้จุดอ่อนหรือข้อผิดพลาดของแอพพลิเคชันที่เครื่องเป้าหมายใช้อยู่ในการโจมตีเครื่องเป้าหมายเอง ไม่ว่าจะเป็นจุดอ่อนของระบบปฏิบัติการ หรือข้อผิดพลาดของซอฟต์แวร์ก็ตาม

ในกรณีเช่นนี้เจ้าของเครื่องสามารถแก้ไขได้เอง โดยการนำโปรแกรมแพ็ช (Patch) หรือเซอร์วิสแพ็ค (Service Pack) ต่างๆ มาลงเพื่อแก้ข้อผิดพลาดเหล่านี้ หรือหลีกเลี่ยงไปใช้โปรแกรมอื่นที่ไม่เกิดปัญหา ซึ่งการโจมตีในลักษณะนี้ไม่อยู่ในขอบเขตที่ศึกษา

3.3 โปรแกรมที่ใช้โจมตีเพื่อให้ปิดบริการที่ศึกษา

โปรแกรมที่ใช้ในการโจมตีเพื่อให้ปิดบริการนี้เกิดขึ้นมากมาย และนับวันจะเพิ่มรูปแบบมากขึ้นเรื่อยๆ แต่โปรแกรมในปัจจุบันยังคงมีรูปแบบการโจมตีไม่มากไปกว่าที่ได้กล่าวมาแล้ว ซึ่งโปรแกรมต่างๆ ที่ได้ศึกษามีดังตารางที่ 3-1

ชื่อโปรแกรม	ประเภท	โพรโตคอลที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
1. syncflood	ส่งแพ็กเก็ตสำหรับการควบคุมปริมาณมาก	TCP	เป็นการส่งสัญญาณ SYN ไปขอเปิดการเชื่อมต่อ แล้วเมื่อได้รับ ACK ก็ไม่ส่งสัญญาณ SNY กลับไป ดังที่ได้อธิบายมาแล้ว	- Windows 95
2. oshare	แฟร็กเมนเตชันผิดปกติ (แพ็กเก็ตเหลื่อมล้ำกัน)	IP	เป็นการส่งแพ็กเก็ตที่เหมือนกันมาที่เครื่องเป้าหมาย ทำให้แพ็กเก็ตที่ส่งมาซ้อนทับกัน	- Windows 95/98 - Windows NT 4 + Service Pack 5 ลงมา
3. land	แฟร็กเมนเตชันผิดปกติ (ส่งแพ็กเก็ตแบบวนลูป)	IP	เป็นการส่งแพ็กเก็ตที่มีแอดเดรสต้นทางและปลายทางเป็นค่าเดียวกัน ทำให้เกิดการส่งแบบวนลูป	- Windows 95
4. smurf	แพ็กเก็ตปริมาณมาก	ICMP	เป็นการส่ง ICMP echo (ping) traffic จำนวนมากโดยการ Broadcast ไปยังเครื่องเป้าหมาย	- Windows 95
5. opentear	แฟร็กเมนเตชันผิดปกติ (ต้องรอแพ็กเก็ตก่อนหน้า)	UDP	เป็นการสร้างแพ็กเก็ตสุดท้ายมา เพื่อหลอกให้ระบบเป้าหมายรอแพ็กเก็ตก่อนหน้านี้	- Windows 95
6. pimp	แฟร็กเมนเตชันผิดปกติ (รีแอสเซมเบิ้ลแล้วมีปัญหา)	IP, IGMP	เป็นการสร้างแพ็กเก็ตที่เมื่อรีแอสเซมเบิ้ลแล้วได้ข้อมูลที่ไม่มีความหมาย เกิดเป็นขยะในระบบ	- Windows 95
7. targa	แฟร็กเมนเตชัน (แพ็กเก็ตเหลื่อมล้ำกัน)	IP	เป็นการส่งแพ็กเก็ตที่เหมือนกันมาที่เครื่องเป้าหมาย ทำให้แพ็กเก็ตที่ส่งมาซ้อนทับกัน	- Windows 95/98 - Windows NT 4 + Service Pack 5 ลงมา

ตารางที่ 3-1 ตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ

ชื่อโปรแกรม	ประเภท	โพรโตคอลที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
8. Gin	แพ็กเก็ตสำหรับควบคุมปริมาณมาก	ICMP	เป็นการส่ง ICMP Echo ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98 - Modem
9. raped	แพ็กเก็ตสำหรับควบคุมปริมาณมาก	TCP	เป็นการส่งสัญญา SYN ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98
10. stream	แพ็กเก็ตปริมาณมาก	TCP	เป็นการส่งแพ็กเก็ต TCP ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98 - Modem
11. moya	แพ็กเก็ตสำหรับควบคุมปริมาณมาก	ICMP	เป็นการส่ง ICMP Echo (ping) ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 98
12. Kox	แฟร็กเมนเตชันผิดปกติ (แพ็กเก็ตเหลื่อมล้ำกัน)	IGMP	ส่งแพ็กเก็ตที่เหลื่อมกันไปยังเครื่องเป้าหมาย ทำให้เครื่องเป้าหมายไม่สามารถประกอบแพ็กเก็ตเหล่านั้นได้	- Windows 98/98SE - Windows 2000 build 2000
13. Sesquipedaliam	แฟร็กเมนเตชันผิดปกติ	UDP	ส่งแพ็กเก็ต UDP ที่มีแฟร็กเมนเตชันผิดปกติไปยังเครื่องเป้าหมาย	- Windows 98/98SE
14. smurf	แพ็กเก็ตสำหรับควบคุมปริมาณมาก	ICMP	ส่งแพ็กเก็ต ICMP echo (ping) ไปยังเครื่องเป้าหมายเป็นปริมาณมาก	- Windows 95/98
15. WinArp	อยู่ในชั้นแอปพลิเคชัน	ARP	เป็นการส่งแพ็กเก็ตไปยังเครื่องปลายทาง โดยเครื่องปลายทางต้องกดปุ่ม "OK" สำหรับทุกแพ็กเก็ตที่เข้ามา	- Windows 95/98 - Windows NT 4.0 ลงมา

ตารางที่ 3-1 ตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ (ต่อ)

ชื่อโปรแกรม	ประเภท	โพรโตคอลที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
16. winnuke	อยู่ในชั้นแอปพลิเคชัน	NetBIOS (Port 139)	เป็นการส่ง OOB (Out Of Band) data ไปยังพอร์ต 139 (NetBIOS) ทำให้เกิดหน้าจอฟ้า ขาดการเชื่อมต่อกับอินเตอร์เน็ต และทำให้เกิดปัญหาในการติดต่อบนเครือข่าย ซึ่งจะเกิดกับ	- WFWG 3.11 - Win95 - WinNT 3.51 - WintNT 4.0 + Service Pack 4 ลงมา

ตารางที่ 3-1 ตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ (ต่อ)

ซึ่งเห็นได้ว่าโปรแกรมที่ใช้โจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี มีมากมาย ในที่นี้หลายโปรแกรมเป็นโปรแกรมเก่า จึงไม่สามารถทำลายระบบปฏิบัติการใหม่ๆ ได้ หรือไม่สามารถโจมตีระบบที่ได้อัปเกรดแล้วได้

นอกจากโปรแกรมเหล่านี้แล้ว ยังมีโปรแกรมอื่นๆ ที่ทำให้เกิดการปิดบริการมากมาย และนอกเหนือจากโปรแกรมเหล่านี้ยังมีโปรแกรมที่ทำการโจมตีในชั้นแอพพลิเคชันอีก เช่น coke, winARP, nuke เป็นต้น ซึ่งอยู่นอกขอบเขตที่ศึกษา

บทที่ 4
ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

4.1 ความหมายของการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Network Intrusion Detection System หรือ NIDS) เป็นแขนงหนึ่งของระบบตรวจจับผู้บุกรุก (Intrusion Detection System หรือ IDS) โดยเน้นไปทางการตรวจจับทางเครือข่ายคอมพิวเตอร์เป็นหลัก

โดยระบบนี้ต้องเก็บข้อมูลของแพ็กเก็ตต่างๆ ที่เข้ามาสู่ระบบ แล้วนำมาวิเคราะห์เปรียบเทียบกับกฎต่างๆ ที่ตั้งไว้ รวมถึงนโยบายขององค์กรก็นำมาพิจารณาด้วย เพื่อตรวจสอบว่ามีสิ่งผิดปกติเกิดขึ้นกับระบบหรือไม่ หากเกิดสิ่งผิดปกติ ก็แจ้งเตือนไปยังผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ต่อไป

การตรวจจับผู้บุกรุกทางคอมพิวเตอร์สามารถแบ่งตามลักษณะของการโจมตีได้ 6 ประเภท ได้แก่

(1) การพยายามเจาะเข้าไปทำลายเครือข่าย (Attempted break-ins)

(2) การปลอมแปลงเพื่อเข้ามาโจมตีเครือข่าย (Masquerade attacks)

(3) การอาศัยจุดบกพร่องของระบบรักษาความปลอดภัยเพื่อเจาะเข้าสู่เครือข่าย (Penetration of the security control system)

(4) การโจมตีโดยอาศัยการขาดแคลนทรัพยากร (Leakage)

(5) การโจมตีเพื่อให้ปิดบริการ (Denial of service)

(6) การเข้ามาใช้งานโดยมีเจตนาร้าย (Malicious use)

4.2 ขอบเขตของระบบต้นแบบการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น มุ่งเน้นการศึกษา ออกแบบ และพัฒนาระบบต้นแบบของการตรวจจับการโจมตีเพื่อให้ปิดบริการ โดยเน้นการตรวจจับการโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพีเป็นหลัก โดดเป็นหนึ่งในประเภทของการตรวจจับผู้บุกรุกตามที่ได้กล่าวมาแล้ว ซึ่งมีแนวโน้มเพิ่มขึ้นทุกวัน

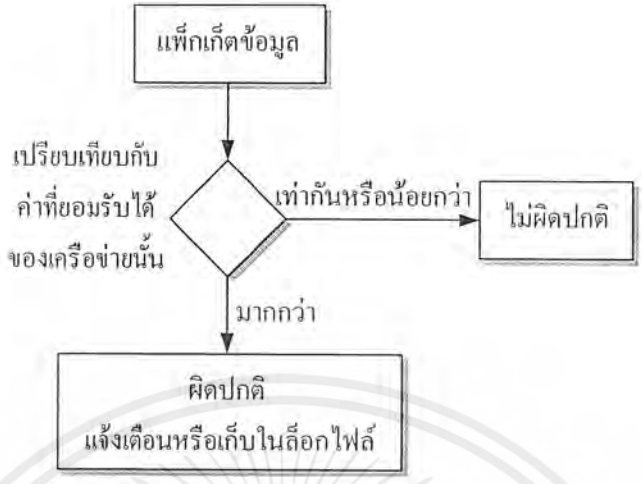
4.3 วิธีการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกที่สร้างขึ้นนี้มีวิธีการตรวจจับผู้บุกรุกที่สามารถแบ่งออกตามประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี เป็น 3 กรณี ได้แก่

4.3.1 การส่งแพ็กเก็ตปริมาณมาก

การตรวจจับแพ็กเก็ตที่เข้ามาในลักษณะนี้ทำได้โดยใช้การนับจำนวนแพ็กเก็ตที่เข้ามาสู่ระบบโดยพิจารณาจากแอดเดรสปลายทาง (Destination Address) ในแพ็กเก็ตเฮดเดอร์ของไอพี หากเป็นค่าเดียวกันให้นับจำนวนแพ็กเก็ตที่เข้ามาในช่วงเวลาหนึ่ง แล้วนำค่าที่ได้มาเปรียบเทียบกับค่าที่ยอมรับได้ หาก

ค่าที่นับได้มากกว่าค่าที่ยอมรับได้ ก็ให้แจ้งเตือนแก่ผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ ซึ่งการทำงานดังกล่าวมานี้ เป็นไปดังรูปที่ 4-1



รูปที่ 4-1 แสดงการตรวจสอบการส่งแพ็กเก็ตปริมาณมาก

ความยากของการวิเคราะห์แบบนี้อยู่ที่การหาค่าที่ระบบยอมรับได้ เพราะขึ้นอยู่กับปัจจัยหลายประการ เช่น ความเร็วของเครือข่าย ความเร็วของหน่วยประมวลผลเครื่อง ปริมาณหน่วยความจำในเครื่อง เป็นต้น

การหาค่าที่ระบบยอมรับได้นี้ สามารถทำได้โดยการเปิดการเชื่อมต่อกับระบบที่วิเคราะห์ จากนั้นหาจำนวนแพ็กเก็ตที่เข้ามาในระบบในลักษณะการใช้งานปกติของแต่ละช่วงเวลา จากนั้นนำค่าสูงสุดที่ได้มาเป็นค่าที่ระบบยอมรับได้ โดยระบบที่วิเคราะห์มีค่าที่ยอมรับได้ประมาณ 20,000 – 30,000 แพ็กเก็ตต่อนาที

4.3.2 ความผิดปกติของแฟร็กเมนต์

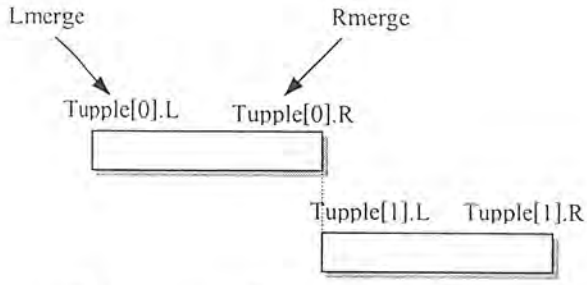
การตรวจสอบความผิดปกติของแฟร็กเมนต์มีขั้นตอนค่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ และแพ็กเก็ตที่มีขนาดเหลื่อมล้ำกัน

การวิเคราะห์ความผิดปกติของแพ็กเก็ตในลักษณะนี้ ต้องวิเคราะห์หลังกระบวนการรีแอสเซมเบิ้ลไปแล้ว ดังนั้นจึงนำบัฟเฟอร์เข้ามาช่วยในการเก็บข้อมูล เพื่อนำมาวิเคราะห์ ดังนี้

- Fragment Buffer

คือ บัฟเฟอร์ที่เก็บข้อมูลในการวิเคราะห์ ซึ่งเก็บข้อมูลของแพ็กเก็ตไอพี และข้อมูลที่จำเป็นอื่นๆ ไว้ ได้แก่ หมายเลขแพ็กเก็ต (ID), จำนวนแพ็กเก็ตที่มีหมายเลขแพ็กเก็ตเดียวกัน, แอดเดรสปลายทาง, ขอบซ้ายและขอบขวาของแพ็กเก็ต (Imerge และ Rmerge) ซึ่งชี้โดยตัวแปร tupple ดังรูปที่ 4-2 จำนวน tupple และแฟล็กแสดงค่าของแพ็กเก็ต



รูปที่ 4-2 แสดงการเก็บข้อมูลของตัวแปร tupple

การเก็บข้อมูลใน Fragment Buffer มีตัวแปรต่างๆ ที่จัดเก็บดังตารางที่ 4-1

ID	count	IP	flag	Lmerge	Rmerge	tupple	Tupple_count	merge

ตารางที่ 4-1 แสดงโครงสร้างการเก็บข้อมูลของ Fragment Buffer

- Overlap Buffer

  คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการเหลื่อมล้ำของแพ็กเก็ต

- Gap Frame Buffer

  คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็กเก็ต

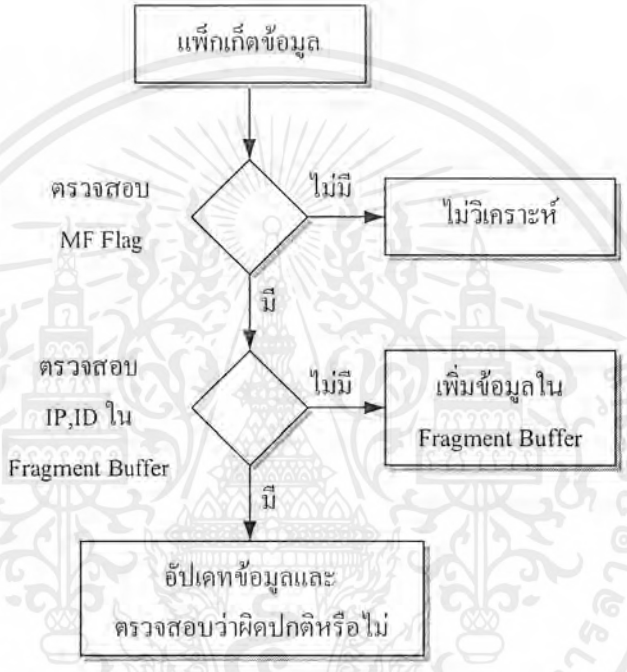
ข้อมูลที่เก็บไว้ของ Overlap Buffer และ Gap Frame Buffer มีโครงสร้างการเก็บข้อมูลที่มีส่วนประกอบเหมือนกัน ดังตารางที่ 4-2

ID	count	IP	Start_min	Start_sec	End_min	End_sec	Etherhdr*

ตารางที่ 4-2 แสดงโครงสร้างการเก็บข้อมูลของ Overlap Buffer และ Gap Frame Buffer

ในการวิเคราะห์ใช้บัฟเฟอร์ทั้งสามนี้ร่วมกัน โดยเก็บข้อมูลแพ็กเก็ตที่เข้ามาทั้งหมดลงใน Fragment Buffer และหากแพ็กเก็ตที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็กเก็ตเดียวที่ต่อเนื่องกัน โดยดูจากขอบซ้ายและขอบขวา เช่น จากรูปที่ 4-2 หาก Tupple[0].R = Tupple[1].L แสดงว่าแพ็กเก็ตทั้งสองนี้สามารถเชื่อมต่อกันได้ ให้รวมเป็นแพ็กเก็ตเดียวกัน โดยแพ็กเก็ตใหม่มี Lmerge = Tupple[0].L และ Rmerge = Tupple[1].R

แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น

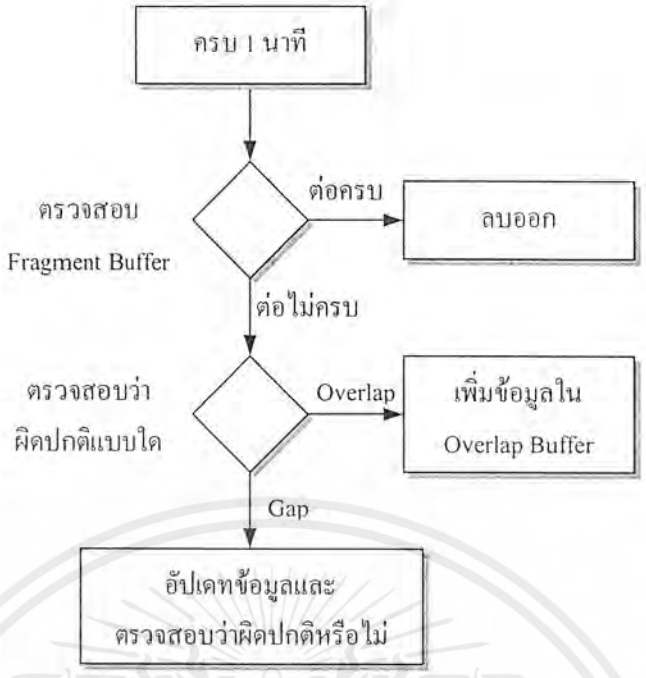


รูปที่ 4-3 แสดงการเก็บข้อมูลลง Fragment Buffer

แต่หากไม่มีความผิดปกติขึ้น เมื่อครบ 1 นาที โปรแกรมตรวจสอบจาก Fragment Buffer ว่าหากมีแพ็กเก็ตใดยังไม่ได้ประกอบ หรือประกอบไม่ครบ ก็ให้เก็บไว้ใน Overlap Buffer หรือ Gap Frame Buffer เช่นเดียวกัน

และเมื่อครบ 1 นาที ข้อมูลใน Overlap Buffer และ Gap Buffer นี้ จะออกมาที่หน้าจอ เพื่อแจ้งให้ผู้ดูแลระบบทราบ หรือเก็บไว้ในล็อกไฟล์ เพื่อบันทึกความผิดปกติที่เกิดขึ้นไว้

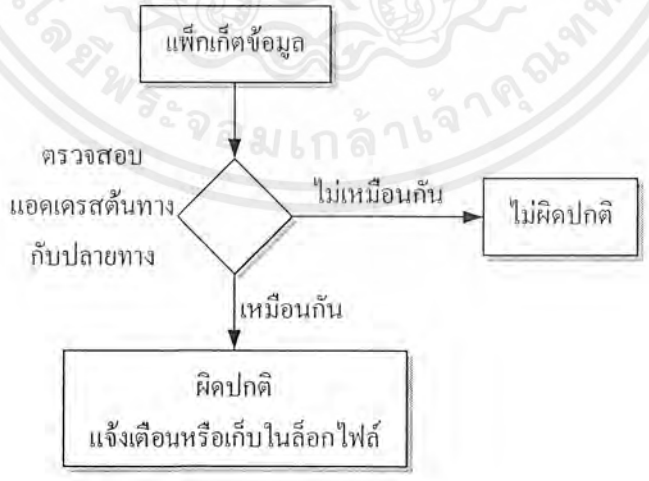
แต่หากไม่มีความผิดปกติใดๆ เกิดขึ้นเลย และแพ็กเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ



รูปที่ 4-4 แสดงการตรวจสอบความผิดปกติในการทำแฟร็กเมนเตชัน

(2) การส่งแพ็กเก็ตแบบวนลูป

สามารถทำได้โดยการเปรียบเทียบค่าแอดเดรสต้นทาง และแอดเดรสปลายทางของแพ็กเก็ตไอพี หากเป็นค่าเดียวกันแสดงว่ามีความผิดปกติเกิดขึ้น เพราะทำให้เกิดการส่งในลักษณะวนลูป ซึ่งขั้นตอนการตรวจสอบเป็นไปตามรูปที่ 4-5



รูปที่ 4-5 แสดงการตรวจสอบแพ็กเก็ตที่ส่งแบบวนลูป

4.3.3 แบบผสม

การวิเคราะห์แพ็กเก็ตประเภทนี้ให้นำวิธีการวิเคราะห์ที่กล่าวข้างต้นมาใช้ร่วมกัน เนื่องจากเกิดจากวิธีการที่ผสมผสานกันระหว่างวิธีต่างๆ ที่ได้กล่าวมาแล้ว ซึ่งสามารถแยกวิเคราะห์ออกเป็นแต่ละแบบหรือวิเคราะห์รวมกันก็ได้

บทที่ 5
การทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

5.1 การทำงานของระบบ

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ แบ่งการทำงานออกเป็น 2 ส่วนได้แก่

5.1.1 การเก็บข้อมูล (Capture Data)

ในส่วนของการเก็บข้อมูลทำงานโดยโปรแกรม **Sniff.c** ซึ่งมีฟังก์ชันการทำงานดังต่อไปนี้

- int InitDevice (char*, int)

จุดมุ่งหมาย : เป็นฟังก์ชันในการเปลี่ยนโหมดของเน็ตเวิร์กการ์ดให้เป็น Promiscuous Mode หรือเปลี่ยนกลับให้เป็นโหมดปกติ

อาร์กิวเมนต์ :char* เป็นค่าของชื่อเน็ตเวิร์กการ์ด ซึ่งก็คือ eth0

pflag กำหนดว่าให้ตั้งค่าเน็ตเวิร์กการ์ดเป็น Promicuous Mode หรือไม่

ขั้นตอนการทำงาน :

(1) เปิดซ็อกเก็ตให้สามารถรับข้อมูลได้ทุกโพรโตคอล

(2) แม็ปซ็อกเก็ตเข้ากับเน็ตเวิร์กการ์ด โดยใช้ชื่อว่า "eth0"

(3) รับค่าแฟล็ก (flag) ของเน็ตเวิร์กการ์ดโดยใช้คำสั่ง ioctl ลงในบัฟเฟอร์ (Buffer)

(4) เปลี่ยนหรือไม่เปลี่ยนเป็น Promiscuous flag ในบัฟเฟอร์ตามเงื่อนไขของ fplay

(5) เก็บค่าแฟล็กที่ได้ลงในอุปกรณ์โดยใช้คำสั่ง ipctl

ผลลัพธ์ : ส่งค่าไปเปลี่ยนโหมดของเน็ตเวิร์คการ์ดให้เป็น Promiscuous Mode หรือโหมดปกติตามแฟล็กที่กำหนด

- void PrintFrame (int hour, int min, int sec, struct etherpacket ep)

จุดมุ่งหมาย : แสดงข้อมูลของแพ็กเก็ตที่เก็บได้ให้อยู่ในรูป

[Time] [Source HW Address] ⟶ [Destination Address]

[PROTO] [Protocol Description]

[Source IP Address] ⟶ [Destination IP Address] [Protocol]

[Packet ID] [Fragment Offset] [Length]

และนำค่าต่างๆ เหล่านี้มาแสดง

- void OpenDataFile ( )

จุดมุ่งหมาย : ทำงาน 2 อย่าง คือ

(1) ดึงค่าเวลาของเครื่องมา แล้วทำการจัดรูปแบบให้อยู่ในรูป hh_dd_mm_yy (hh แทนชั่วโมง, dd แทนวันที่, mm แทนเดือน, yy แทนปี) เพื่อตั้งชื่อไฟล์

(2) เปิดไฟล์นั้นเป็นให้สามารถแก้ไขเพิ่มเติมข้อมูลได้

ไฟล์ดังกล่าวใช้เก็บข้อมูลของแพ็กเก็ตไอพี โดยแยกเก็บเป็นรายชั่วโมง

ขั้นตอนการทำงาน :

(1) ดึงค่าเวลาของเครื่อง โดยใช้คำสั่ง time

(2) เปลี่ยนรูปแบบของค่าที่ได้เป็น ASCII โดยใช้คำสั่ง asctime ซึ่งได้สตริง (String) ของเวลาในขณะนั้น เช่น

" TUE Mar 7 20:20:38 ICT 2000 "

(3) ตัดต่อจัดรูปแบบให้อยู่ในรูป hh_dd_mm_yy จากข้อมูลในข้อ 2) ได้เป็น

" 20_07_Mar_2000 "

(4) สร้างหรือเปิดไฟล์ดังกล่าว

- void CloseDataFile ( )

จุดมุ่งหมาย : ปิดไฟล์ที่เปิดอยู่

- void KeepData ( )

จุดมุ่งหมาย : เขียนข้อมูลลงไฟล์ที่กำหนดไว้

ขั้นตอนการทำงาน :

(1) ตรวจสอบว่าครบ 1 ชั่วโมงหรือยัง หากครบแล้วก็ปิดไฟล์เดิม แล้วเปิดไฟล์ใหม่ขึ้นมาเก็บข้อมูลแทน เนื่องจากการเก็บข้อมูลจะเก็บเป็นรายชั่วโมง

(2) เขียนข้อมูลที่ได้ลงไฟล์ที่เปิดอยู่

- void ExitProgram ( )

จุดมุ่งหมาย : มีจุดมุ่งหมาย 2 ประการ คือ

(1) เปลี่ยนเน็ตเวิร์กการ์ดให้เป็นโหมดปกติ

(2) ปิดไฟล์ที่เปิดอยู่

ขั้นตอนการทำงาน :

(1) เรียกฟังก์ชัน InitDevice โดยป้อนชื่ออุปกรณ์เป็น eth0 และให้ pflag = 0

(2) เรียกฟังก์ชัน CloseDataFile เพื่อปิดไฟล์

(3) สั่ง exit(0) ซึ่งเป็นการออกจากโปรแกรมแบบปกติ (Normal Exit)

- void Sniff (int, int, int)

จุดมุ่งหมาย : เก็บข้อมูลในเครือข่าย

อาร์กิวเมนต์ :มีตัวแปร Integer ทั้งสามเป็นแฟล็กสำหรับ

(1) เก็บข้อมูลลงไฟล์

(2) พิมพ์ค่าต่างๆ ในแพ็กเก็ตที่เก็บได้

(3) วิเคราะห์เครือข่ายแบบทันทีทันใด (Real Time)

ขั้นตอนการทำงาน :

(1) ตั้งค่าเน็ตเวิร์กการ์ดให้เป็น Promiscuous Mode และกำหนดค่าเริ่มต้นที่จำเป็นสำหรับแฟล็กแต่ละตัว

(2) สร้างลูปไม่รู้จบ (Infinite Loop) โดยในลูปทำงานดังต่อไปนี้

- ตั้งค่าเน็ตเวิร์กการ์ดให้อ่านข้อมูล
- ตั้งค่าความถี่ในการอ่าน
- ใช้คำสั่ง recvfrom เพื่อดึงข้อมูลจากเน็ตเวิร์กการ์ดมายังบัฟเฟอร์ที่กำหนด
- ตรวจสอบแฟล็กแล้วทำงานตามแฟล็กต่างๆ ที่กำหนดไว้

หมายเหตุ : ในฟังก์ชันนี้ก่อนการเข้าสู่ลูปไม่รู้จบมีการตั้งค่าเอาไว้ว่า หากมีการกด Ctrl+c หรือฆ่าโพรเซส (kill process) ที่ทำงานอยู่ ให้เรียกฟังก์ชัน ExitProgram อัตโนมัติ เพื่อจุดประสงค์ดังนี้

(1) ตั้งค่าเน็ตเวิร์กการ์ดให้กลับเป็นโหมดปกติ

(2) ปิดไฟล์ข้อมูลที่เปิดอยู่

(3) ออกจากโปรแกรมแบบปกติ

ผลลัพธ์ : ข้อมูลของแพ็กเก็ตที่เขียนลงไฟล์ หรือพิมพ์ออกหน้าจอ ซึ่งได้แก่ เฮดเดอร์ของแพ็กเก็ตในชั้นทีซีพี/ไอพี และเฮดเดอร์ของอีเทอร์เน็ตบางส่วน รวมทั้งวันเวลาที่แพ็กเก็ตเข้ามาด้วย

5.1.2 การวิเคราะห์ข้อมูล (Analyze Data)

ในส่วนนี้โปรแกรมจะนำข้อมูลของแพ็กเก็ตที่เก็บไว้มาวิเคราะห์ว่าเกิดความผิดปกติหรือไม่ และเกิดความผิดปกติในรูปแบบใด โดยแบ่งโปรแกรมที่ใช้วิเคราะห์ออกเป็น 3 โปรแกรม เพื่อใช้ในการวิเคราะห์ความผิดปกติที่แตกต่างกัน 3 ประเภท (การใช้งานจริงมีตัวมาเรียกให้ทำงานพร้อมกันทั้ง 3 โปรแกรม) ดังต่อไปนี้

(1) Analys1.c เป็นโปรแกรมสำหรับตรวจสอบการโจมตีแบบมีการส่งแบบวนลูป

(2) Analys2.c เป็นโปรแกรมตรวจสอบการโจมตีที่ส่งข้อมูลจำนวนมาก (Flood)

(3) Fragment.c เป็นโปรแกรมสำหรับวิเคราะห์แพ็กเก็ตที่มีความผิดปกติของแฟร็กเมนต์

(4) BombCheck.c เป็นโปรแกรมสำหรับตรวจสอบการส่งแพ็กเก็ตมาโจมตีปริมาณมาก

(5) HostCheck.c เป็นโปรแกรมตรวจสอบโฮสต์ที่ต้องการตรวจจับไฟล์ /etc/isagnid.host

(6) IPCheck.c เป็นโปรแกรมสำหรับนับจำนวนแพ็กเก็ตของไอพีต่อช่วงเวลา และต่อโฮสต์ โดยแต่ละโปรแกรมมีฟังก์ชันการทำงานที่แตกต่างกัน ดังต่อไปนี้

5.1.2.1 Analys1.c

สามารถตรวจสอบการส่งแบบวนลูปโดยมีฟังก์ชันการทำงานหลัก ได้แก่

void LANDCheck (char*, int)

จุดมุ่งหมาย : ตรวจสอบการโจมตีที่มีการส่งแพ็กเก็ตที่ทำให้เกิดการวนลูป (LAND Attack)

อาร์กิวเมนต์ : char* ชื่อของไฟล์ข้อมูล

int ค่าแฟล็กของล็อกไฟล์

การตรวจสอบ : การส่งแพ็กเก็ตแบบที่ทำให้เกิดลูปนี้เกิดจากการกำหนดค่าแอดเดรสต้นทาง (Source Address) และแอดเดรสปลายทาง (Destination Address) เป็นค่าเดียวกัน ดังนั้นการตรวจสอบจึงสามารถตรวจสอบได้จากค่าทั้งสองนี้

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) สร้างลูปให้อ่านข้อมูลจนกว่าจะจบไฟล์ โดยภายในลูปให้ตรวจสอบแอดเดรสต้นทางและแอดเดรสปลายทางว่าเป็นค่าเดียวกันหรือไม่ หากเป็นค่าเดียวกันให้เก็บค่าไว้ในบัฟเฟอร์

(3) ปิดไฟล์ข้อมูล

(4) พิมพ์หรือเก็บข้อมูลที่เก็บไว้ในบัฟเฟอร์ออกมาตามที่แฟล็กกำหนด

ผลลัพธ์ : ผลการโจมตีพิมพ์ออกที่หน้าจอหรือเก็บลงล็อกไฟล์ ว่า "LAND Attack", วัน เดือนปีที่เกิด, แ อดเดรสของเครื่องเป้าหมาย, ช่วงเวลาที่โจมตี, ปริมาณของแพ็กเก็ต

5.1.2.2 Analys2.c

ใช้ในการวิเคราะห์ในกรณีมีการส่งแพ็กเก็ตจำนวนมาก ซึ่งแพ็กเก็ตที่เข้ามามีหมายเลขแพ็กเก็ต (Packet's ID) ในไอพีเฮดเดอร์ต่างกัน ซึ่งหมายความว่าแพ็กเก็ตเหล่านั้นมาไม่ได้เป็นแพ็กเก็ตที่มีจากการทำแฟร็กเมนเตชันของแพ็กเก็ตเดียวกัน เนื่องจากอาจมีกรณีที่แพ็กเก็ตเข้ามาเป็นปริมาณมากเนื่องจากแพ็กเก็ตที่ส่งเข้ามายังเครื่องเป้าหมายเป็นแพ็กเก็ตขนาดใหญ่จึงต้องมีการแบ่งแพ็กเก็ตย่อยๆ ปริมาณมาก ซึ่งไม่ถือว่าเกิดการโจมตีแบบนี้ โดยมีฟังก์ชันการทำงานดังนี้

- void CheckFlood ( )

  จุดมุ่งหมาย :เป็นการตรวจสอบการโจมตีกรณีส่งแพ็กเก็ตปริมาณมาก (Flooding) ซึ่งไม่ได้เกิดจากการทำแฟร็กเมนเตชันของแพ็กเก็ตเดียวกัน (มีหมายเลขแพ็กเก็ตเดียวกัน)

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) นับจำนวนแพ็กเก็ตที่มีแอดเดรสปลายทางเดียวกัน แต่มีหมายเลขแพ็กเก็ตต่างกัน แล้วเปรียบเทียบกับค่าที่เหมาะสมของเครือข่าย (แต่ละเครือข่ายมีค่านี้แตกต่างกันออกไป แล้วแต่ประสิทธิภาพของเครือข่าย)

(3) พิมพ์หรือเก็บในล็อกไฟล์หากมีจำนวนแพ็กเก็ตมากเกินปกติ

ผลลัพธ์ : ผลการโจมตีพิมพ์ออกที่หน้าจอหรือเก็บลงล็อกไฟล์ ว่า "Flood Attack", วัน เดือนปีที่เกิด, แ อดเดรสของเครื่องเป้าหมาย, ช่วงเวลาที่โจมตี, ปริมาณของแพ็กเก็ต

- void ShowFloodResult ( )

จุดมุ่งหมาย :แสดงผลทางหน้าจอ หรือเก็บในล็อกไฟล์เมื่อตรวจพบ

5.1.2.3 Fragment.c

เป็นการวิเคราะห์แพ็กเก็ตที่มีความผิดปกติของการทำแฟร็กเมนเตชัน และการรีแอสเซมเบิ้ล โดยมีฟังก์ชันต่างๆ ที่ทำงานดังต่อไปนี้

- void CheckFragment (char* datafile)

จุดมุ่งหมาย : ตรวจสอบการโจมตีแบบ Fragment จาก file

อาร์กิวเมนต์ : ชื่อไฟล์ข้อมูล

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) ตั้งค่าเริ่มต้นของตัวแปรต่างๆ ที่จำเป็นต้องใช้

(3) อ่านไฟล์ข้อมูล แล้วเรียก ฟังก์ชัน RealTimeFragChk ( ) เพื่อวิเคราะห์ข้อมูล

(4) ปิดไฟล์ข้อมูล

(5) แสดงผลการวิเคราะห์ข้อมูล

ผลลัพธ์ : ผลการโจมตีพิมพ์ออกที่หน้าจอหรือเก็บลงล็อกไฟล์ ว่า "LAND Attack", วัน เดือนปีที่เกิด, แ อดเดรสของเครื่องเป้าหมาย, ช่วงเวลาที่โจมตี, ปริมาณของแพ็กเก็ต

- void RealTimeFragChk ( )

จุดมุ่งหมาย : ทำการวิเคราะห์ระบบจากบัฟเฟอร์ ซึ่งสามารถทำงานได้แบบตามเวลาจริง สามารถตรวจสอบการโจมตีแบบแพ็กเก็ตเหลื่อมกัน ซ้อนทับกัน หรือประกอบกันไม่ได้

ขั้นตอนการทำงาน :

(1) ตรวจสอบว่างแพ็กเก็ตนั้นมีการแฟร็กเมนเตชันหรือไม่ ถ้าไม่มีการแฟร็กเมนเตชันก็ไม่นำแพ็กเก็ตนั้นมาวิเคราะห์

(2) ตรวจสอบค่าหมายเลขแพ็กเก็ต (ID) ของแพ็กเก็ตนั้นว่ามีอยู่ใน Fragment Buffer หรือไม่ถ้ามี ให้เรียกฟังก์ชัน FoundInBuffer ( ) แต่ถ้าไม่มี ให้เรียกฟังก์ชัน NotFoundBuffer ( )

- void Initialize ( )

จุดมุ่งหมาย : ตั้งค่าเริ่มต้นของตัวแปรต่างๆ ที่จำเป็นต้องใช้

- void NotFoundInBuffer ( )

จุดมุ่งหมาย : ทำงานทุกๆ ขั้นตอน ในกรณีที่มีแพ็กเก็ตที่มีหมายเลขแพ็กเก็ตไม่ซ้ำกับข้อมูลใน Fragment Buffer

ขั้นตอนการทำงาน :

(1) เพิ่มข้อมูลของแพ็กเก็ตที่รับเข้ามาใหม่ใน Fragment Buffer ซึ่งมี Lmerge, Rmerge, IP, ID, tupple count, count, tupple และ flag

(2) ถ้าข้อมูลเต็มบัฟเฟอร์ (มีแพ็กเก็ตเข้ามามากกว่าที่ได้จองบัฟเฟอร์เอาไว้) แสดงว่ามีแนวโน้มการเกิด flooding จึงมีการเรียกใช้ฟังก์ชัน CheckFlood ( ) ถ้าบัฟเฟอร์เต็ม

- void FoundInBuffer ( )

จุดมุ่งหมาย : ทำงานทุกๆ ขั้นตอนในกรณีที่มีแพ็กเก็ตที่มีหมายเลขแพ็กเก็ตซ้ำกับข้อมูลใน Fragment Buffer

ขั้นตอนการทำงาน :

(1) ทำงานอัปเดทค่าแฟล็กของข้อมูลที่มีหมายเลขแพ็กเก็ตเดียวกัน

(2) หลังจากที่อัปเดทแฟล็กแล้ว จึงตรวจสอบแฟล็กว่าสามารถรีแอสเซมเบิ้ลได้หรือไม่

(3) กรณีที่แพ็กเก็ตใหม่ที่รับเข้ามาสามารถรีแอสเซมเบิ้ลได้ ให้ทำการรีแอสเซมเบิ้ล โดยเรียกฟังก์ชัน FixAndReduce ( ) ถ้าทำไม่ได้ก็ให้เพิ่มข้อมูล tupple ลงไปในบัฟเฟอร์ โดยเรียกฟังก์ชัน AddNewTupple ( ) มาทำงาน

(4) ในกรณีที่เรียกใช้ฟังก์ชัน FixAndReduce จนค่า flag = 3 และ tupple_count = 1 แสดงว่าแพ็กเก็ตที่ได้รับมาสมบูรณ์ ไม่ผิดปกติ ให้ลบข้อมูลในส่วนแพ็กเก็ตเหล่านั้นออกทั้งหมด

- void NormalCheck (int)

จุดมุ่งหมาย : ตรวจสอบว่าแพ็กเก็ตที่รับเข้ามาสามารถรีแอสเซมเบิ้ลได้หรือไม่

การทำงาน :

(1) ตรวจสอบของซ้ายของแพ็กเก็ตที่รับเข้ามาว่าเท่ากับขอบขวาของแพ็กเก็ตที่รับเข้ามาแล้วหรือไม่

(2) ตรวจสอบของขวาของแพ็กเก็ตที่รับเข้ามาว่าเท่ากับขอบซ้ายของแพ็กเก็ตที่รับเข้ามาแล้วหรือไม่

- void AbnormalCheck ( )

จุดมุ่งหมาย : ตรวจสอบว่าแพ็กเก็ตที่รับเข้ามา มีการรีแอสเซมเบิ้ลที่ผิดปกติหรือไม่ ซึ่งสามารถตรวจสอบได้ 3 แบบ คือ

(1) แพ็กเก็ตใหม่มีการซ้อนทับกับส่วนหน้าของแพ็กเก็ตเดิม

(2) แพ็กเก็ตใหม่มีการซ้อนทับกับส่วนหลังของแพ็กเก็ตเดิม

(3) แพ็กเก็ตใหม่มีการซ้อนทับกับแพ็กเก็ตเดิมทั้งแพ็กเก็ต

การทำงาน : สร้างเงื่อนไขตรวจสอบขอบซ้าย ขวาของแพ็กเก็ตใหม่ว่า มีการเหลื่อมหรือซ้อนทับกับแพ็กเก็ตเดิมหรือไม่ ถ้ามีการเหลื่อมหรือซ้อนทับกัน ก็บันทึกข้อมูลความผิดปกติไว้ใน Overlap Buffer

- void AddOverlapID ( )

จุดมุ่งหมาย : บันทึกหรืออัปเดทข้อมูลการเกิดการเหลื่อมล้ำกันของแพ็กเก็ต (Overlap)

การทำงาน : ค้นหาข้อมูลหมายเลขแพ็กเก็ตว่ามีอยู่ใน Overlap Buffer หรือไม่

(1) ถ้าไม่มี ให้เพิ่มข้อมูลการฉจมตี ซึ่งได้แก่ ID, Destination IP, count และเวลาของการเริ่มโจมตี

(2) ถ้ามีอยู่แล้ว ให้อัปเดทข้อมูล count และเวลาของการโจมตี

- void ShowResult ( )

จุดมุ่งหมาย : แสดงผลข้อมูลการโจมตีที่เกิดจากแฟร็กเมนเตชัน

การทำงาน : สามารถแสดงผลได้ 2 แบบ คือ

(1) แสดงผลออกหน้าจอ

(2) เก็บเข้าล็อกไฟล์

ขั้นตอนการทำงานจึงเริ่มจากการตรวจสอบค่า flag.log ว่าตั้งค่าไว้หรือไม่ ถ้าตั้งค่าไว้แล้ว (set) ก็เปิดล็อกไฟล์ แล้วบันทึกข้อมูลการโจมตีไว้ในไฟล์นั้น แต่ถ้ายังไม่ได้ตั้งค่าไว้ (unset) ก็ให้พิมพ์ข้อมูลออกทางหน้าจอ

- void AddNewTupple ( )

จุดมุ่งหมาย :เพิ่มข้อมูลของ tupple ลงใน Fragment Buffer

ขั้นตอนการทำงาน :

(1) ตรวจสอบใน Fragment Buffer ที่มีหมายเลขแพ็กเก็ตเดียวกับแพ็กเก็ตใหม่ ว่าจำนวน tupple มากกว่าที่กำหนดหรือไม่

(2) ถ้ามีจำนวน tupple มากเกินค่าที่กำหนด แสดงว่าแพ็กเก็ตไม่สามารถประกอบแพ็กเก็ตได้ เนื่องจากเกิดช่องว่างระหว่างแพ็กเก็ต จึงบันทึกความผิดปกตินี้ลงใน Gap Frame Buffer

(3) เพิ่มข้อมูล tupple ใหม่ลงไปใน Fragment Buffer โดยใช้ค่า lFragOff และ rFragOff ของ tupple

- void GapDetect ( )

จุดมุ่งหมาย :เก็บค่าของข้อมูลในกรณีที่ไม่สามารถประกอบแพ็กเก็ตได้ เพราะมีช่องว่างระหว่างแพ็กเก็ต

การทำงาน : ตรวจสอบใน Gap Frame Buffer ว่ามีข้อมูลที่มีหมายเลขแพ็กเก็ตเดียวกับหมายเลขแพ็กเก็ตใหม่หรือไม่

(1) ถ้าไม่มี ให้เพิ่มข้อมูลการโจมตี ซึ่งได้แก่ ID, Destination IP, count และเวลาของการเริ่มโจมตี

(2) ถ้ามี ให้ทำการอัปเดทข้อมูล count และเวลาของการโจมตี

- void FixAndReduce ( )

จุดมุ่งหมาย :เป็นกระบวนการรีแอสเซมเบิ้ล tupple ที่สามารถประกอบกันได้ เพื่อจำลองกระบวนการรีแอสเซมเบิ้ลของแพ็กเก็ต

การทำงาน :

ตรวจสอบค่า Lmerge กับ Rmerge ว่าค่าไหนมากกว่ากัน

(1) ถ้าค่า Lmerge < Rmerge ให้นำ tupple ที่ Lmerge ต่อด้วย tupple ที่ Rmerge แล้วลดค่า tupple_count

(2) ถ้าค่า Rmerge < Lmerge ให้นำ tupple ที่ Rmerge ต่อด้วย tupple ที่ Lmerge แล้วลดค่า tupple_count

5.1.2.4 BombCheck.c

โปรแกรมนี้ใช้ตรวจสอบการโจมตีในกรณีที่มีการส่งแพ็กเก็ตเข้ามายังเครื่องเป้าหมายเป็นปริมาณมากๆ โดยไม่สนใจว่าแพ็กเก็ตเหล่านั้นเป็นแพ็กเก็ตที่ผ่านการทำแฟร็กเมนเตชันมาจากแพ็กเก็ตเดียวกันหรือไม่ โดยมีฟังก์ชันการทำงานดังต่อไปนี้

- void BombCheck ( )

จุดมุ่งหมาย :เป็นการตรวจสอบการโจมตีกรณีส่งแพ็กเก็ตปริมาณมาก

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) นับจำนวนแพ็กเก็ตที่มีแอดเดรสปลายทางเดียวกัน แล้วเปรียบเทียบกับค่าที่เหมาะสมของเครือข่าย (แต่ละเครือข่ายมีค่านี้แตกต่างกันออกไป แล้วแต่ประสิทธิภาพของเครือข่าย)

(3) พิมพ์หรือเก็บในล็อกไฟล์หากมีจำนวนแพ็กเก็ตมากเกินปกติ

ผลลัพธ์ : ผลการโจมตีพิมพ์ออกที่หน้าจอหรือเก็บลงล็อกไฟล์ ว่า "LAND Attack", วัน เดือนปีที่เกิด, แ อดเดรสของเครื่องเป้าหมาย, ช่วงเวลาที่โจมตี, ปริมาณของแพ็กเก็ต

- void ShowBombResult (int min, int sec)

จุดมุ่งหมาย : ทำหน้าที่แสดงผลทางหน้าจอ หรือเก็บในล็อกไฟล์เมื่อตรวจพบการโจมตี

อาร์กิวเมนต์ : min คือ ค่านาทีที่แพ็กเก็ตเข้ามา

sec คือ ค่าวินาทีที่แพ็กเก็ตเข้ามา

การทำงาน : เมื่อปริมาณแพ็กเก็ตที่เข้าสู่เครือข่ายมีปริมาณมากกว่าปริมาณแพ็กเก็ตที่เครือข่ายยอมรับได้ ให้แจ้งเตือน หรือเก็บข้อมูลเหล่านั้นไว้ในล็อกไฟล์

5.1.2.5 HostCheck.c

เป็นการตรวจสอบชื่อโฮสต์ในไฟล์ /etc/isagnid.host ซึ่งผู้ใช้เป็นผู้กำหนดโฮสต์ที่ต้องการตรวจจับลงในไฟล์นี้ ดังนั้นโปรแกรมนี้จึงเลือกเฉพาะโฮสต์ที่ผู้ใช้กำหนดไว้ในไฟล์ดังกล่าวมาวิเคราะห์ โดยมีฟังก์ชันการทำงานดังต่อไปนี้

- void ReadHostFromFile ( )

จุดมุ่งหมาย : ทำหน้าที่ในการอ่านลิสต์ของโฮสต์จากไฟล์ลงไปไว้ในโฮสต์บัฟเฟอร์

ผลลัพธ์ : โฮสต์บัฟเฟอร์ที่เก็บค่าของโฮสต์ที่วิเคราะห์

- int SearchHost (u_long ip, int hostflag)

จุดมุ่งหมาย : เลือกข้อมูลของแพ็กเก็ตที่เข้ามาในบัฟเฟอร์เฉพาะแพ็กเก็ตที่มีไอพีตรงกับที่กำหนดไว้ เพื่อเก็บข้อมูล หรือนำมาวิเคราะห์

อาร์กิวเมนต์ : ip คือ ค่าไอพีแอดเดรสของโฮสต์ที่ต้องการตรวจจับ

hostflag เป็นค่าที่บอกว่าแพ็กเก็ตที่เข้ามาเป็นแพ็กเก็ตที่มาจากโฮสต์ที่ต้องการตรวจจับหรือไม่

ขั้นตอนการทำงาน :

(1) นำไอพีของแพ็กเก็ตที่เข้ามาในเครือข่ายไปเปรียบเทียบกับค่าไอพีในโฮสต์บัฟเฟอร์แล้วดูว่าตรงกันหรือไม่

(2) หากตรงกันจึงนำแพ็กเก็ตนั้นมาวิเคราะห์หรือเก็บข้อมูลไว้ แล้วแต่ผู้ใช้กำหนด

ผลลัพธ์ : ข้อมูลของแพ็กเก็ตหรือผลการวิเคราะห์ เฉพาะโฮสต์ที่ต้องการวิเคราะห์

5.1.2.6 IPCheck.c

ทำหน้าที่ในการนับแพ็กเก็ตของไอพี เพื่อตรวจสอบว่ามีความผิดปกติในจำนวนของแพ็กเก็ตที่ส่งหรือไม่ โดยมีฟังก์ชันการทำงานหลักๆ ดังต่อไปนี้

- void CountIPFrame (char* datafile, int logflag)

จุดมุ่งหมาย : นับจำนวนแพ็กเก็ตของไอพีในไฟล์ข้อมูล

อาร์กิวเมนต์ :char* ชื่อของไฟล์ข้อมูล

int เป็นแฟล็กที่บอกว่าให้เก็บไว้ในล็อกไฟล์หรือไม่

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) อ่านข้อมูลทีละแพ็กเก็ต แล้วนับไปเรื่อยๆ จนจบไฟล์

(3) ปิดไฟล์ข้อมูล

(4) พิมพ์หรือเก็บข้อมูลการนับลงล็อกไฟล์ตามที่แฟล็กกำหนด

ผลลัพธ์ : จำนวนแพ็กเก็ตในไฟล์ พิมพ์ออกหน้าจอ หรือเก็บลงไฟล์

- void FramePerHost (char* datafile, int logflag)

จุดมุ่งหมาย : นับจำนวนแพ็กเก็ตของไอพีแยกตามไอพีของเครื่องปลายทาง (Destination IP)

อาร์กิวเมนต์ :char* ชื่อของไฟล์ข้อมูล

int เป็นแฟล็กที่บอกว่าให้เก็บไว้ในล็อกไฟล์หรือไม่

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) อ่านข้อมูลในไฟล์ข้อมูล แล้วนับจำนวนแพ็กเก็ตตามไอพีของเครื่องปลายทาง จากนั้นแยกเก็บไว้ในบัฟเฟอร์

(3) ปิดไฟล์ข้อมูล

(4) พิมพ์หรือเก็บข้อมูลจากบัฟเฟอร์ลงล็อกไฟล์ตามที่แฟล็กกำหนด

ผลลัพธ์ : จำนวนแพ็กเก็ตของเครื่องปลายทางเดียวกันในไฟล์ พิมพ์ออกหน้าจอหรือเก็บลงไฟล์

- void FramePerMin (char* datafile, int logflag)

จุดมุ่งหมาย : นับจำนวนแพ็กเก็ตของไอพี โดยแยกเป็นนาที

อาร์กิวเมนต์ :char* ชื่อของไฟล์ข้อมูล

int เป็นแฟล็กที่บอกว่าให้เก็บไว้ในล็อกไฟล์หรือไม่

ขั้นตอนการทำงาน :

(1) เปิดไฟล์ข้อมูล

(2) ทำลูปอ่านค่าของข้อมูลในไฟล์ไปเรื่อยๆ จนกว่าจะจบไฟล์ โดยให้ในลูปตรวจสอบค่านาทีของข้อมูลที่อ่านได้ ถ้ามีการเปลี่ยนแปลงให้พิมพ์หรือเก็บลงล็อกไฟล์เป็นจำนวนแพ็กเก็ตไอพีในนาทีก่อนหน้านั้นลงไป

ผลลัพธ์ : จำนวนแพ็กเก็ตของเครื่องปลายทางเดียวกันในไฟล์ พิมพ์ออกหน้าจอหรือเก็บลงไฟล์

5.1.3 การเรียกใช้งานโปรแกรม

ในการเรียกใช้งานทั้งส่วนของการเก็บข้อมูล และการวิเคราะห์แพ็กเก็ตนั้นสามารถเรียกรวมกันด้วยโปรแกรมเดียว คือ **rmonit.c** โดยมีฟังก์ชันการทำงานต่างๆ ดังต่อไปนี้

- int main ( )

จุดมุ่งหมาย :เป็นฟังก์ชันหลักในการเรียกใช้งานฟังก์ชันอื่นๆ

ขั้นตอนการทำงาน :

(1) ตั้งค่าเริ่มต้นให้แฟล็กต่างๆ ที่ใช้งาน โดยการเรียกฟังก์ชัน InitFlag ( )

(2) ตรวจสอบค่าแฟล็กที่กำหนดมาจากคอมมานไลน์ (command line) ขึ้นอยู่กับความต้องการใช้งานของผู้ใช้ โดยเรียกใช้ฟังก์ชัน CheckFlag ( )

(3) ตรวจสอบค่าแฟล็กที่ถูกเซตจากขั้นตอนที่ (2) แล้วเรียกฟังก์ชันต่างๆ ตามที่แฟล็กกำหนดมาทำงาน (ได้แก่ ฟังก์ชันการเก็บข้อมูล และวิเคราะห์ที่ได้กล่าวไปแล้วข้างต้น)

ผลลัพธ์ : ค่าแฟล็กที่ผู้ใช้กำหนดส่งไปสั่งงานโปรแกรมอื่นๆ

- void InitFlag ( )

จุดมุ่งหมาย :ตั้งค่าเริ่มต้นของแฟล็กต่างๆ ที่ใช้งาน

- void CheckFlag ( )

จุดมุ่งหมาย : ตรวจสอบค่าแฟล็กต่างๆ ที่กำหนดมาจากคอมมานไลน์ แล้วเรียกฟังก์ชันการทำงานต่างๆ ที่ตรงตามแฟล็กที่กำหนดมาทำงาน โดยแฟล็กต่างๆ มีความสัมพันธ์กับฟังก์ชัน ดังตารางที่ 5-1

แฟล็ก	ฟังก์ชันที่เรียกใช้	การทำงาน
Count	CountIPFrame	นับจำนวนแพ็กเก็ตใน 1 ชั่วโมง
frame min	FramePerMin	นับจำนวนแพ็กเก็ตในแต่ละนาที
frame host	FramePerHost	นับจำนวนแพ็กเก็ตในแต่ละเครื่อง
land	LANDCheck	ตรวจสอบการโจมตีแบบวนลูป
MF	CheckFragment	ตรวจสอบการโจมตีแบบมีแฟร็กเมนเตชันผิดปกติ
sniff	Sniff	เก็บข้อมูลแพ็กเก็ตลงไฟล์
verbose	Sniff	พิมพ์รายละเอียดต่างๆ ของแพ็กเก็ตออกทางหน้าจอ
realtime	Sniff	วิเคราะห์แพ็กเก็ตแบบต่างๆ แบบ Real Time

ตารางที่ 5-1 แสดงแฟล็ก ฟังก์ชันที่แฟล็กเรียกใช้ และการทำงานของฟังก์ชันเหล่านั้นที่สัมพันธ์กับแฟล็ก

- void GetDateHour ( )

  จุดมุ่งหมาย : ดึงค่าชั่วโมง (Hour) และวันเดือนปี (Date) จากชื่อของไฟล์ที่เก็บข้อมูล เพื่อใช้เป็นข้อมูลที่เก็บอยู่ในล็อกไฟล์

- void PrintHowto ( )

  จุดมุ่งหมาย : แสดงวิธีการใช้งานอย่างย่อๆ โดยเรียกขึ้นมาเมื่อมีการเรียกใช้โปรแกรมไม่ถูกต้องตาม syntax ที่กำหนด

- int ParseCmdLine ( )

  จุดมุ่งหมาย : กำหนดค่าแฟล็กต่างๆ จากอาร์กิวเมนต์ที่กำหนดเข้ามาทางคอมมานไลน์

5.2 คุณสมบัติของระบบ

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น มีความสามารถในการทำงานดังต่อไปนี้

(1) เก็บข้อมูลของแพ็กเก็ต และสามารถแสดงข้อมูลของแพ็กเก็ตผ่านหน้าจอได้

(2) นับจำนวนแพ็กเก็ตที่เข้ามาระบบต่อโฮสต์ได้

(3) นับจำนวนแพ็กเก็ตที่เข้ามาระบบต่อนาทีได้

(4) วิเคราะห์แพ็กเก็ตที่เกิดความผิดปกติแบบมีปริมาณมากเกินไปได้

(5) วิเคราะห์แพ็กเก็ตที่เกิดความผิดปกติแบบมีการทำแฟร็กเมนเตชันที่ผิดปกติได้

(6) วิเคราะห์แพ็กเก็ตที่เกิดความผิดปกติแบบมีการส่งแพ็กเก็ตแบบวนลูปได้

(7) วิเคราะห์แพ็กเก็ตจากไฟล์ข้อมูลของระบบที่เก็บไว้ หรือวิเคราะห์แบบตามเวลาจริงก็ได้

(8) เปลี่ยนไดเรกทอรีที่เก็บข้อมูลได้

(9) เลือกวิเคราะห์เฉพาะเครื่องที่ต้องการได้

5.3 ข้อจำกัดของระบบ

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์สร้างขึ้นโดยมีข้อจำกัดของการใช้งาน ดังต่อไปนี้

(1) ระบบนี้สร้างขึ้นบนระบบปฏิบัติการลีนุกซ์ซึ่งอาจแตกต่างกัน หากนำไปใช้งานในระบบอื่น

(2) ผู้ที่มีสิทธ์รูท (root) เท่านั้นที่จะสามารถใช้งานโปรแกรมนี้ได้

(3) ระบบนี้สามารถตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ได้เฉพาะเครื่องปลายทางที่อยู่ในบรอดคาสโดเมน (Broadcast domain) เดียวกันเท่านั้น

(4) ในกรณีที่เครือข่ายนั้นมีแพ็กเก็ตปริมาณมาก อาจทำให้บัฟเฟอร์ที่ใช้เก็บข้อมูลของแพ็กเก็ตก่อนนำมาวิเคราะห์เต็มได้

5.4 การใช้งานระบบ

5.4.1 ฟังก์ชันการใช้งานระบบ

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้นมานี้ สามารถเรียกใช้ได้โดยการเรียก **IsagNID** โดยสามารถเลือกฟังก์ชันใช้งานเองได้ ซึ่งมีฟังก์ชันการทำงานมากมาย ได้แก่

- s เก็บข้อมูลของแพ็กเก็ตลงไฟล์
- v พิมพ์ข้อมูลของแพ็กเก็ตที่เก็บได้ออกหน้าจอ
- R ตรวจสอบการโจมตีแบบตามเวลาจริง
- f <ชื่อไฟล์> เลือกไฟล์มาวิเคราะห์

กรณีเลือก – f มักนำไปใช้ร่วมกับ options อื่นๆ ดังต่อไปนี้

- o นับจำนวนแพ็กเก็ตต่อโฮสต์ในไฟล์
- m นับจำนวนแพ็กเก็ตต่อนาทีในไฟล์
- c นับจำนวนแพ็กเก็ตไอพีในไฟล์
- M ตรวจสอบการเกิดแฟร็กเมนเตชันที่ผิดปกติในไฟล์
- l ตรวจสอบการส่งแพ็กเก็ตแบบวนลูปในไฟล์

นอกจากนั้น options ดังกล่าวข้างต้น ยังสามารถใช้งานร่วมกับ options อื่นๆ อีก ดังต่อไปนี้

- h กำหนดโฮสต์ที่ต้องการวิเคราะห์ โดยโปรแกรมจะดูชื่อโฮสต์จาก /etc/isagnid.host
- d <ชื่อไดเรกทอรี> เปลี่ยนไดเรกทอรีที่ใช้เก็บล็อกไฟล์ ซึ่งปกติอยู่ใน /var/log/isagnid
- L เก็บข้อมูลของแพ็กเก็ตที่ผิดปกติลงล็อกไฟล์

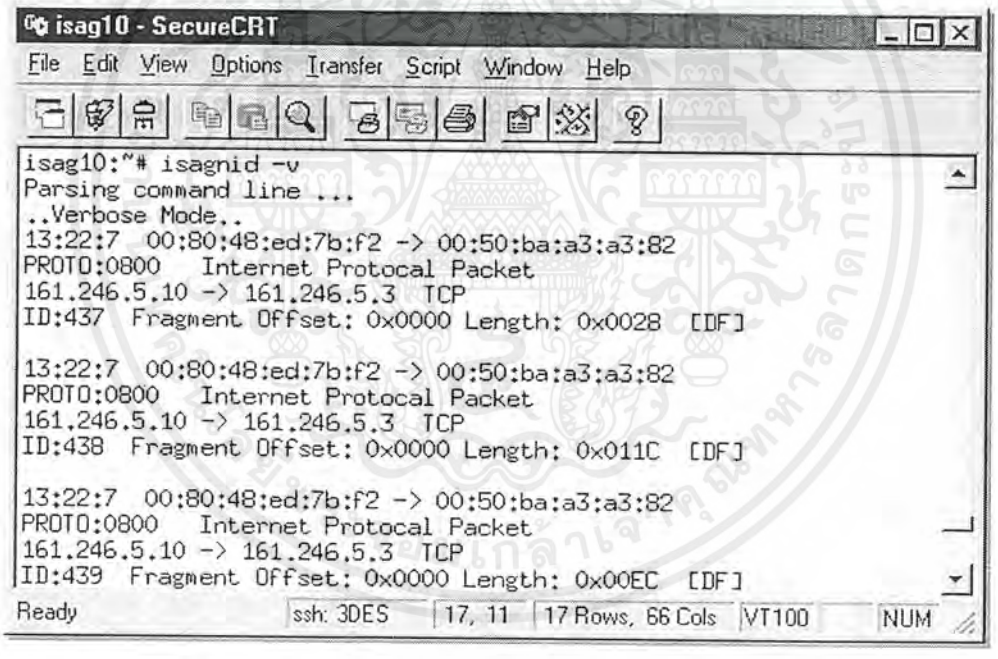
5.4.2 การเลือกใช้ฟังก์ชันการทำงาน

การทำงานของระบบแบ่งออกได้เป็น 2 ส่วนใหญ่ๆ คือ การเก็บข้อมูลของแพ็กเก็ตที่เข้าสู่ระบบ และการวิเคราะห์แพ็กเก็ตเหล่านั้น ซึ่งในการเลือกใช้ฟังก์ชันของระบบในการทำงานแต่ละส่วนนั้น ผู้ใช้สามารถเลือกใช้ได้แล้วแต่ความเหมาะสมของสภาพแวดล้อม และลักษณะของการใช้งาน ซึ่งมีข้อแนะนำในการใช้งานดังต่อไปนี้

5.4.2.1 การเก็บข้อมูล

ในขั้นตอนนี้เป็นการเก็บข้อมูลของแพ็กเก็ตที่เข้าสู่ระบบ โดยผู้ใช้อาจเลือกให้ระบบเก็บข้อมูลดังกล่าวลงล็อกไฟล์ หรือแสดงข้อมูลของแพ็กเก็ตเหล่านั้นบนหน้าจอก็ได้ โดยมีฟังก์ชันในการทำงานดังกล่าวดังต่อไปนี้

(1) กรณีต้องการดูแพ็กเก็ตโดยไม่ต้องการเก็บข้อมูลเหล่านั้นไว้ ผู้ใช้สามารถเลือกใช้ฟังก์ชัน **isagnid –v** ซึ่งเมื่อสั่งคำสั่งดังกล่าว ระบบจะแสดงข้อมูลของแพ็กเก็ตที่เข้าสู่ระบบผ่านหน้าจอเทอร์มินอล โดยแสดงข้อมูลของเฮดเดอร์ของแพ็กเก็ตทั้งของทีซีพี/ไอพี และอีเทอร์เน็ตบางส่วน รวมทั้งวันและเวลาที่แพ็กเก็ตเหล่านั้นเข้ามาด้วย ดังรูปที่ 5-1



รูปที่ 5-1 แสดงการแสดงผลการเก็บข้อมูลแพ็กเก็ตผ่านหน้าจอเทอร์มินอล

การดูข้อมูลของแพ็กเก็ตในลักษณะนี้มีข้อดีที่ไม่ต้องเปลืองเนื้อที่ในการเก็บข้อมูล แต่เป็นเพียงการเรียกดูข้อมูลอย่างเดียว ไม่ได้มีการเก็บข้อมูลของแพ็กเก็ตที่เข้ามาในระบบไว้เลย

(2) กรณีต้องการเก็บข้อมูลของแพ็กเก็ต ให้ใช้คำสั่ง **isagnid –s** ซึ่งระบบจะเก็บข้อมูลของแพ็กเก็ตลงล็อกไฟล์ โดยระบบจะจัดเก็บไฟล์เหล่านั้นเป็นรายชั่วโมง และเพื่อให้สะดวกในการเลือกข้อมูลมาวิเคราะห์ โดยล็อกไฟล์เหล่านี้จัดเก็บอยู่ในไดเรกทอรี /var/log/isagnid หรืออาจเปลี่ยนแปลงโดยเรียกใช้คำสั่ง **isagnid –d <ชื่อไดเรกทอรี>** ซึ่งมีชื่อไฟล์ตามรูปแบบดังนี้

<ชั่วโมง>_<วันที่>_<เดือน>_<ปี>

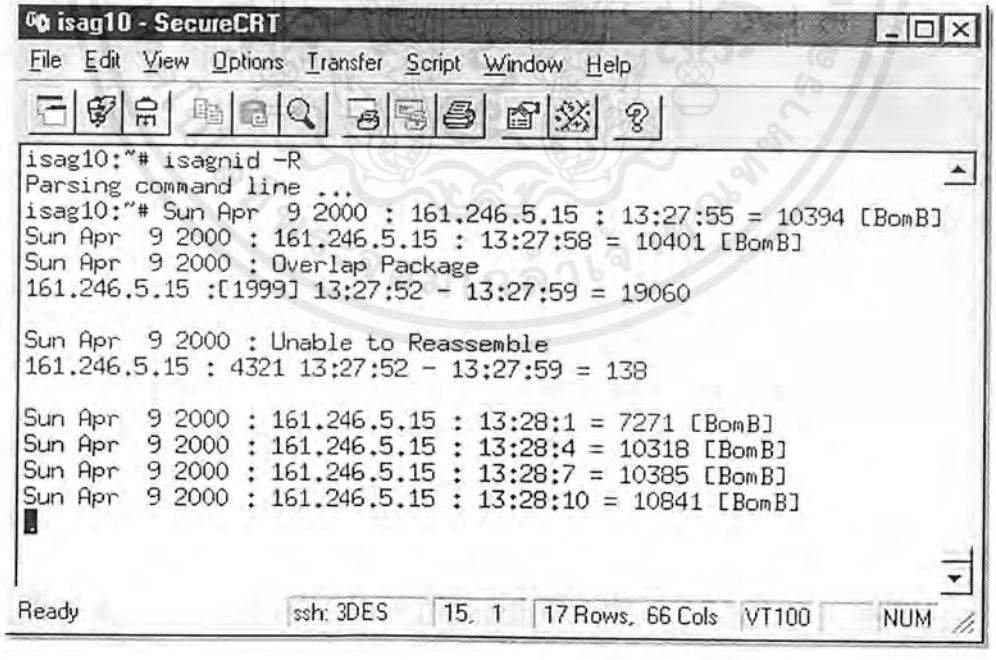
เช่น 12_09_Apr_2000 หมายถึง ไฟล์ข้อมูลแพ็กเก็ตที่เก็บไว้ในช่วงเวลา 12:00 น. ถึง 12:59 น. ของวันที่ 9 เดือนเมษายน ค.ศ. 2000 เป็นต้น แต่ในการเลือกให้เก็บข้อมูลไว้ในล็อกไฟล์ในลักษณะนี้ มีข้อเสียตรงที่เปลืองเนื้อที่ในการจัดเก็บมาก เพราะข้อมูลของแพ็กเก็ตที่เข้ามามักมีปริมาณมาก ดังนั้นหากผู้ใช้ต้องการเลือกใช้ฟังก์ชันนี้ควรคำนึงถึงเนื้อที่ในการจัดสำหรับการจัดเก็บข้อมูลด้วย

5.4.2.2 การวิเคราะห์ข้อมูล

ในขั้นตอนของการวิเคราะห์ข้อมูลนั้น สามารถเลือกให้ระบบวิเคราะห์ได้ 2 รูปแบบ ได้แก่

(1) การวิเคราะห์แบบตามเวลาจริง

ในกรณีนี้เป็นการนำข้อมูลของแพ็กเก็ตที่เข้ามาขณะนั้น มาวิเคราะห์ในขณะนั้นเลย โดยเรียกใช้คำสั่ง **isagnid –R** ทำให้ได้ข้อมูลการโจมตีตามเวลาจริงๆ ซึ่งแสดงผลออกมาผ่านหน้าจอเทอร์มินอล โดยจะแสดงผลทั้งเครื่องเป้าหมายที่ถูกโจมตี วันและช่วงเวลาที่ถูกโจมตี จำนวนแพ็กเก็ตที่โจมตีเข้ามา และชนิดของการโจมตีด้วย ดังรูปที่ 5-2



รูปที่ 5-2 แสดงหน้าจอการแสดงผลเมื่อเกิดการโจมตี

(2) การวิเคราะห์จากล็อกไฟล์

หากผู้ใช้ต้องการนำไฟล์ที่เก็บข้อมูลของแพ็กเก็ตมาวิเคราะห์ ให้เลือกไฟล์ที่ต้องการนำมาวิเคราะห์ก่อน โดยเรียกใช้คำสั่ง **isagnid –f <ชื่อไฟล์>** จากนั้นจึงเลือกฟังก์ชันต่างๆ ในการวิเคราะห์ แต่ละประเภทดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งการวิเคราะห์ในลักษณะนี้มีข้อดีที่ทำให้ผู้ใช้สามารถเลือกวิเคราะห์ข้อมูลเฉพาะช่วงเวลาที่ต้องการได้ เนื่องจากล็อกไฟล์ดังกล่าวมีการเก็บข้อมูลเป็นรายชั่วโมง

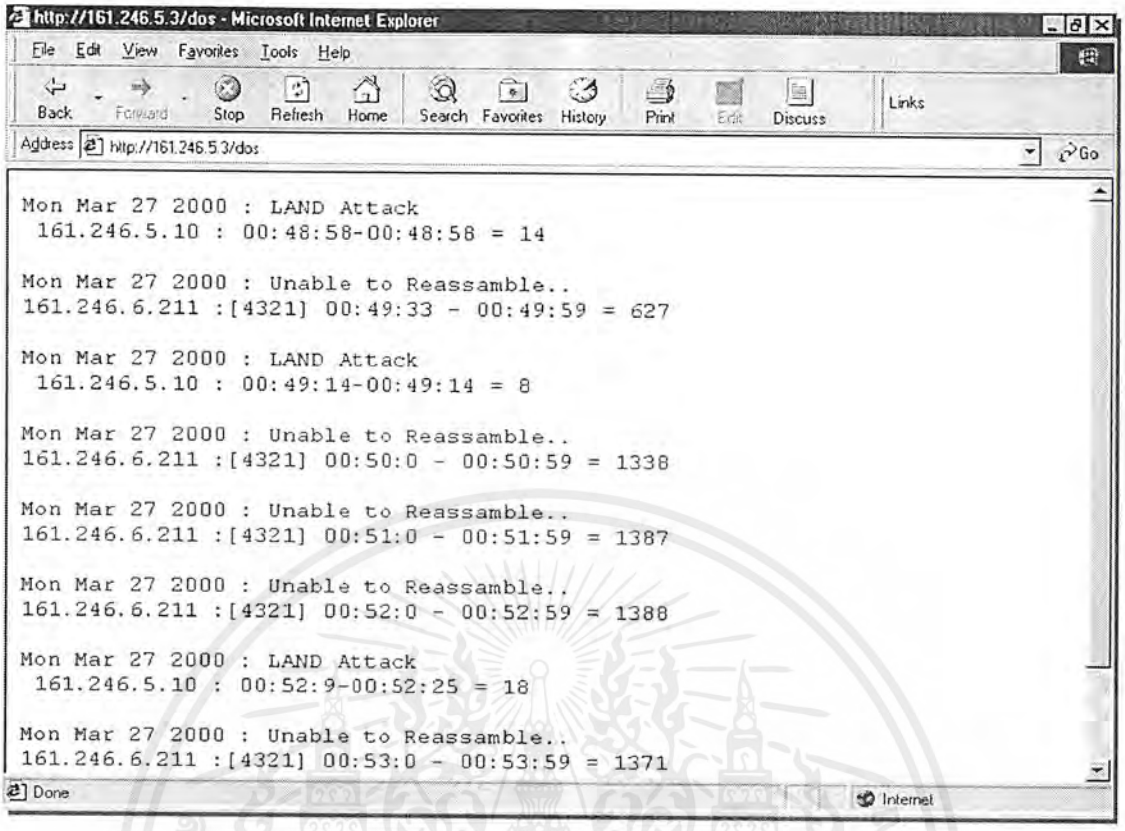
ในขั้นตอนของการวิเคราะห์ข้อมูลนั้น หากผู้ใช้ต้องการเก็บข้อมูลของความผิดปกติที่เกิดขึ้นกับระบบไว้ในล็อกไฟล์ก็สามารถเรียกใช้คำสั่ง **isagnid –L** ได้ ซึ่งล็อกไฟล์ที่ได้จะเก็บอยู่ในไดเรกทอรี /var/log/isagnid ซึ่งเก็บข้อมูลไว้ในไฟล์ชื่อ dos.log

นอกจากนั้นผู้ใช้ยังสามารถกำหนดให้โปรแกรมเลือกเก็บข้อมูล และวิเคราะห์ข้อมูลเฉพาะบางเครื่องได้ เพื่อให้สะดวกในการเลือกเครื่องที่ต้องการให้ระบบทำงาน ซึ่งสามารถกำหนดได้มากกว่าหนึ่งเครื่อง และนอกจากนั้นยังเป็นการช่วยประหยัดเนื้อที่ หากผู้ใช้เลือกให้เก็บข้อมูลของแพ็กเก็ตและความผิดปกติลงล็อกไฟล์ด้วย โดยเรียกใช้คำสั่ง **isagnid –h** โดยต้องไปกำหนดเครื่องต่างๆ ไว้ในไฟล์ /etc/isagnid.host โดยอาจกำหนดเป็นไอพี หรือชื่อเครื่องก็ได้ เช่น 161.246.5.3, isag03.ce.kmitl.ac.th เป็นต้น

นอกจากการเรียกดูข้อมูลด้วยฟังก์ชันต่างๆ ผ่านหน้าจอเทอร์มินอลแล้ว ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ยังสามารถแสดงข้อมูลการโจมตีผ่านเว็บบราวเซอร์ได้ด้วย เพื่ออำนวยความสะดวกให้แก่ผู้ดูแลระบบ ให้สามารถเรียกดูข้อมูลได้ไม่ว่าอยู่ที่ใดก็ตาม ดังรูปที่ 5-3 โดยใส่ URL ตามรูปแบบดังนี้

http://<host's IP or domain name>/dos

เช่น http://161.246.5.3 /dos ในกรณีที่ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ติดตั้งอยู่บนเครื่อง 161.246.5.3



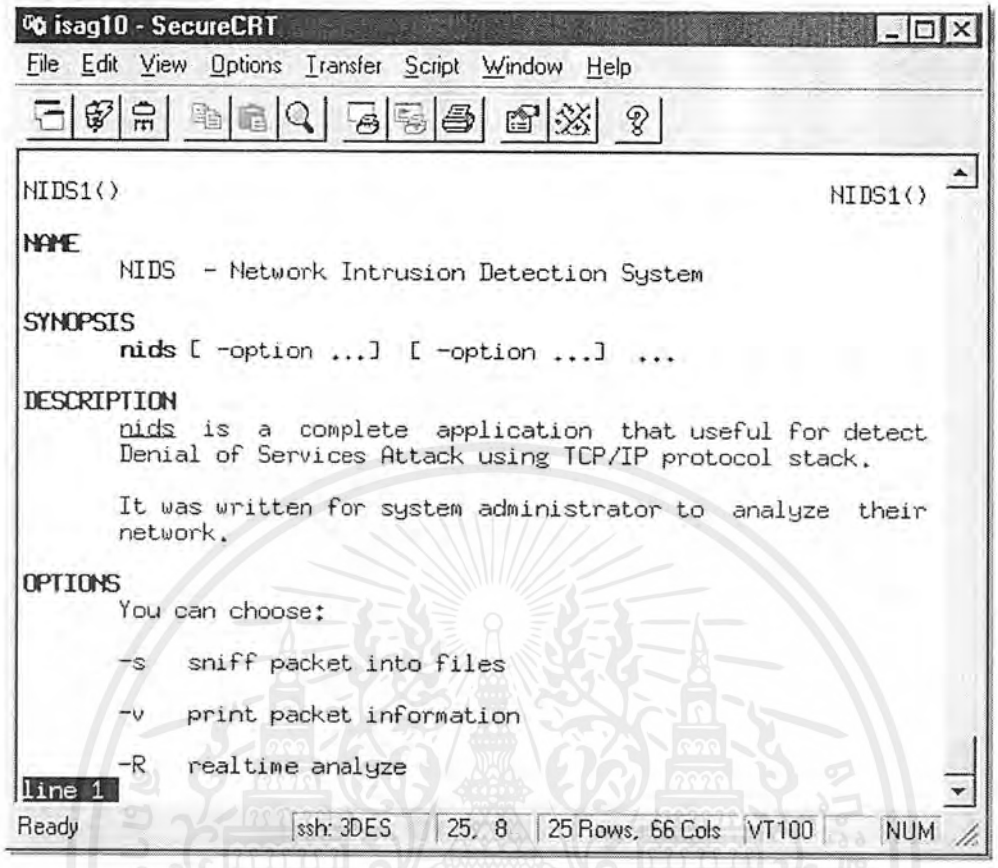
รูปที่ 5-3 แสดงการเรียกดูข้อมูลผ่านเว็บบราวเซอร์

5.4.3 การเรียกขอความช่วยเหลือจากระบบ

หากเกิดปัญหาในการใช้งานระบบยังสามารถเรียกขอความช่วยเหลือจากระบบได้ โดยการเรียกแมนเพจ (Manual Page) ขึ้นมาดู โดยใช้คำสั่ง

man isagnid

ซึ่งนำเสนอฟังก์ชันในการทำงานต่างๆ ที่ให้เลือกใช้งาน และข้อมูลต่างๆ เกี่ยวกับระบบ ดังรูปที่ 5-4



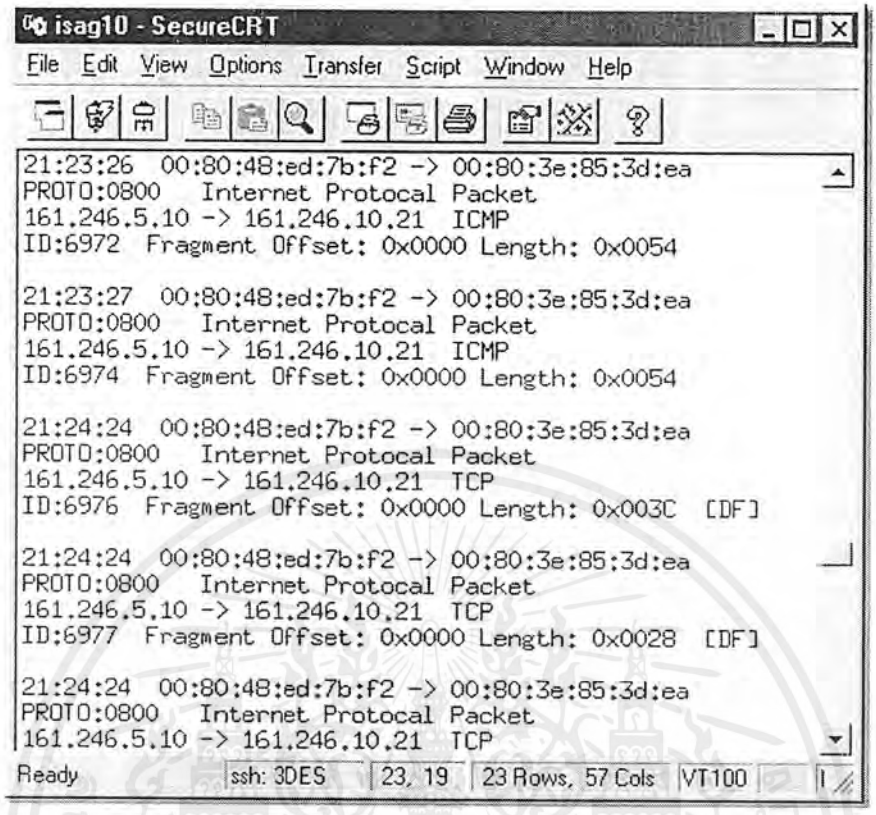
รูปที่ 5-4 แสดงการเรียกขอความช่วยเหลือจากระบบ

5.4.4 ตัวอย่างหน้าจอการทำงานของโปรแกรม

5.4.4.1 การเก็บข้อมูล

ในการเก็บข้อมูลแพ็กเก็ตนั้น ระบบจะบันทึกข้อมูลต่างๆ ดังต่อไปนี้

- เวลาที่แพ็กเก็ตนั้นเข้ามา
- แอดเดรสต้นทาง (Source Address)
- แอดเดรสปลายทาง (Destination Address)
- โพรโตคอลที่ใช้
- หมายเลขประจำแพ็กเก็ต (ID)
- แฟร็กเมนต์ออปเซต (Fragment Offset)
- ความยาวของแพ็กเก็ต (Length)



รูปที่ 5-5 แสดงการแสดงผลข้อมูลของแพ็กเก็ตที่เข้าสู่ระบบ

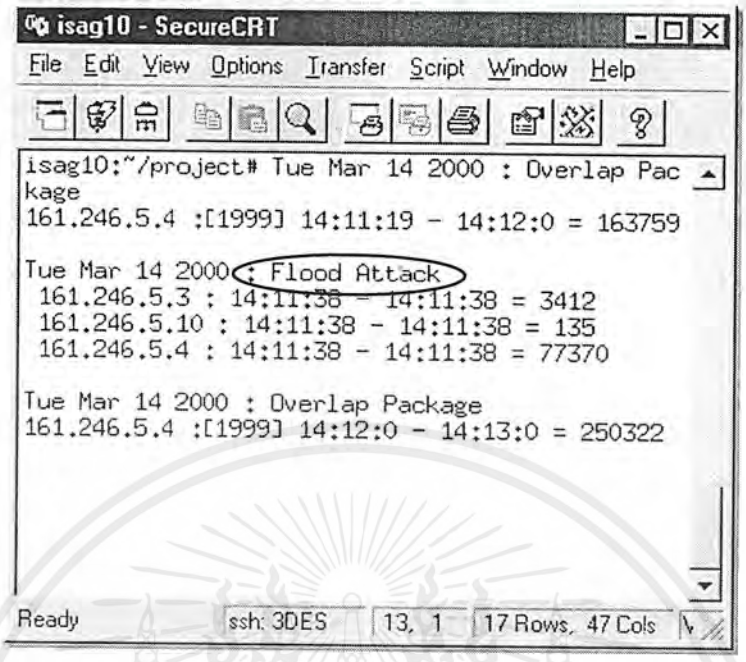
5.4.4.2 การวิเคราะห์ข้อมูล

ในการวิเคราะห์ข้อมูลนั้น หากพบว่าแพ็กเก็ตเกิดความผิดปกติ ก็แจ้งเตือนออกทางหน้าจอ แล้วเก็บไว้เป็นล็อกไฟล์ ซึ่งมีข้อมูลที่แสดงและจัดเก็บ ดังต่อไปนี้

- วัน เดือน ปี ที่แพ็กเก็ตเหล่านั้นเข้ามา
- ชนิดของการโจมตี
- แอดเดรสปลายทาง (Destination Address)
- ช่วงเวลาที่แพ็กเก็ตเหล่านั้นเข้ามา
- จำนวนแพ็กเก็ตที่เข้ามา

ซึ่งผลการวิเคราะห์สามารถแยกวิเคราะห์ได้ตามแต่ละประเภท หรือกำหนดให้วิเคราะห์รวมกันทุกประเภทก็ได้ ดังตัวอย่างต่อไปนี้

(1) การส่งแพ็กเก็ตปริมาณมาก

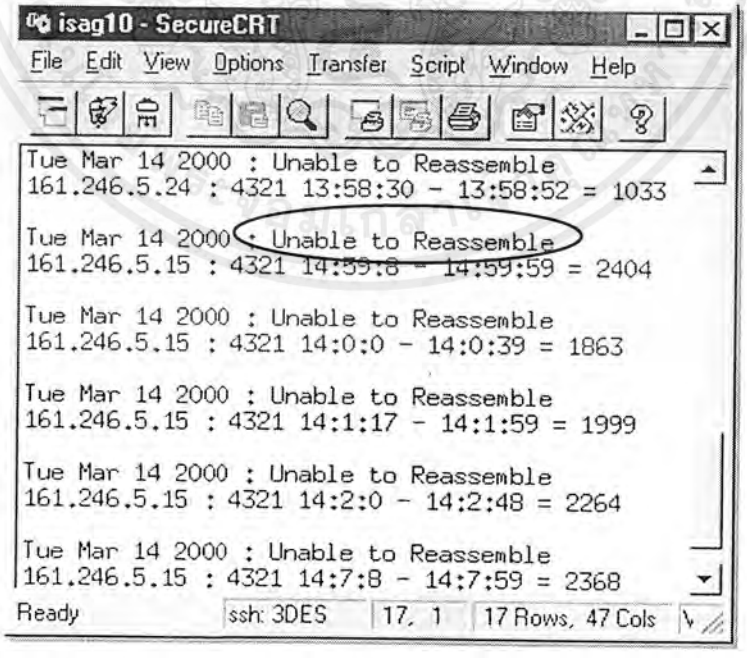


รูปที่ 5-6 แสดงการแจ้งเตือนเมื่อตรวจพบการโจมตีแบบแพ็กเก็ตปริมาณมาก

(2) แฟร็กเมนเตชันผิดปกติ

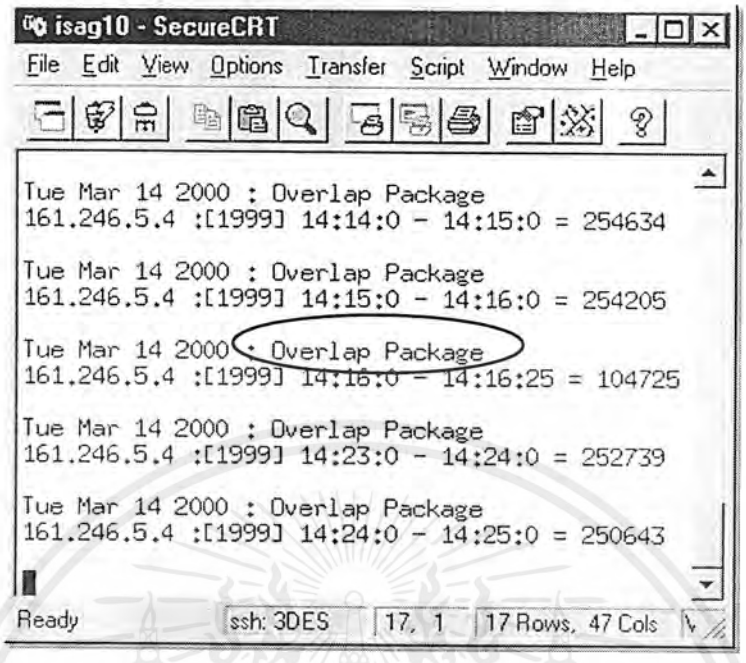
ซึ่งแบ่งออกได้ดังต่อไปนี้

- การส่งแพ็กเก็ตที่มีลำดับผิดปกติ



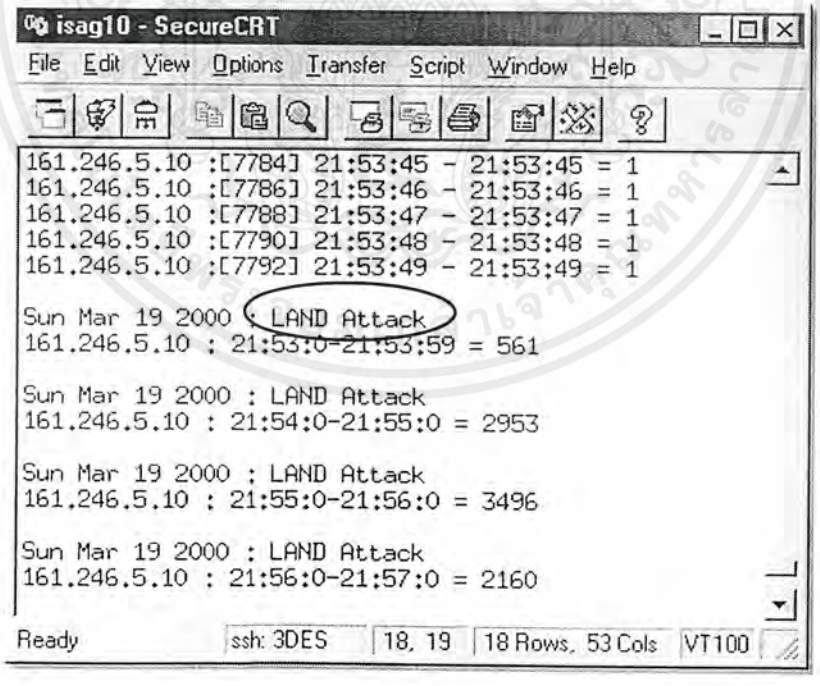
รูปที่ 5-7 แสดงการแจ้งเตือนเมื่อไม่สามารถประกอบแพ็กเก็ตได้

- การส่งแพ็กเก็ตที่มีขนาดเหลื่อมล้ำกัน



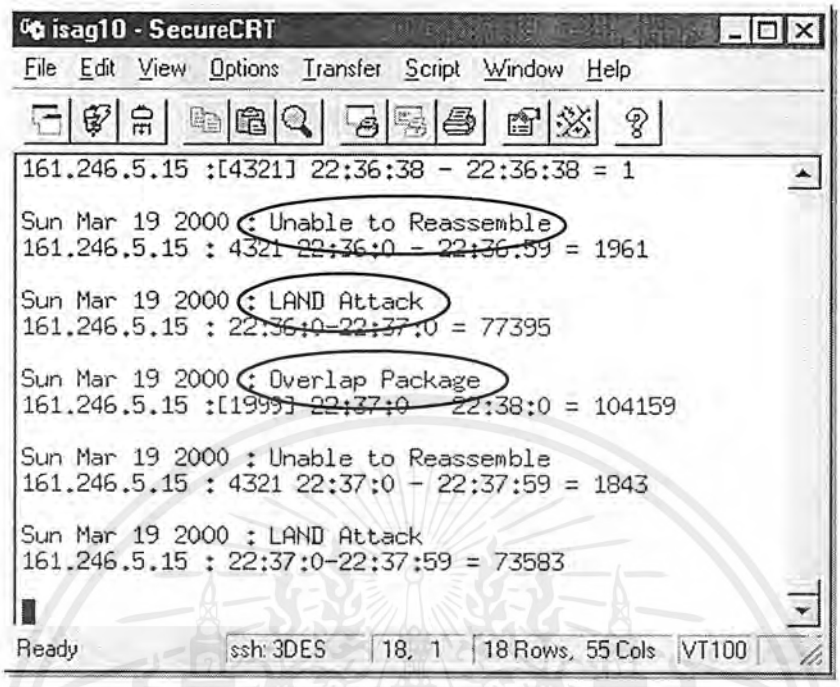
รูปที่ 5-8 แสดงการแจ้งเตือนเมื่อมีการเหลื่อมล้ำกันของแพ็กเก็ต

- การส่งแพ็กเก็ตแบบวนลูป



รูปที่ 5-9 แสดงการแจ้งเตือนเมื่อมีการส่งแพ็กเก็ตแบบวนลูป

- การส่งแพ็กเก็ตแบบผสมกันหลายแบบ



รูปที่ 5-10 แสดงการแจ้งเตือนเมื่อมีแพ็กเก็ตผิดปกติแบบผสม

5.5 จุดเด่นและจุดบกพร่องในการทำงานของระบบ

5.5.1 จุดเด่น

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้นนี้มีลักษณะเด่น ดังต่อไปนี้

(1) สามารถตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่อาศัยช่องทางของโพรโตคอลในชั้นอินเทอร์เน็ต และทรานสปอร์ตได้เกือบทุกรูปแบบ

(2) มีฟังก์ชันการทำงานให้เลือกใช้งานหลากหลาย สามารถเลือกเก็บข้อมูล แยกหรือรวมกับการวิเคราะห์ก็ได้

(3) สามารถนำแนวทางต่อไปนี้ไปพัฒนาต่อได้หลากหลาย

5.5.2 จุดบกพร่อง

ระบบที่สร้างขึ้นยังมีจุดบกพร่องอยู่บ้าง ได้แก่

(1) การเก็บข้อมูล โดยเรียกฟังก์ชัน –s ซึ่งเป็นการเก็บข้อมูลของแพ็กเก็ตทั้งหมดลงไฟล์ ดังนั้นหากมีการเก็บข้อมูลของแพ็กเก็ตนานๆ หรือเกิดการโจมตีเข้ามา อาจทำให้ฮาร์ดดิสก์ไม่เพียงพอในการเก็บข้อมูลได้

(2) การป้อนพารามิเตอร์ให้ระบบทำงาน หากป้อนเยอะมาก ระบบอาจสับสนได้

บทที่ 6
สรุปและวิจารณ์

6.1 ปัญหาและอุปสรรค

ในการพัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์นี้ มีปัญหาและอุปสรรคในการพัฒนาหลายประการ ได้แก่

(1) โปรแกรมที่ใช้ในการโจมตีเพื่อให้ปิดบริการมีมากมาย การศึกษาการทำงานของโปรแกรมเหล่านั้นและการนำมาจัดแบ่งประเภท ทำได้ยาก และใช้เวลานาน

(2) ยังไม่สามารถตรวจสอบการโจมตีในชั้นแอพพลิเคชันได้

6.2 แนวทางการวิจัยและพัฒนาต่อ

เนื่องจากระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้นมานี้ สามารถตรวจจับผู้บุกรุกได้แค่ระดับที่ทราบว่ามีการโจมตีระบบ โดยการส่งแพ็กเก็ตในชั้นอินเทอร์เน็ตและทรานสปอร์ต และเป็นระบบที่สร้างขึ้นเพื่อใช้งานบนระบบปฏิบัติการลีนุกซ์เท่านั้น ดังนั้นจึงสามารถนำไปพัฒนาต่อได้โดย

(1) สร้างระบบป้องกันขึ้นด้วย กล่าวคือ เมื่อมีการแจ้งเตือนสามารถกันไม่ได้รับแพ็กเก็ตที่มาจากไอพีต้นทาง (Source IP) หรือแม็คแอดเดรสต้นทาง (Source MAC) นั้นได้ โดยให้ไปตั้งค่าที่ไฟล์วอล์ล (Firewall) หรือสวิตช์ชิ่ง (Switching) ให้โดยอัตโนมัติ

(2) ทำให้สามารถตรวจสอบได้ว่าผู้บุกรุกเป็นใคร โดยเมื่อมีความผิดปกติเกิดขึ้นให้ Trace กลับไปยังเครื่องที่โจมตีมาได้

(3) เพิ่มความละเอียดของการวิเคราะห์ไปถึงชั้นเน็ตเวิร์กอินเตอร์เฟส คือ มีการนำแม็คแอดเดรสมาใช้ด้วย

(4) เพิ่มความสามารถในการตรวจจับให้มากขึ้นได้ โดยการเพิ่มขอบเขตการตรวจจับให้กว้างขึ้น (ทำเป็น Distributed)

(5) สร้างให้ระบบสามารถทำงานบนแพล็ตฟอร์มอื่นได้ เช่น Windows 9x, Windows NT, Windows 2000 เป็นต้น

6.3 เปรียบเทียบระบบกับผลิตภัณฑ์อื่นที่มีอยู่ในตลาด

ปัจจุบันผลิตภัณฑ์ในท้องตลาดที่ออกมาสนับสนุนการตรวจจับการโจมตีในลักษณะนี้เท่าที่พบมี 4 ผลิตภัณฑ์ ได้แก่ Abinet, IBM, Internet Security Systems และ NFR/Anzen ซึ่งถึงแม้ว่ามีขอบเขตไม่เหมือนระบบตรวจจับผู้บุกรุกที่พัฒนาขึ้นมานี้ แต่ก็มีลักษณะการตรวจจับที่คล้ายกัน แต่ผลิตภัณฑ์ทั้งสี่นี้ยังไม่สามารถตรวจจับการโจมตีในบางประเภทได้ ซึ่งสามารถเปรียบเทียบกับระบบที่สร้างขึ้นได้ดังตารางที่ 6-1

โปรแกรมโจมตี	ผลการทดสอบ				
	Abirnet	IBM	Internet Security Systems	NFR/Anzen	IsagNID
1. Sendmail	✗	✗	✓	✓	✗
2. Port Scan 2	✓	✗	✓	✓	✗
3. Satan	❍	✓	✓	✓	✗
4. SYN Flood	✓	❍	❍	✓	✓
5. PING of Death	✓	✓	✓	✓	✓
6. Land	✓	❍	✓	✓	✓
7. teardrop	✓	✓	✗	✓	✓
8. oshare	✗	✗	✗	❍	✓

ตารางที่ 6-1 แสดงการเปรียบเทียบระบบกับผลิตภัณฑ์อื่น

✓ คือ สามารถตรวจจับว่ามีการโจมตีอย่างถูกต้อง

✗ คือ ไม่สามารถตรวจจับได้

❍ คือ ตรวจจับได้ว่าเกิดการโจมตี แต่ไม่สามารถบ่งชี้ได้ว่าเป็นการโจมตีแบบใด

จากตัวอย่างของการทดสอบระบบตรวจจับผู้บุกรุกต่างๆ ด้วยโปรแกรมโจมตีแบบต่างๆ ปรากฏว่าระบบตรวจจับผู้บุกรุกส่วนใหญ่ไม่สามารถตรวจจับการโจมตีในรูปแบบของ oshare ได้ หรือได้ แต่ไม่สามารถระบุชนิดของการโจมตีได้อย่างถูกต้อง เนื่องจากการโจมตีของ oshare นั้นเป็นแบบการทำแฟร็กเมนเตชันที่ผิดปกติ โดยการส่งแพ็กเก็ตแบบเดียวกันมา ทำให้ไม่สามารถรีแอสเซมเบิ้ลแพ็กเก็ตเหล่านั้นได้

แต่เห็นได้อีกว่าผลิตภัณฑ์ส่วนใหญ่ที่สร้างขึ้น มีขอบเขตการทำงานกว้างกว่าระบบที่พัฒนาขึ้นมาใหม่นี้ ทั้งนี้ดูได้จากการตรวจสอบการโจมตีในชั้นแอพพลิเคชัน เช่น Sendmail, Port Scan 2, Satan เป็นต้น ซึ่งเป็นการโจมตีที่โฮสต์ในชั้นแอพพลิเคชัน ซึ่งอยู่นอกขอบเขตของระบบตรวจจับผู้บุกรุกที่พัฒนาขึ้นมา

บรรณานุกรม

หนังสืออ้างอิง

[1] Mark A. Miller, P.E., *"Troubleshooting TCP/IP Analyzing the Protocols of the Internet"*, M&T Books, 1993, pp. 121-215

[2] William Stallings, *"Data and Computer Communications"*, 5th Edition, Prentice Hall, 1997, pp. 497-526, 585-619

[3] Neil Matthew, Richard Stones, *"Beginning Linux Programming"*, Wrox Press, 1996

[4] Lowell Jay Arthur, Ted Burns, *"UNIX Shell Programming"*, 4th Edition, John Wiley & Sons, 1997

[5] สันติ ศรีลาศักดิ์, วรวุฒิ เที่ยงธรรม, *"เจาะประเด็นงานเขียนโปรแกรมบนลีนุกซ์"*, ออฟเซ็ท เพรส, 2542

[6] สุวัฒน์ ปุณณชัยยะ, ต้น ตันฑ์สุทธิวงศ์, สุพจน์ ปุณณชัยชนะ, *"เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต"*, โปรวิชั่น, 2543

เว็บไซต์อ้างอิง

[1] http://www.securityfocus.com

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

[3] http://www.idg.net/crd_detection_16738.html

[4] http://archive.infoworld.com/pageone/gif/980504iwss16.gif

[5] http://www.abirnet.com/sw3intro.html

[6] http://www-1.ibm.com/services/continuity/recover1.nsf/ers/Home