โปรแกรมตรวจสอบและวิเคราะห์เครือข่ายระดับทีซีพีบนระบบยูนิกซ์

TCP Monitoring and Analysis Tool for Unix System

นาย ขจรศักดิ์ ทีจันทึก

นาย สุรพงศ์ เลิศวิวัฒน์กุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขา วิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

โปรแกรมตรวจสอบและวิเคราะห์เครือข่ายระดับทีซีพีบนระบบยูนิกซ์

TCP Monitoring and Analysis Tool for Unix System

โดย

นายขจรศักดิ์ ทีจันทึก

นายสุรพงศ์ เลิศวิวัฒน์กุล

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์สุมณฑา หลิมศิริวงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขา วิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมตรวจสอบและวิเคราะห์เครือข่ายระดับทีซีพีบนระบบยูนิกซ์

TCP Monitoring and Analysis Tool for Unix System

ผู้จัดทำ

1. นายขจรศักดิ์ ทีจันทึก รหัส 39013231

2. นายสุรพงศ์ เลิศวิวัฒน์กุล รหัส 39013258

_____ อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์สุมณฑา หลิมศิริวงษ์)

โปรแกรมตรวจสอบและวิเคราะห์เครือข่ายระดับทีซีพีบนระบบยูนิกซ์

ขจรศักดิ์ ทีจันทึก
สุรพงศ์ เลิศวิวัฒน์กุล
อ. ธนา หงษ์สุวรรณ
อ. สุมณฑา หลิมศิริวงษ์
ปีการศึกษา 2541

บทคัดย่อ

หน้าที่ของผู้ดูแลระบบคอมพิวเตอร์คือ การตรวจสอบการทำงานของระบบและโปรแกรมต่างๆ ให้อยู่ในสภาพที่ถูกต้องและทำงานอย่างสอดคล้องกัน หากมีการบุกรุกเข้าสู่ระบบ ถ้าการบุกรุกนั้นเข้ามาที่เครื่องเซิร์ฟเวอร์ ผู้ดูแลระบบสามารถตรวจสอบได้โดยการพิจารณาจาก ล็อกไฟล์ของระบบ หรือ หากการบุกรุกกระทำในระบบเครือข่าย ผู้ดูแลระบบสามารถตรวจสอบความผิดปกติได้จาก ลักษณะการติดต่อทางเครือข่าย โดยมีเครื่องมือที่ช่วยในการตรวจสอบที่ดีตัวหนึ่งชื่อว่า "tcpdump" ซึ่งทำหน้าที่ตรวจจับแพ็กเกจที่อยู่ในเครือข่าย แต่ข้อมูลที่ได้จากโปรแกรมนี้เป็นข้อมูลทางเทคนิค ซึ่งต้องอาศัยความรู้และความชำนาญในการวิเคราะห์มาก โครงงานนี้ได้พัฒนาโปรแกรมที่ทำหน้าที่ วิเคราะห์แพ็กเกจ TCP/IP ที่จับได้ในระบบ เครือข่าย พร้อมทั้งเสนอวิธีในการตรวจจับการบุกรุกทางเครือข่ายที่เป็นที่รู้จัก เช่น TCP SYN Flooding, Scan Port, และ IP- Spoofing และรายงานผลการตรวจจับและวิเคราะห์ผ่านทางเว็บเพจ

TCP Monitoring and Analysis Tool for Unix System

Mr.Kajornsak Teejantuek
Mr.Suraphong Lahtwiwatkul

Mr.Thana Hongsuwan Advisor
Miss.Sumonta Linsiriwong Advisor

ABSTRACT

The response of the network system administrator is to inspect the operation of the system to ensure that is working correctly. The administrator can detect the intrusion access into the fileserver by inspecting the log file, however if the intrusion access is not intend to the fileserver, the administrator can detect this intrusion by using analysis tools such as "tcpdump" which is a software for detect the packet in the network. Because of the derived data from tcpdump is in a technical term which must be translate into the readable document. This project is the development of program to analysis the derived TCP/IP packet and propose the ways to detect the intruding into the network system such as TCP SYN Flooding, Scan Port, and IP-Spoofing. The program allow the user to analysis the netwoke packet via web browser, this mean that the administrator can analysis the network easier and faster.

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญภาพประกอบ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญ และที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตงานของปริญญานิพนธ์	1
1.4 วิธีการดำเนินการ	1
บทที่ 2 โพรโตคอลทีซีพีไอพี (TCP/IP Protocol)	3
2.1 ทีซีพีไอพี (TCP/IP)	3
2.2 มาตรฐานระบบการสื่อสาร	3
2.3 โครงสร้างของระบบส่งข้อมูล	6
2.4 การทำงานของทีซีพีตามชั้น	7
2.5 แพ็กเกจของไอพี	8
บทที่ 3 tcpdump	10
3.1 tcpdump คืออะไร?	10
3.2 การใช้งานโปรแกรม tcpdump	11
3.2.1 OPTION ของโปรแกรม tcpdump	11
3.2.2 EXPRESSIONของโปรแกรม tcpdump	12
3.3 ตัวอย่างการใช้งานและผลลัพธ์ของโปรแกรม tcpdump	14
3.3.1 เวลา (Time Stamp)	15
3.3.2 ลิงค์เลเยอร์	15
3.3.3 เน็ตเวิร์คเลเยอร์	16
3.3.4 ทรานสปอร์ตเลเยอร์	17
บทที่ 4 CGI และ ภาษาเพริ์ล (Perl Language)	19
4.1 การเลือกระบบปฏิบัติการที่จะใช้เขียนสคริปต์	19
4.1.1 งานเกี่ยวกับข้อความและการค้นหา	19
4.1.2 งานในการประมวลผลข้อมูลที่ไม่ใช่ข้อความ	20
4.1.3 งานที่เกี่ยวกับการประมวลผลฐานข้อมูล	20

สารบัญ(ต่อ)

	หน้าที่
4.2 หลักทั่วไปในการเขียนโปรแกรมซีจีไอ	20
4.2.1 การรับข้อมูลจากไคลเอนต์	21
4.2.2 การแยกข้อมูลจากไคลเอนต์	21
4.2.3 การส่งผลลัพธ์กลับไปยังไคลเอนต์	21
4..3 CGI สคริปต์ทำงานอย่างไร	22
4.4 ภาษา Perl	24
บทที่ 5 การโจมตีเครือข่าย	27
5.1 TCP SYN Flooding	27
5.1.1 การทำงานของ TCP SYN Flooding	27
5.1.2 คิวของการรับการขอติดต่อ	28
5.1.3 ผลกระทบจาก TCP SYN Flooding	30
5.2. Scan Port	30
5.2.1 การทำงานของ Scan Port	30
5.3 IP-spoofing	31
บทที่ 6 การออกแบบและการสร้าง	32
6.1 การบันทึกการติดต่อในเครือข่าย	32
6.1.1 หลักการและการออกแบบการบันทึกการติดต่อภายในเครือข่าย	34
6.1.2 การทำงานของโปรแกรมที่บันทึกการปิดการติดต่อภายในเครือข่าย	36
6.1.3 การทำงานของโปรแกรมที่บันทึกการเปิดการติดต่อภายในเครือข่าย	38
6.2 การวิเคราะห์และตรวจจับการบุกรุกภายในเครือข่าย	40
6.2.1 การตรวจจับ TCP SYN Flooding	41
6.2.2 การตรวจจับสแกนพอร์ต	44
6.2.3 การตรวจจับการปลอมไอพี (IP-spoofing)	46
6.3 การทำงานของโปรแกรมในส่วน CGI	48
6.3.1 การทำงานของโปรแกรมที่ทำหน้าที่แสดงผลการติดต่อที่ปิดแล้ว	48
6.3.2 การทำงานของโปรแกรมที่ทำหน้าที่แสดงผลการติดต่อที่ยังเปิดใช้อยู่	49
6.3.3 การทำงานของโปรแกรมที่ทำหน้าที่แสดงผลการตรวจจับการบุกรุก	50
บทที่ 7 ตัวอย่างและการใช้งานโปรแกรม	51
7.1 การสอบถามข้อมูลของการติดต่อที่เกิดขึ้นภายในเครือข่าย	52
7.1.1 การสอบถามข้อมูลของการติดต่อที่เปิดอยู่	52

สารบัญ(ต่อ)

หน้าที่

7.1.2 การสอบถามข้อมูลของการติดต่อที่ปิดไปแล้ว 54

7.2 การรายงานผลการตรวจจับการบุกรุก 58

7.2.1 รายละเอียดของการตรวจพบสแกนพอร์ต 59

7.2.2 รายละเอียดของการตรวจพบ TCP SYN Flooding 59

7.2.3 รายละเอียดของการตรวจพบ IP-Spoofing 60

บทที่ 8 สรุปและวิจารณ์ 61

8.1 ปัญหาและอุปสรรค 61

8.2 แนวทางการวิจัยต่อ 61

กิตติกรรมประกาศ 62

บรรณานุกรม 63

สารบัญภาพประกอบ

หน้าที่

รูปที่ 2-1 Data Encapsulation 5
รูปที่ 2-2 แสดงชั้นการทำงานของทีซีพี 7
รูปที่ 2-3 แสดงรูปแบบของไอพี แพ็กเกจ 8
รูปที่ 3-1 แสดงความหมายของแพ็กเกจที่ได้จากโปรแกรม tcpdump 18
รูปที่ 5-1 แสดงการจัดการคิวของทีซีพีและแอปพลิเคชั่น แยกกัน 29
รูปที่ 5-2 แสดงการ Scan Port แบบ Half Scan 30
รูปที่ 5-3 แสดงการ Scan Port แบบ Full 30
รูปที่ 6-1 แสดงการติดต่อที่ใช้ทีซีพีโพรโตคอล 32
รูปที่ 6-2 แสดงการทำงานการบันทึกการติดต่อเครือข่ายเพื่อบันทึกและการปิดการติดต่อ 35
รูปที่ 6-3 แสดงผังการทำงานของโปรแกรมบันทึกการปิดติดต่อภายเครือข่าย 36
รูปที่ 6-4 แสดงผังการทำงานของโปรแกรมบันทึกการเปิดติดต่อภายเครือข่าย 38
รูปที่ 6-5 แสดงการไหลของข้อมูลในส่วนของการตรวจจับการบุกรุก 40
รูปที่ 6-6 แสดงการกำหนด State ที่จะใช้ในโปรแกรม 41
รูปที่ 6-7 แสดงผังการทำงานของโปรแกรมตรวจจับ TCP SYN Flooding 42
รูปที่ 6-8 แสดงผังการทำงานของโปแกรมตรวจจับสแกนพอร์ต 46
รูปที่ 6-9 แสดงผังการทำงานของโปรแกรมตรวจจับการปลอมไอพี 47
รูปที่ 7-1 แสดง Home Page ที่แสดงหน้าหลักของงาน 51
รูปที่ 7-2 แสดงการเลือกเพื่อจะดูการติดต่อภายในเครือข่าย 52
รูปที่ 7-3 แสดงผลของการเลือกดูผลของการติดต่อที่ยังค้างอยู่ในปัจจุบัน 53
รูปที่ 7-4 แสดงการระบุข้อมูลเพื่อเลือกโฮสต์และระบุบริการที่ต้องการ 53
รูปที่ 7-5 แสดงของการติดต่อในปัจจุบันของโฮสต์นั้น 54
รูปที่ 7-6 แสดงการติดต่อที่ปิดไปแล้ว 55
รูปที่ 7-7 แสดงการกรอกข้อมูลเพื่อทำการสอบถามการติดต่อที่ปิดไปแล้ว 55
รูปที่ 7-8 แสดงผลของการติดต่อที่ปิดไปแล้วจากรูปที่ 7-7 56
รูปที่ 7-9 แสดงผลของการติดต่อที่ปิดไปแล้ว 57
รูปที่ 7-10 แสดงรายงานแสดงการโจมตีที่ตรวจพบ 58
รูปที่ 7-11 แสดงรายละเอียดของการตรวจพบการสแกนพอร์ต 59
รูปที่ 7-12 แสดงการละเอียดของการตรวจพบ TCP SYN Flooding 59
รูปที่ 7-13 แสดงรายละเอียดของการตรวจพบ IP-Spoofing 60

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพีไอพี 3

ตารางที่ 2-2 แสดงบริการและหมายเลขพอร์ตของทีซีพีโพรโตคอล 4

ตารางที่ 2-3 แสดงชื่อบริการและหมายเลขพอร์ตของยูดีพีโพรโตคอล 5

ตารางที่ 5-1 แสดงขนาดของ backlog คิว 29

ตารางที่ 6-1 แสดงการทำงานของโปรแกรมบันทึกการติดต่อตามเฟล็กของแพ็กเกจ 34

บทที่ 1
บทนำ

1.1 ความสำคัญ และที่มา

ในการใช้งานคอมพิวเตอร์ในสภาพแวดล้อมที่เป็นเครือข่ายนั้นแตกต่างจากการใช้งานในสภาพแวดล้อมที่มีผู้ใช้เพียงคนเดียว สิ่งที่ผู้ดูแลระบบต้องคำนึงถึงคือ ประสิทธิภาพของเครือข่าย ความถูกต้องแน่นอนของข้อมูล การป้องกันผู้บุกรุกระบบ ในปริญญานิพนธ์นี้ คือ การพัฒนาแอปพลิเคชั่นเพื่อใช้ประกอบการเรียนรู้กระบวนการส่งข้อมูลผ่านเครือข่าย โดยสามารถติดต่อดูแพ็กเกจที่วิ่งอยู่ในเครือข่าย ตรวจสอบถึงสาเหตุต่างๆ ที่เกิดขึ้นจากข้อมูลต่างๆ ในเครือข่าย โดยข้อมูลที่ได้สามารถนำไปใช้ เช่น การบันทึกการสนทนาระหว่างเครื่อง วิเคราะห์ปริมาณการใช้งาน วิเคราะห์ประสิทธิภาพการใช้งานเครือข่าย และการตรวจจับการบุกรุกเข้ามาในเครือข่าย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 เพื่อศึกษาการทำงานของเครือข่าย เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับเครือข่าย

1.2.2 เพื่อศึกษาเครื่องมือและวิธีการ ที่จะทำให้สามารถตรวจสอบเครือข่ายได้

1.2.3 เพื่อศึกษาการจราจรของข้อมูลในเครือข่าย และการส่งข้อมูลภายใต้โพรโตคอลต่างๆ ในเครือข่าย

1.2.4 เพื่อบันทึกการติดต่อที่เกิดขึ้นในเครือข่าย และสามารถดูผลทางเว็บเพจได้

1.2.5 เพื่อตรวจจับการบุกรุก โดยใช้ TCP SYN Flooding, สแกนพอร์ต (Scan Port), การปลอมไอพี (IP-spoofing) ได้และรายงานผลทางเว็บเพจได้

1.3 ขอบเขตงานของปริญญานิพนธ์

ปริญญานิพนธ์นี้เป็น การสร้างซอฟต์แวร์เพื่อช่วยในการดูแลเครือข่าย โดยมีการทำงานดังนี้

1.3.1 การบันทึกการติดต่อภายในเครือข่าย นั่นคือ มีการบันทึกการติดต่อที่เกิดขึ้นภายในเครือข่าย โดยสนใจเฉพาะการทำงานของแอปพลิเคชั่นที่ใช้โพรโตคอลทีซีพี (TCP Protocol)

1.3.2 การตรวจจับ การโจมตีเครือข่ายโดยใช้ TCP SYN Flooding,การสแกนพอร์ต, และการปลอมไอพี

1.3.3 การสอบถามข้อมูลและการแสดงผลออกทางเว็บเพจ

1.4 วิธีการดำเนินการ

งานในโครงงานนี้จะเริ่มจากการศึกษาทฤษฎีพื้นฐานที่เกี่ยวข้องกับโครงงาน ซึ่งได้แก่การศึกษาเรื่องหลักๆ ดังนี้คือศึกษาโพรโตคอลทีซีพีไอพี (บทที่ 2), ศึกษาโปรแกรม tcpdump เพื่อใช้ในการเก็บแพ็กเกจ (Packet) (บทที่ 3), ศึกษาภาษาเพร์ล (Perl) การทำงานและเขียนโปรแกรมซีจีไอ(CGI)

(บทที่ 4), ศึกษาการโจมตีเครือข่ายโดยใช้ TCP SYN Flooding,การสแกนพอร์ต, และการปลอมไอพี (บทที่ 5) เพื่อหาวิธีตรวจจับซึ่งอยู่ในส่วนของการออกแบบ (บทที่ 6) และตัวอย่างการใช้งานใน (บทที่ 7)

บทที่ 2
โพรโตคอลทีซีพีไอพี (TCP/IP Protocol)

2.1 ทีซีพีไอพี (TCP/IP)

เป็นโพรโตคอลมาตรฐานที่ใช้กันอยู่ในระบบปฏิบัติการแบบยูนิกซ์ เริ่มพัฒนาโดยกระทรวงกลาโหมของสหรัฐ ในปี ค.ศ. 1969 เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์หลายชนิดที่อยู่ห่างไกลกัน เครือข่ายที่จัดตั้งในระยะแรกชื่อว่า อาร์พาเน็ต (ARPANET)

ต่อมาได้พัฒนาเป็นเครือข่ายอินเตอร์เน็ต โพรโตคอลนี้เหมาะสำหรับเชื่อมต่อคอมพิวเตอร์ทั้งใกล้และไกลเข้าด้วยกัน และมีมาตรฐานรองรับทำให้ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์สามารถสร้างอุปกรณ์และโปรแกรมที่จะรองรับการทำงานของโพรโตคอลนี้ ทำให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ไม่ว่าจะเป็นเครื่อง ขนาดเล็กหรือขนาดใหญ่ หรือจะใช้ระบบปฏิบัติการอะไรก็ตาม ทีซีพีไอพี (TCP/IP) เป็นชุดโพรโตคอลที่ประกอบด้วยโพรโตคอลต่างๆหลายโพรโตคอลแต่ละโพรโตคอลมีคุณลักษณะและมีความสามารถในการทำงานแตกต่างกันโดยที่ในบทนี้จะได้กล่าวถึงรายละเอียดและคุณสมบัติของโพรโตคอลที่สำคัญบางโพรโตคอล

2.2 มาตรฐานระบบการสื่อสาร

กระบวนการติดต่อสื่อสารระหว่างอุปกรณ์สื่อสารหรือคอมพิวเตอร์นั้น สามารถแบ่งเป็นหน้าที่ย่อยได้เป็น 7 ระดับตามมาตรฐานของโอเอสไอ (ISO:International Organization for Standardization) การออกแบบโพรโตคอลทีซีพีไอพี นั้นไม่ได้เป็นไปตามรูปแบบของโอเอสไอโมเดล เนื่องด้วยถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลานานในการออกแบบตลอดจนการรับรองมาตรฐานต่างกับโพรโตคอลทีซีพีไอพี ที่ถูกแบบด้วยความต้องการอันเร่งด่วนของรัฐบาลสหรัฐ จึงทำให้การพัฒนาโพรโตคอลทีซีพีไอพี มีเงื่อนไขของในด้านความต้องการที่ต่างจาก โอเอสไอโมเดล ซึ่งหากเรามองโดยรวมแล้วจะเห็นว่าโพรโตคอลทีซีพีไอพี มีการแบ่งเป็นเลเยอร์ที่น้อยกว่าโอเอสไอซึ่งจะเปรียบเทียบกับ 5 ระดับตามมาตรฐานของทีซีพีไอพี ดังนี้

Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transmission Control Protocol & User Datagram Protocol
Network Layer	Internet Protocol
Data link Layer	Network Interface Protocol
Physical Layer	Physical Networks

ตารางที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของ โอเอสไอกับเลเยอร์ของทีซีพีไอพี

โดยที่ในแต่ละระดับมีหน้าที่แตกต่างกันซึ่งทำงานร่วมกัน

สาเหตุที่ต้องแบ่งระดับการสื่อสารออกเป็นหลายระดับก็เพื่อให้สินค้าของผู้ผลิตที่ต่างกันสามารถนำมาเชื่อมต่อกันได้ เช่น ผู้ผลิตการ์ดเครือข่ายสามารถเขียนโปรแกรมในระดับดาต้าลิงค์เลเยอร์ (Data link Layer) เพื่อให้รับข้อมูลจากผู้ผลิตที่เขียนโปรแกรมในระดับเน็ตเวิร์คเลเยอร์ (Network Layer) ได้

การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ในแต่ละระดับ จะติดต่อกันเองในลักษณะที่เรียกว่า peer-to-peer คือแต่ละ เลเยอร์ จะมองเฟรม (frame) ข้อมูลที่มาจากเลเยอร์ ข้างบนเป็นข้อมูลและจะนำเอาข้อมูลที่ใช้สำหรับการทำงานในเลเยอร์ ของตนเองครอบข้อมูลไว้ที่ส่วนหัวและส่วนท้าย แล้วส่งต่อไปยัง ชั้นข้างล่าง และทำเช่นนี้ไปเรื่อยๆจนกระทั่งส่งออกไปยังสายการสื่อสาร เมื่อเครื่องผู้รับได้รับเฟรมข้อมูลก็จะถอดรหัสส่วนหัวและส่วนท้ายออกไป แล้วนำส่งข้อมูลให้กับชั้นข้างบนต่อไปเป็นเช่นนี้ไปเรื่อยๆ จนถึงโปรแกรมแอปพลิเคชั่น

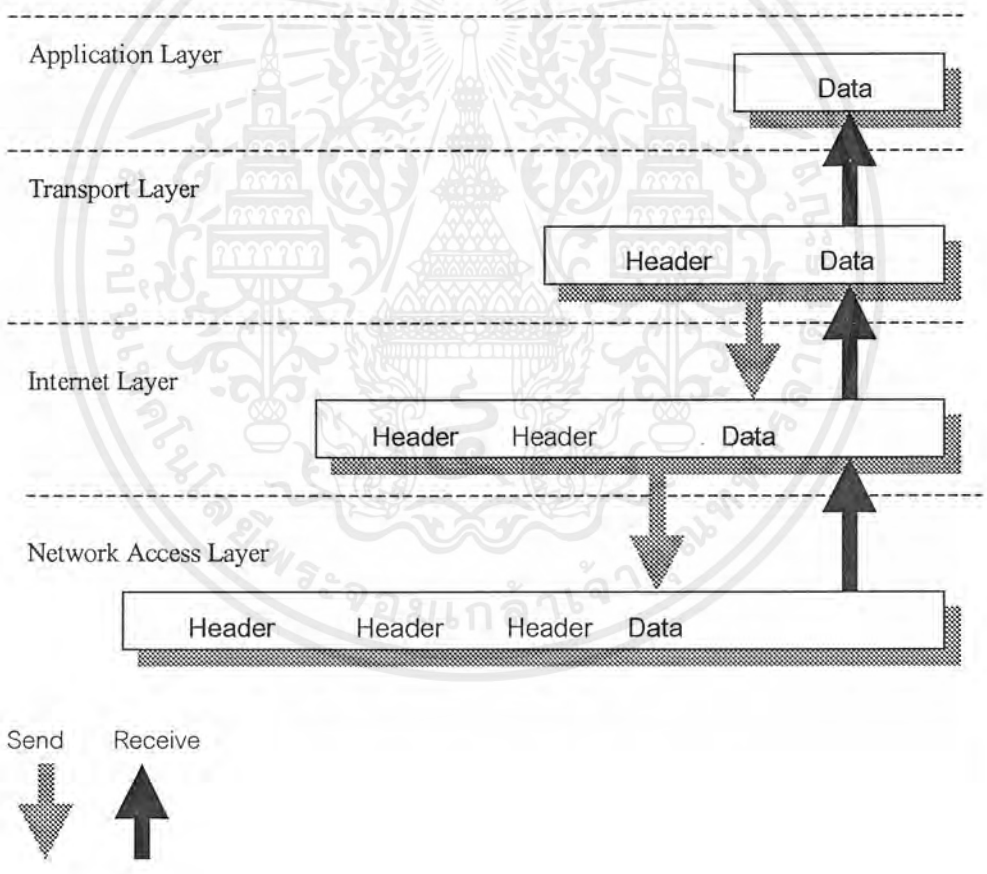
การทำงานในระดับทรานสปอร์ตเลเยอร์ (Transport Layer) ที่มีโพรโตคอลย่อยทีซีพี หรือยูดีพี (UDP) นั้นโปรแกรมที่อยู่ด้านบนจะเรียกใช้ผ่านช่องทางที่เป็นตัวเลขที่เรียกว่าหมายเลขพอร์ต (Port Number) โดยหมายเลขนี้เป็นเลขที่มาตรฐานในโพรโตคอลแบบทีซีพี มาตรฐานของหมายเลขพอร์ต ทั้ง ทีซีพี และ ยูดีพีได้แสดงไว้ในตารางต่อไปนี้

TCP Port Number	Application Layer Services
5	Remote Job Entry
7	Echo
20	FTP Data
21	FTP Control
23	TELNET
25	SMTP
37	Time
53	Domain Name Server(DNS)
66	Oracle SQL*NET
80	World Wide Web HTTP
110	Post Office Protocol(POP3)

ตารางที่ 2-2 แสดงบริการและหมายเลขพอร์ตของทีซีพีโพรโตคอล

UDP Port Number	Application Layer Interface
7	Echo
13	DayTime
37	Time
69	Travel File Transfer Protocol(TFTP)
70	Gopher
107	Remote Telnet
161	SNMP(Simple Network Management Protocol)

ตารางที่ 2-3 แสดงชื่อบริการและหมายเลขพอร์ตของยูดีพีโพรโตคอล



รูปที่ 2-1 Data Encapsulation

ทีซีพีไอพีโพรโตคอล ถูกออกแบบให้ทำหน้าที่ในการเชื่อมโยงเครือข่ายกลุ่มย่อยเข้า ด้วยกัน เป็นขนาดใหญ่ เรียกว่า อินเตอร์เน็ต โดยที่การทำงานจะไม่ขึ้นกับชนิดของเครื่อง โปรแกรม ที่ควบคุม

ระบบ รวมทั้งตัวกลาง และเทคโนโลยีที่ใช้ในการเชื่อมโยง ซึ่งมีขั้นตอนของการจัดการกับความผิดพลาดที่เกิดขึ้นในเครือข่าย

2.3 โครงสร้างของระบบส่งข้อมูล

ลักษณะการทำงานของโมเดล ในลักษณะนี้คือ ข้อมูลจะถูกส่งลงมาจากชั้นบนลงมาชั้นล่างสุดซึ่งมีหน้าที่จัดการเกี่ยวกับการส่งข้อมูลผ่านสายสัญญาณไปยังจุดหมายปลายทาง เมื่อข้อมูลไปถึงจุดหมายแล้วก็จะกลับย้อนจากล่างไปบนสุด ซึ่งเป็นชั้นที่โปรแกรมใช้งานต่าง ๆ ทำงานอยู่

ขณะที่ข้อมูลถูกส่งผ่านจากชั้นบนลงมายังชั้นล่าง แต่ละชั้นจะทำการเพิ่มข้อมูลควบคุมเข้าไปเพื่อให้การส่งข้อมูลถูกต้อง และเป็นการส่งพารามิเตอร์ที่จำเป็นไปให้กับชั้นของมันในเครื่องปลายทาง ข้อมูลควบคุมเหล่านี้เรียกว่า ส่วนหัว (Header) แต่ละชั้นจะมีส่วนหัว ที่มีรูปแบบเป็นของตัวเอง การเพิ่ม ส่วนหัวเข้าไปกับข้อมูลนั้นเราเรียกว่า data encapsulation ดังรูปที่ 2-1

ขั้นตอนการทำงานเพื่อให้เกิดการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ เป็นไปอย่างถูกต้อง มีขั้นตอนดังนี้

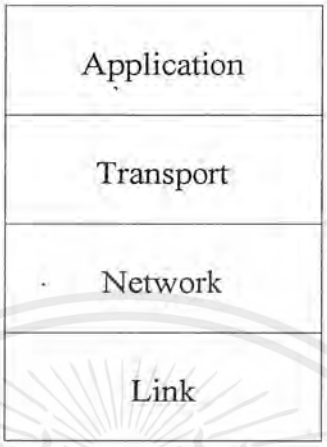
- กำหนดรูปแบบข้อมูล
- จัดเตรียมชุดข้อมูล
- กำหนดเส้นทางการส่งข้อมูล
- กำหนดอัตราความเร็วในการส่งข้อมูล
- ส่งข้อมูลผ่านตัวกลาง
- รวบรวมและจัดลำดับชุดข้อมูลที่ส่งมา
- ตรวจสอบว่ามีชุดข้อมูลซ้ำหรือไม่
- ตอบกลับไปให้ผู้ส่ง รู้ว่าได้รับแล้ว
- ผ่านข้อมูลไปให้โปรแกรมใช้งาน

การรูปแบบโพรโตคอลทีซีพีไอพี ของระบบปฏิบัติการต่างๆไม่เหมือนกันเช่น ในระบบปฏิบัติการยูนิกซ์โพรโตคอล ทีซีพีไอพี จะเป็นส่วนหนึ่งของระบบปฏิบัติการ ในระบบปฏิบัติการวินโดวส์เอ็นที, VMS, OS/2, วินโดว์ (Windows) จะอยู่ในรูปของดีไวซ์ไดรเวอร์ (Device Driver) เพื่อให้โปรแกรมเรียกใช้ ซึ่งต้องทำการติดตั้งดีไวซ์ไดรเวอร์ก่อน ในระบบปฏิบัติการเอ็มเอส (MS-DOS) มักจะสร้างเป็นโปรแกรมแบบฝังตัว (TSR : Terminate Stay Resident) เมื่อเรียกโปรแกรมแล้วจะฝังตัวอยู่ในหน่วยความจำเพื่อให้โปรแกรมสามารถเรียกใช้งานได้

กรณีของระบบปฏิบัติการแบบ วินโดว์ โพรโตคอลทีซีพีไอพี จะถูกสร้างอยู่ในรูปของดีแอลแอล (DLL: Dynamic Link Library) เพื่อให้โปรแกรมในวินโดวส์สามารถเรียกใช้งานได้ โดยมีมาตรฐานเรียกว่า วินซ็อค (Winsock: Window Socket) ซึ่งมีบริษัทต่างๆที่ได้คิดค้นโพรโตคอลแบบทีซีพีไอพี ที่ทำงานในลักษณะนี้หลายๆบริษัท เช่น บริษัทไมโครซอฟต์, บริษัทโนเวล, FTP Software Corp, NetManage Corp รวมทั้งโปรแกรมที่เป็นประเภทแชร์แวร์ (Shareware) ได้แก่ Trumpet Winsock

2.4 การทำงานของทีซีพีตามชั้น (Layer)

ทีซีพีไอพี ได้แบ่งการทำงานของการติดต่อสื่อสารกันในเครือข่ายเป็นชั้น ซ้อนกันอยู่ 4 ชั้น แต่ละชั้นมีหน้าที่ต่างกัน ดังรูป



รูปที่ 2-2 แสดงชั้นการทำงานของทีซีพี

1. ลิงค์เลเยอร์ (Link layer) บางทีเรียกว่า ดาต้าลิงค์เลเยอร์ (Data link layer) หรือ เน็ตเวิร์คอินเตอร์เฟสเลเยอร์ (Network interface layer) เป็นส่วนที่รวมดีไวซ์ไดเวอร์ (device driver) ในโอเอสและตัวอินเตอร์เฟสการ์ด (interface card) ในคอมพิวเตอร์รวมทั้ง รายละเอียดฮารด์แวร์ (hardware) ของการอินเตอร์เฟส (interfacing) สายหรือรูปแบบของข้อมูล

2. เน็ตเวิร์คเลเยอร์ (Network layer) บางทีเรียกว่า อินเตอร์เน็ตเลเยอร์ (Internet layer) เป็นชั้นที่ทำหน้าที่ควบคุมและเลือกเส้นทางเดินของข้อมูล ประกอบด้วยเราท์เตอร์ของแพ็กเกจ (Routing - of packets) เช่น ไอพี (IP : Internet Protocol) ,ไอซีเอ็มพี (ICMP: Internet Control Message Protocol) และไอจีเอ็มพี (IGMP: Internet Group Management Protocol)

3 ทรานสปอร์ตเลเยอร์ (Transport layer) เป็นชั้นที่จัดการกับข้อมูลสื่อสารระหว่างต้นทางกับปลายทางมาตราฐานในชั้นนี้ ได้แก่ ทีซีพีและยูดีพี ทีซีพีจะจัดการเกี่ยวกับการส่งข้อมูลระหว่างต้นทางกับปลายทาง เช่น ขนาดของข้อมูล, Setting Timeout, acknowledges packets, acknowledges received packets ยูดีพีจะจัดการกับข้อมูลจากชั้นของแอปพลิเคชั่นจากนั้นส่งและรับข้อมูลเป็น แพ็กเกจ เรียกว่า คาต้าแกรม (datagram) แต่ไม่รับประกันว่าคาต้าแกรมจะถูกส่งถึงจุดหมายหรือไม่

4. แอปพลิเคชั่นเลเยอร์ (Application layer) จะเป็นงานด้านต่าง ๆ แบ่งเป็น

Telnet for remote login

FTP (File Transfer Protocol)

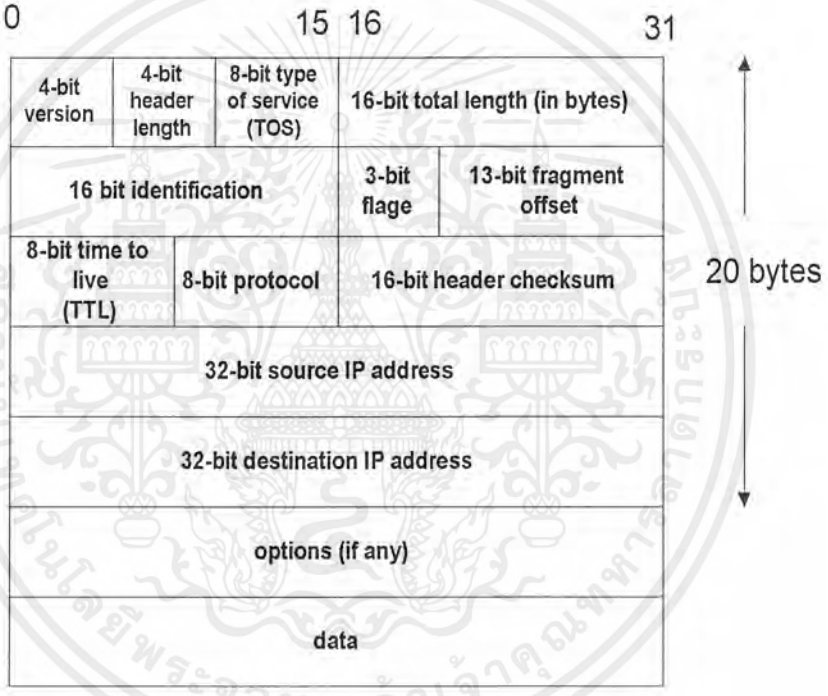
SMTP (Simple Mail Transfer protocol), electronic mail

SNMP (Simple Network Management Protocol)

หลังจากศึกษาการทำงานชั้นต่าง ๆ ของ ทีซีพีไอพี การจะส่งข้อมูลจากต้นทางไปยังปลายทาง จะมีอุปกรณ์ที่ทำหน้าที่ปรับข้อมูล เรียกว่า เกตเวย์ (Getaway) และ เราท์เตอร์ (router) ซึ่งเป็นตัวหาเส้นทางโดยเมื่อตัว เราท์เตอร์ รับชุดข้อมูลมา มันจะดูในส่วน ส่วนหัวของไอพี ซึ่งมีหมายเลขของเครือข่ายของจุดหมายปลายทาง แล้วตัดสินใจว่าจะส่งข้อมูลไปทางไหน โดยดูจากตารางเส้นทางที่มันสร้างไว้

2.5 แพ็กเกจของไอพี

ส่วนหัวของไอพี เป็น แพ็กเกจ ที่บอกว่าส่งมาจากทางใดและจะส่งไปยังปลายทางที่ไหนดังมีรายละเอียดละความหมายดังนี้



รูปที่ 2-3 แสดงรูปแบบของไอพีแพ็กเกจ

Version : ในช่องนี้จะบอก เวอร์ชั่น ของ ไอพี ถ้าเป็น เวอร์ชัน 4 จะเรียก IPv4.

Header length : จะบอกความยาวของ ส่วนหัวของไอพีแพ็กเกจ

TOS : Type-of-service field : บอกค่า ทีโอเอส (TOS) 4 แบบซึ่งเป็นค่าที่ บอกถึงชนิดของการบริการของแพ็กเกจ ซึ่งประกอบด้วย minimize delay, maximize throughput, maximize reliability, และ minimize monetary cost.

Total length field : จะบอกความยาวรวมทั้งหมดของไอพีดาต้าแกรม (IP Datagram) สูงสุด 65535 ไบต์

Identification field : เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ทำการส่งจะเป็นผู้กำหนด และจะเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งไอพีดาต้าแกรมใหม่

Flag field และ Fragmentation offset field : บอกค่าเกี่ยวกับการแบ่งไอพีดาต้าแกรม (fragmentation)

Time-to-live filed or TTL : ใช้กำหนดจำนวนเราท์เตอร์สูงสุด ที่ไอพีดาต้าแกรมสามารถผ่านได้ โดยทั่วไปจะอยู่ที่ระหว่าง 32 หรือ 64

ICMP message Protocol field : ใช้สำหรับแยกชนิดของโพรโตคอล (demultiplex) ที่จะนำไปใช้ในโพรโตคอลชั้นบน

Header checksum : ใช้คำนวณหาความถูกต้องของข้อมูลว่ามีความผิดพลาดหรือไม่โดยไม่

บทที่ 3

tcpdump

3.1 tcpdump คืออะไร?

tcpdump คือ โปรแกรมที่ทำหน้าที่จับแพ็กเกจที่วิ่งอยู่ในเครือข่าย โดยโปรแกรม tcpdump จะทำงานร่วมกับเน็ตเวิร์กอินเตอร์เฟสการ์ดในโหมด Promiscuous – mode บนเครื่องที่ได้รับการติดตั้งโปรแกรม tcpdump โดยการทำงานร่วมกันในโหมดนี้ แพ็กเกจทั้งหมดในเครือข่ายจะสามารถถูกโปรแกรม tcpdump จับขึ้นมาได้ เพราะการทำงานของเน็ตเวิร์กอินเตอร์เฟสการ์ดในโหมดนี้จะไม่มีการเปรียบเทียบแอดเดรสปลายทางของข้อมูลกับแอดเดรสของเน็ตเวิร์คอินเตอร์เฟสการ์ด นั่นก็คือทุกๆ แพ็กเกจที่มาถึงยังเน็ตเวิร์คอินเตอร์เฟสการ์ดจะถูกโปรแกรม tcpdump จับขึ้นมาได้ทั้งหมดไม่เลือกว่าจะระบุแอดเดรสปลายทางของเน็ตเวิร์กอินเตอร์-เฟสการ์ดไหน ข้อนี้ถือเป็นข้อดีของโปรแกรม tcpdump เพราะว่าสามารถทำให้ดูข้อมูลของแพ็กเกจได้ทั้งเครือข่าย

โปรแกรม tcpdump จะถูกใช้ในการเฝ้าดูการจราจรในเครือข่ายเพื่อการบริหารเครือข่ายให้มีประสิทธิภาพและแก้ปัญหาต่างๆ ที่เกิดจากการทำงานผิดพลาดของเครือข่ายและยังเป็นเครื่องมือที่ดีในการศึกษาการทำงานของโพรโตคอลรวมถึงการดูแลความปลอดภัยของระบบเครือข่าย

tcpdump จะทำการพิมพ์ของส่วนหัวของแพ็กเกจที่วิ่งในเครือข่ายออกมาในรูปของเท็กซ์(text) ทำให้สามารถใช้โปรแกรมที่สามารถประมวลผลข้อมูลเท็กซ์ มาวิเคราะห์แพ็กเกจได้ แต่การรันโปรแกรม tcpdump จะมีข้อกำหนดสิทธิ์ในการรันแตกต่างกันในแต่ละระบบดังนี้

- ภายใต้ระบบ SunOS ที่มี nit หรือ bpf

การรัน tcpdump จะต้องการมีสิทธิ์ การอ่าน (read access) /dev/nit หรือ /dev/bpf

- ภายใต้ระบบ Solaris ที่มี dlpi

การรัน tcpdump จะต้องใช้สิทธิ์การอ่านอุปกรณ์ network pseudo device เช่น /dev/le

- ภายใต้ HP - UX ที่มี dlpi

การรัน tcpdump จะต้องใช้สิทธิ์รูท (root)

- ภายใต้ IRIX ที่มี snoop

การรัน tcpdump จะต้องใช้สิทธิ์รูท

- ภายใต้ระบบ BSD

จะต้องมีสิทธิ์ในการอ่าน /dev/bpf

3.2 การใช้งานโปรแกรม tcpdump

tcpdump สามารถจับแพ็กเกจที่วิ่งผ่านสายในเครือข่าย โดยจุดมุ่งหมายของการสร้างโปรแกรมต้องการที่จะหลีกเลี่ยงการนำโปรแกรมไปใช้ในทางที่ผิด เช่น การเก็บพาสเวิร์ด (password) หรือการกระทำเพื่อการเจาะระบบ (Hack) ต่างๆ ผลลัพธ์ของโปรแกรมจึงออกมาเป็นเลขฐานสิบหก (Hex) ในส่วนที่เป็นข้อมูลทั้งหมดของแพ็กเกจ ยกเว้นส่วนที่เป็นส่วนหัวของแพ็กเกจที่จะแสดงในรูปเท็กซ์

3.2.1 OPTION ของโปรแกรม tcpdump

ใช้กำหนดรูปแบบของการทำงานและการแสดงผลดังอธิบายย่อๆ ดังนี้

-a แปลงแอดเดรสไปเป็นชื่อ

-c ออกจากโปรแกรมหลังจากจับแพ็กเกจได้ครบตามจำนวนที่กำหนดไว้

-e พิมพ์แอดเดรสของเน็ตเวิร์คอินเตอร์เฟสการ์ด

-F ใช้เพื่อนำเอาไฟล์ที่เก็บ expression มาเป็นอินพุตของ command line

-i กำหนดเน็ตเวิร์คอินเตอร์เฟสการ์ดที่จะทำการจับแพ็กเกจในกรณีที่มีจำนวนเน็ตเวิร์คอินเตอร์เฟสการ์ดมากกว่าหนึ่งการ์ด ตามปกติถ้าไม่กำหนด tcpdump จะเลือกเอาหมายเลขที่ต่ำที่สุด

-l ทำให้ผลลัพธ์ออกทาง Stdout เช่น ในการจับแพ็กเกจลงไปเก็บไว้ในไฟล์จะไม่สามารถมองเห็นผลลัพธ์ทางหน้าจอได้ จะต้องใช้ -l ร่วมด้วยในกรณีต้องการให้ออกที่หน้าจอ

-n ไม่ต้องแปลงแอดเดรสไปเป็นชื่อ

-N ไม่พิมพ์ domain name ของชื่อโฮสต์ เช่น ถ้า -N ถูกกำหนดชื่อโฮสต์"nic.ddn.mil" จะพิมพ์เพียง "nic"

-p ไม่ต้องการการทำงานในโหมด promiscuous mode

-q พิมพ์ผลลัพธ์โดยเร็ว, พิมพ์ผลลัพธ์ออกมาสั้นๆ

-r อ่านแพ็กเกจจากไฟล์ (ที่สร้างจาก -w) และสามารถอ่านจาก Standard input โดยกำหนดชื่อไฟล์เป็น "-"

-s กำหนดความยาวขนาดของแพ็กเกจที่ต้องการจะจับ

-t ไม่ต้องพิมพ์ timestamp ในแต่ละ แพ็กเกจ

-tt พิมพ์ timestamp ที่ยังไม่ได้จัดฟอร์แมตในแต่ละ แพ็กเกจ

-v เพิ่มรายละเอียดของผลลัพธ์ที่ออกมา เช่น time to live และ type of service

-vv เพิ่มรายละเอียดของผลลัพธ์ โดยจะมากกว่า -v

-w เขียนแพ็กเกจที่จับได้ไฟล์ แทนที่จะออกหน้าจอ โดยจะสามารถพิมพ์ออกมาโดยใช้ -r ในภายหลัง และสามารถเขียนลง Standard output ถ้ากำหนดให้ชื่อไฟล์เป็น "-"

-x พิมพ์แต่ละแพ็กเกจทั้งหมดออกมาในรูปแบบเลขฐานสิบหก โดยจะรวมเอาส่วนของข้อมูลด้วย

3.2.2 EXPRESSION ของโปรแกรม tcpdump

ใช้เพื่อเลือกแพ็กเกจที่จะทำการจับ โดยถ้าไม่กำหนด tcpdump จะทำการจับแพ็กเกจทุกๆ แพ็กเกจแต่ถ้าหากมีการกำหนด expression ก็จะจับเฉพาะแพ็กเกจที่เป็นจริง ตาม expression เท่านั้น โดยจะประกอบด้วย expressionn ต่าง ๆ ดังนี้

type ใช้เพื่อกำหนดชื่อหรือหมายเลขของสิ่งที่ต้องการอ้างอิงถึง

- *host* : กำหนดชื่อโฮสต์เช่น "host foo"
- *net* : กำหนดชื่อเครือข่ายเช่น "net 182.3"
- *port* : กำหนดชื่อหรือหมายเลขของพอร์ตเช่น " port 20 "

ถ้าหากไม่มีการกำหนด *type* tcpdump จะกำหนดให้เป็น *host*

dir ใช้เพื่อกำหนดทิศทางของแพ็กเกจที่ต้องการจับโดยสามารถใช้ and/or ร่วมด้วย

- *src* : กำหนดต้นทาง *type* ของแพ็กเกจที่ต้องการ เช่น "src foo"
- *dst* : กำหนดปลายทาง *type* ของแพ็กเกจที่ต้องการ เช่น "dst net 128.3"
- *src or dst* : กำหนดต้นทางหรือปลายทาง *type* ของแพ็กเกจที่ต้องการ เช่น " src or dst port ftp - data"
- *src and dst* : กำหนดต้นทางและปลายทาง *type* ของแพ็กเกจที่ต้องการ เช่น "src and dst net 128.3"

ถ้าหากไม่มีการกำหนด *dir* tcpdump จะกำหนดให้เป็น *src or dst*

proto ใช้เพื่อกำหนดโพรโตคอลเฉพาะลงไปในของแพ็กเกจที่ต้องการจะจับโดยมีคำที่สามารถใช้ได้ดังนี้

- ether
- fddi
- ip
- arp
- rarp
- decnet
- lat
- sca
- moprc
- mopdi
- tcp
- udp

ถ้าหากไม่มีการกำหนดโพรโตคอล tcpdump จะจับทุก ๆ โพรโตคอล

นอกจากนี้ ยังมี expression พิเศษอีก คือ

- Gateway
- Broadcast
- Less
- Greater
- Arithmetic expression >,<,>=,<=,=,!= และค่าคงที่

รวมทั้งการทำงานไบนารี [+,-,*,/,\,[]

- การเข้าถึงข้อมูลภายในแพ็กเกจโดยมีรูปแบบดังนี้

proto[expr : size]

proto : ชนิดของโพรโตคอล

expr : ตำแหน่งของข้อมูลภายในแพ็กเกจนับเป็นไบต์

Size : จำนวนข้อมูลที่ต้องการมีหน่วยเป็นไบต์

นอกจากนี้ ยังสามารถใช้

( ) กลุ่มของ expression

" ! " or 'not'

" &&" or 'and'

" || ' or 'or'

โดย '!' or 'not' จะมีความสำคัญ (priority) สูงสุด '&&' '|' มีสำคัญเท่ากันเท่ากัน และมีลำดับเท่ากันและมีสำคัญจากซ้ายไปขวา

ตัวอย่างการใช้ Expression

host ใช้เพื่อกำหนดโฮสต์ของแพ็กเกจที่ต้องการจะจับ

#tcpdump host diamond.ce.kmitl.ac.th

จับแพ็กเกจที่เข้าหรือออกจากโฮสต์ diamond.ce.kmitl.ac.th

net ใช้เพื่อกำหนดหมายเลขของเครือข่ายของแพ็กเกจที่ต้องการจับ

#tcpdump net 161.246

จับแพ็กเกจที่มาจากเครือข่ายหมายเลข 161.246 ที่ผ่านมาทางโฮสต์ที่รันโปรแกรม tcpdump

port ใช้เพื่อกำหนดพอร์ตของแพ็กเกจที่ต้องการจับ

#tcpdump port 20

จับแพ็กเกจที่ผ่านเข้าออกทางพอร์ต 20

src,dst ใช้เพื่อกำหนดทิศทางของแพ็กเกจที่ต้องการจับ

#tcpdump src diamond.ce.kmitl.ac.th

จับแพ็กเกจที่ต้นทางมาจากโฮสต์ diamond.ce.kmitl.ac.th

#tcpdump dst net 161.246

จับแพ็กเกจที่ปลายทางเป็นเครือข่ายหมายเลข 161.246

proto [expr : size] ใช้เพื่อต้องการเข้าถึงข้อมูลในแพ็กเกจ โดยที่

proto = ether,fddi,ip,arp,rarp,tcp,udp,icmp อย่างใดอย่างหนึ่ง

expr = ไบต์ออฟเซตของข้อมูลในแพ็กเกจ

size = ขนาดของข้อที่ต้องการ มีหน่วยเป็นไบต์

#tcpdump 'ether[0] & 1 != 0'

จับแพ็กเกจชนิดอีเธอร์เน็ต (Ethernet) โดยที่ไบต์แรก (expr = 0) ขนาด 1 ไบต์ (ถ้า size ไม่มี tcpdump จะกำหนดให้เป็น 1 ไบต์) and กับ 1 แล้วไม่เท่ากับ 0

#tcpdump 'ip[6:2] & 0x1fff = 0'

จับแพ็กเกจชนิดไอพีโดยที่ไบต์ที่ 6 (expr = 6) ขนาด 2 ไบต์ (size = 2) and กับ 0x1fff แล้วไม่เท่ากับ 0

3.3 ตัวอย่างการใช้งานและผลลัพธ์ของโปรแกรม tcpdump

ผลลัพธ์ของโปแกรม tcpdump จะขึ้นอยู่กับโพรโตคอล การกำหนดออปชัน (option) และ experssion ทำให้สามารถกำหนดขอบเขตของข้อมูลที่จะจับและรูปแบบของผลลัพท์ที่นำไปใช้ต่อไป ตัวอย่างข้างล่างนี้เป็นตัวอย่างการใช้คำสั่งและผลของโปรแกรม

#tcpdump –e ← คำสั่ง

00:43:27.481297 0:80:ad:6:a6:79 0:a0:24:b3:54:e0 ip 114: media02.ce.kmitl.ac.th.1467 > ← บรรทัดที่ 1

as2-13.qualitynet.net.10516: P 30339803:30339863(60) ack 6363488 win 8350 (DF) ← บรรทัดที่ 2

00:43:27.481297 2:60:8c:6a:19:7d 0:a0:24:b3:54:e0 ip 88: Zintoo.kmitl.ac.th.1078 > ← บรรทัดที่ 3

Chaokhun.kmitl.ac.th.domain: 63163+ (46) ← บรรทัดที่ 4

00:43:39.821297 0:a0:24:b3:54:e0 Broadcast arp 60: ← บรรทัดที่ 5

arp who-has digit06.ce.kmitl.ac.th tell compnet.ce.kmitl.ac.th ← บรรทัดที่ 6

00:43:44.661297 0:20:18:61:82:cf 0:20:18:62:38:aa 0028 60: a1f60600.00:20:18:61:82:cf.4068 >

19990503.00:00:00:00:00:01.451: ipx-ncp 10

จากข้างบนผลของคำสั่ง tcpdump จะทำการแปลงแพ็กเกจให้ออกมาอยู่ในรูปเท็กซ์ บรรทัดที่ 1 และบรรทัดที่ 2 จะเป็น 1 แพ็กเกจ บรรทัดที่ 3,4 เป็นอีก 1 แพ็กเกจและบรรทัดที่ 5,6 เป็นอีก 1 แพ็กเกจ ในแต่ละชุดของเท็กซ์จะถูกแบ่งออกเป็นส่วนๆ เพื่อแสดงข้อมูลของแต่ละส่วน โดยจะประกอบด้วย เวลา (Time Stamp) ข้อมูลในส่วนหัวของชั้นลิงค์เลเยอร์, ข้อมูลในส่วนหัวของชั้นเน็ตเวิร์คเลเยอร์ และข้อมูลในส่วนหัวของชั้น

ทรานสปอร์ตเลเยอร์ดังนี้

00:43:27.481297 ← Time Stamp

0:80:ad:6:a6:79 0:a0:24:b3:54:e0 ip 114: ← Link Layer

media02.ce.kmitl.ac.th.1467 > as2-13.qualitynet.net.10516: ← Network Layer

P 30339803:30339863(60) ack 6363488 win 8350 (DF) ← Transport Layer

การตีความหมายของข้อมูลก็จะแบ่งอธิบายแยกตามแต่ละส่วนดังนี้

3.3.1 เวลา (Time Stamp)

จะเป็นเวลาที่เน็ตเวิร์คอินเตอร์เฟสการ์ดได้รับแพ็กเกจ โดยจะมีรูปแบบอยู่ 2 แบบ คือแบบที่ 1 เวลาที่มีการจัดรูปแบบแล้วโดยจะมีรูปแบบเป็น [ชั่วโมง:นาที:วินาที] และแบบที่ 2 เวลาที่ยังไม่ได้จัดรูปแบบ โดยจะมีรูปแบบเป็น [วินาที] ซึ่งเป็นวินาทีที่เริ่มนับจากวันที่ 1 เดือนมกราคม ค.ศ. 1970 และต้องกำหนดด้วยออปชั่น –tt

00:43:27.481297 ← แบบที่ 1

911482869.041368 ← แบบที่ 2

3.3.2 ลิงค์เลเยอร์ (Link Layer)

ในขั้นนี้จะมีโพรโตคอลคือ อีเธอร์เน็ต, FDDI และ SLIP การที่จะให้โปรแกรม tcpdump แสดงรายละเอียดของข้อมูลในขั้นนี้ออกมาจะต้องใช้ –e ผลลัพธ์ของโปแกรมจะมีความแตกต่างกันเล็กน้อยยกตัวอย่าง เช่น บนอีเธอร์เน็ต กับ FDDI ผลลัพท์ของแพ็กเกจที่วิ่งบน FDDI จะมีการพิมพ์ “frame control” ออกมาด้วยในขณะที่บนอีเธอร์เน็ตจะไม่มี ตัวอย่างต่อไปนี้เป็นแพ็กเกจที่วิ่งบนเครือข่ายอีเธอร์เน็ต

0:80:ad:6:a6:79 0:a0:24:b3:54:e0 ip 114:

มีความหมายดังนี้

0:80:ad:6:a6:79 ← Ethernet Source Address

แสดงแอดเดรสต้นทางของเน็ตเวิร์คอินเตอร์เฟสการ์ด

0:a0:24:b3:54:e0 ← Ethernet Destination Address

แสดงแอดเดรสปลายทางของเน็ตเวิร์คอินเตอร์เฟสการ์ด

ip ← Fram Type

แสดงชนิดของเฟรมนี้คือไอพีซึ่งก็คือโพรโตคอลที่อยู่ในชั้นบนของอีเธอร์เน็ตเฟรมจะประกอบด้วย

ไอพี	0x0800
เออาร์พี	0x0806
อาร์เออาร์พี	0x8305
แอปเปิ้ลทอร์ค	0x809B
แอปเปิ้ลทอร์ค เออาร์พี	0x80F3
เน็ตแวร์ ไอพีเอ็กซ์/เอสพีเอ็กซ์	0x8137

114: ← Frame Length

แสดงขนาดความยาวของอีเธอร์เน็ตเฟรมซึ่งมีหน่วยเป็นไบต์

3.3.3 เน็ตเวิร์คเลเยอร์ (Network Layer)

ในส่วนนี้จะเป็นข้อมูลในส่วนของโพรโตคอลที่ถูกหุ้มโดยอีเธอร์เน็ตเฟรมชั้นหนึ่งซึ่งเป็นชั้นของเน็ตเวิร์คเลเยอร์จะประกอบไปด้วยโพรโตคอล ไอพี (IP), เออาร์พี (ARP), อาร์เออาร์พี(RARP)

00:45:43.441297 2:60:8c:6a:19:7d 0:a0:24:b3:54:e0 ip 86: ← บรรทัดที่ 1

Zintoo.kmitl.ac.th.1108 > Chaokhun.kmitl.ac.th.domain: 63192+ (44) ← บรรทัดที่ 2

00:43:51.921297 0:a0:24:b3:54:e0 0:80:48:ed:74:b0 arp 60: ← บรรทัดที่ 3

arp who-has technician02.ce.kmitl.ac.th (0:80:48:ed:74:b0) tell compnet.ce.kmitl.ac.th ← บรรทัดที่ 4

00:43:51.921297 0:80:48:ed:74:b0 0:a0:24:b3:54:e0 arp 60: ← บรรทัดที่ 5

arp reply technician02.ce.kmitl.ac.th is-at 0:80:48:ed:74:b0 ← บรรทัดที่ 6

จากตัวอย่างข้างบนความหมายของข้อมูลส่วนของเน็ตเวิร์คจะมีความแตกต่างกันตามชนิดของโพรโตคอลซึ่งได้ยกเอาแพ็กเกจของไอพี (บรรทัดที่ 1,2) กับเออาร์พี (บรรทัดที่ 3-6) มาเป็นตัวอย่าง

Zintoo.kmitl.ac.th.1108 > Chaokhun.kmitl.ac.th.domain: ← IP Header

มีความหมายดังนี้

Zintoo.kmitl.ac.th. ← IP Source Address

แสดงแอดเดรสต้นทางของไอพีแพ็กเกจโดยเป็นชื่อหรือหมายเลขไอพี

1108 ← Source Port Number

แสดงหมายเลขพอร์ตต้นทางของไอพีแพ็กเกจโดยเป็นชื่อบริการหรือหมายพอร์ต

> ← Direction

แสดงทิศทางต้นทางกับปลายทางของไอพีแพ็กเกจ

Chaokhun.kmitl.ac.th. ← IP Destination Address

แสดงแอดเดรสปลายทางของไอพีแพ็กเกจโดยเป็นชื่อหรือหมายเลขไอพี

domain: ← Destination Port Number

แสดงหมายเลขพอร์ตปลายทางของไอพีแพ็กเกจโดยเป็นชื่อบริการหรือหมายพอร์ต

arp who-has technician02.ce.kmitl.ac.th (0:80:48:ed:74:b0) tell compnet.ce.kmitl.ac.th ←ARP Request

มีความหมายดังนี้

arp who-has

แสดงชนิดของแพ็กเกจที่ทำการร้องขอเออาร์พี

technician02.ce.kmitl.ac.th (0:80:48:ed:74:b0)

แสดงชื่อหรือหมายเลขไอพีของเครื่องที่ต้องการทราบหมายเลขเน็ตเวิร์กอินเตอร์เฟสการ์ด ในกรณีที่มีข้อมูลอยู่ในหน่วยความจำของเออาร์พี (ARP Cache) จะมีวงเล็บหมายเลขของเน็ตเวิร์คอินเตอร์เฟสการ์ดด้วย

tell compnet.ce.kmitl.ac.th

แสดงชื่อหรือหมายเลขไอพีของเครื่องที่ทำการร้องขอเออาร์พี เพื่อให้เครื่องที่ทราบทำการตอบกลับ

arp reply technician02.ce.kmitl.ac.th is-at 0:80:48:ed:74:b0 ← ARP Reply

มีความหมายดังนี้

arp reply

แสดงชนิดของแพ็กเกจที่ทำการตอบรับการร้องขอเออาร์พี

technician02.ce.kmitl.ac.th is-at 0:80:48:ed:74:b0

แสดงชื่อหรือหมายเลขไอพีและหมายเลขเน็ตเวิร์คอินเตอร์เฟสการ์ด

3.3.4 ทรานสปอร์ตเลเยอร์ (Transport Layer)

ในชั้นนี้จะเป็นข้อมูลของโพรโตคอลถูกไอพีคาต้าแกรมหุ้มอยู่ ซึ่งประกอบด้วย ทีซีพี และ ยูดีพี

00:43:27.791297 0:a0:24:b3:54:e0 0:80:48:ed:74:b0 ip 60: Chaokhun.kmitl.ac.th.3128 >

technician02.ce.kmitl.ac.th.1208: S 1637511044:1637511044(0) ack 123689 win 8760 <mss 1460> (DF)

00:43:22.611297 2:60:8c:6a:19:7d 0:a0:24:b3:54:e0 ip 86: Zintoo.kmitl.ac.th.1077 >

Chaokhun.kmitl.ac.th.domain: 63162+ (44)

จากตัวอย่างข้างบนจะประกอบด้วย 2 แพ็กเกจ คือ ทีซีพีแพ็กเกจและยูดีพีแพ็กเกจ

S 1637511044:1637511044(0) ack 123689 win 8760 <mss 1460> (DF) ← TCP Header

มีความหมายดังนี้

S ← Flag

แสดงแฟล็กซึ่งใช้ในการติดต่อในส่วนของทีซีพีซึ่งประกอบด้วย S (SYN), F (FIN), R (RST), P (PSH), . (No Flag Set)

1637511044:1637511044(0) ← Sequence Number

แสดง หมายเลข Sequence Number ของแพ็กเกจและจำนวนของข้อมูล (ที่มีหน่วยเป็นไบต์) ที่อยู่ภายในวงเล็บที่เกิดจากผลต่างหมายเลข Sequence Number ตัวหน้ากับตัวหลังโคลอน

ack 123689 ← Acknowledgment Number

แสดง หมายเลข Acknowledgment Number ของแพ็กเกจ ซึ่งแฟล็ก ACK จะถูกแยกออกมาต่างหากไม่รวมอยู่ในส่วนของแฟล็ก S, F, R ,P และ .

win 8760 ← Window Size

แสดง ขนาดของวินโดว์ที่ใช้สำหรับรับข้อมูลที่ทางฝั่งรับสามารถจะรับได้

<mss 1460> ← Maximum Segment Size (MSS)

แสดง ค่าขนาดของทีซีพีเซ็กเมนต์ (TCP header + TCP data) ที่ทางด้านผู้รับไม่ต้องการรับข้อมูลที่มีขนาดใหญ่เกินนี้

(DF) ← Don't Fragment

แสดง การอนุญาตให้มีการแบ่งแพ็กเกจ (Fragment) หรือไม่โดยที่หากมี (DF) จะถือว่าแพ็กเกจนี้ไม่ยอมให้มีการแบ่งแพ็กเกจย่อยออกไปอีก แต่ถ้าหากไม่มีแพ็กเกจอาจถูกแบ่งออกไปอีกเพื่อประโยชน์ในการส่งแพ็กเกจ สำหรับในยูดีพีจะมีความแตกต่างกันตามบริการที่ยูดีพีให้บริการซึ่งแต่ละบริการจะมีความหมายเฉพาะแยกตามบริการดังตัวอย่าง

rwho packet:

actinide.who > broadcast.who: udp 84

Name server requests :

h2opolo.1538 > helios.domain: 3+ A? ucbvax.berkeley.edu. (37)

Name server responses :

helios.domain > h2opolo.1537: 2 NXDomain* 0/1/0 (97)

ในส่วนนี้จะไม่ขออธิบายเพราะไม่ได้ใช้ในโครงงาน นอกจากนี้ tcpdump ยังสามารถแสดงข้อมูลทั้งแพ็กเกจในรูปของแพ็กเกจเลขฐานสิบหก ในกรณีที่ใช้ออปชัน –x

secure3:/home/staff/sintoo# tcpdump –x ← คำสั่ง

tcpdump: listening on eth0

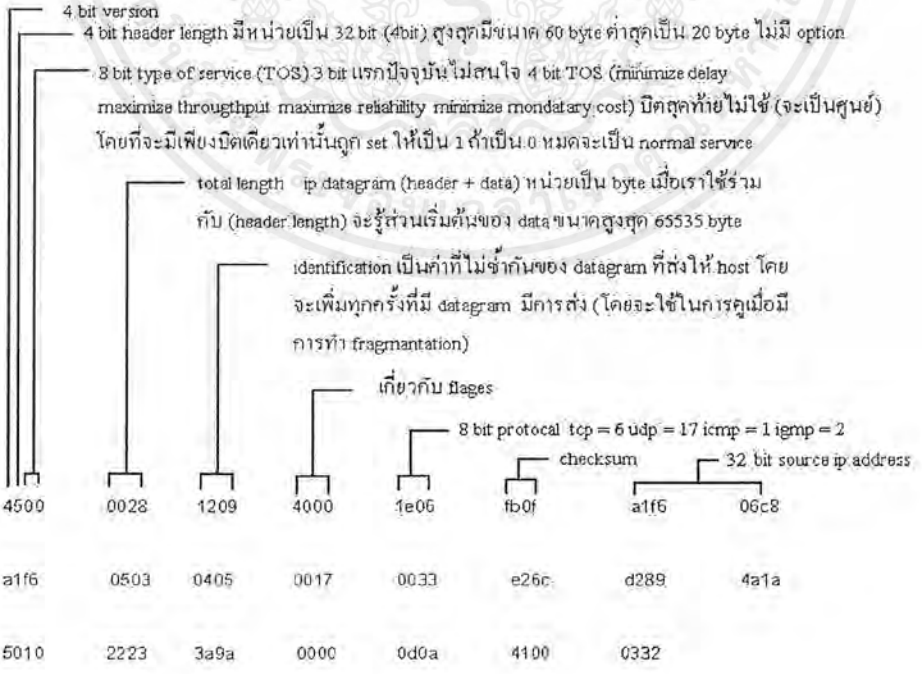
15:44:42.683233 office0.ce.kmitl.ac.th.1029 > secure3.ce.kmitl.ac.th.telnet: . ack 3532212762 win 8739 (DF)

4500 0028 1209 4000 1e06 fb0f a1f6 06c8

a1f6 0503 0405 0017 0033 e26c d289 4a1a

5010 2223 3a9a 0000 0d0a 4100 0332

ข้อมูลในส่วนของเลขฐานสิบหกจะเป็นไอพีแพ็กเกจมีความหมายดังนี้



รูปที่ 3-1 แสดงความหมายของแพ็กเกจที่ได้จากโปรแกรม tcpdump

บทที่ 4
CGI และ ภาษาเพริ์ล (Perl language)

CGI (Common Gateway Interface) เป็นตัวกลางที่จะเชื่อมต่อระหว่างเว็บเซิร์ฟเวอร์ กับฐานข้อมูลภายนอกหรือโปรแกรมประยุกต์อื่นๆ ที่อยู่ทางด้านเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์จะทำการรับข้อมูล (input) จากไคลเอนต์แล้วจากนั้นจึงส่งข้อมูลนั้นเข้าไปซีจีไอ (CGI) โปรแกรมแล้วซีจีไอโปรแกรมก็จะทำการประมวลผลข้อมูล เมื่อได้ผลลัพธ์ที่ต้องการแล้วจึงทำการส่งผลลัพธ์นั้นกลับไปยังเว็บเซิร์ฟเวอร์ เพื่อให้เว็บเซิร์ฟเวอร์ส่งไปให้ไคลเอนต์ต่อไป

การเขียนโปรแกรมซีจีไอนั้นสามารถเขียนได้โดยการใช้ภาษาคอมพิวเตอร์ใดๆ ก็ได้ และสามารถที่จะเขียนบนระบบปฏิบัติการใดก็ได้ ทั้งนี้ขึ้นอยู่กับความถนัดของผู้เขียนและขึ้นอยู่กับเซิร์ฟเวอร์ด้วยว่าทำงานอยู่ภายใต้ระบบปฏิบัติการใดซึ่งแต่ละภาษาก็มีความเหมาะสมแตกต่างกันไปตามระบบปฏิบัติการและงานที่ทำ

4.1 การเลือกระบบปฏิบัติการที่จะใช้เขียนสคริปต์

ระบบปฏิบัติการแต่ละตัวย่อมมีข้อดีข้อเสียที่แตกต่างกันไป ซึ่งถ้าหากเข้าใจถึงข้อดีข้อเสียของระบบปฏิบัติการแต่ละตัวแล้ว ก็จะเป็นผลดีในการตัดสินใจเลือก ระบบปฏิบัติการที่จะเขียนซีจีไอโปรแกรม ซึ่งมีงานบางอย่างที่สามารถทำได้ง่ายบนระบบปฏิบัติการหนึ่งแต่จะมีความยุ่งยากมากขึ้นเมื่อทำบนอีกระบบปฏิบัติการหนึ่ง ดังนั้นการที่จะเขียนซีจีไอโปรแกรมควรที่จะเลือก ระบบปฏิบัติการให้เข้ากับชนิดของงานที่เราจะทำด้วย ซึ่งพอจะสรุปได้ดังนี้

4.1.1 งานเกี่ยวกับข้อความและการค้นหา (Text Manipulation and Searching)

ถ้าหากงานที่จะนำไปประยุกต์ใช้งานเป็นงานที่เกี่ยวกับข้อความที่เป็น ASCII (ASCII Text file) ก็ควรที่จะเขียนสคริปต์โปรแกรมบนระบบปฏิบัติการยูนิกซ์ เนื่องจากยูนิกซ์ เป็น ระบบปฏิบัติการ ที่มีคำสั่งในภาษาสูงที่จะทำงานเกี่ยวกับการค้นหา, การเรียงลำดับ และการทำงานอื่น ๆ ที่เกี่ยวกับเท็กซ์มาก ทำให้เกิดความสะดวกในการเขียนโปรแกรม ส่วน ระบบปฏิบัติการดอส ก็มีความสามารถในการค้นหาข้อความเช่นกันแต่จะมีข้อจำกัดมากกว่ายูนิกซ์ อยู่มากส่วนอีกระบบปฏิบัติการ ที่จะพิจารณาก็คือ วินโดว์ไม่สามารถที่จะทำงานในโหมดของ Command Line ได้ดังนั้นจึงต้องอาศัยดอส ในการทำเป็นแบตซ์ซึ่งจะทำให้ วินโดว์ จะต้องเสียเวลาในการเปิดปิด แบตซ์ไฟล์ทำให้วินโดว์ไม่เหมาะที่จะนำมาใช้ในการทำซีจีไอโปรแกรมที่เป็นงานเกี่ยวกับการค้นหาข้อมูลที่เป็นเท็กซ์

4.1.2 งานในการประมวลผลข้อมูลที่ไม่ใช่ เท็กซ์ (Non Text Data Manipulation)

งานที่ไม่ใช่การประมวลผลข้อมูลที่เป็นเท็กซ์ แต่เป็นการประมวลผลข้อมูลที่เป็นข้อมูลของเวิร์คสเตชัน หรือ โปรแกรมสเปรดชีต ควรที่จะเขียนโปรแกรมซีจีไอที่ทำงานบน ระบบปฏิบัติการแบบวินโดว์เนื่องจากวินโดว์มีความสามารถในการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ด้วยกันเองหรือที่เรียกว่าการทำ Dynamic Data Exchange (DDE) และการทำ OLE ซึ่งจะทำให้การเขียนโปรแกรมซีจีไอ ทำได้ง่ายและสะดวกกว่าการเขียนบนระบบปฏิบัติการที่เป็นแบบยูนิกซ์ และ คอส

4.1.3 งานที่เกี่ยวกับการประมวลผลฐานข้อมูล (Database Access)

สำหรับการเลือกระบบปฏิบัติการ สำหรับการเขียนโปรแกรมซีจีไอที่จะไปทำการประมวลผลฐานข้อมูลนั้นๆ จะต้องขึ้นอยู่กับว่าฐานข้อมูลของเรานั้นเป็นระบบฐานข้อมูลที่ทำงานภายใต้ระบบปฏิบัติการอะไรซึ่งถ้าหากระบบฐานข้อมูลที่เราใช้เป็นระบบฐานข้อมูลบนยูนิกซ์ เช่น Oracle, SyBase, Informix เราก็ควรที่จะเขียนโปรแกรมซีจีไอ ที่ทำงานบนระบบปฏิบัติการยูนิกซ์ และถ้าหากระบบฐานข้อมูลของเราเป็นระบบฐานข้อมูลบน ระบบปฏิบัติการวินโดว์ เช่น Microsoft Access ก็ควรที่จะเขียนโปรแกรมซีจีไอที่ทำงานภายใต้ระบบปฏิบัติการแบบวินโดว์

4.2 หลักทั่วไปในการเขียนโปรแกรมซีจีไอ

เมื่อเซิร์ฟเวอร์รับข้อมูลจากไคลเอนต์โดยผ่านทาง URL หรือ Post Method แล้วโปรแกรมเซิร์ฟเวอร์ จนทำการส่งผ่านข้อมูลไปยังสคริปต์หรือซีจีไอโปรแกรม จากนั้นจึงสั่งให้โปรแกรมหรือสคริปต์นั้นๆ ทำงานสคริปต์หรือโปรแกรมก็จะอ่านข้อมูลที่เซิร์ฟเวอร์ส่งผ่านมาเพื่อที่จะนำไปทำการประมวลผล และเมื่อได้ผลลัพธ์จากการประมวลผลก็จะทำการส่งกลับไปยังเซิร์ฟเวอร์ทางด้านเซิร์ฟเวอร์ก็จะทำการส่งข้อมูลกลับไปยังทางด้านไคลเอนต์โดยใช้ HTTP Protocol จากหลักการทำงานของซีจีไอโปรแกรมนั้นสามารถที่จะแยกขั้นตอนการทำงานได้ 3 ขั้นตอนดังนี้

4.2.1 การรับข้อมูล (Receiving the Data)

การรับข้อมูลของซีจีไอโปรแกรมจากเซิร์ฟเวอร์ สามารถทำได้หลายวิธีด้วยกันทั้งนี้ก็ต้องขึ้นอยู่กับระบบปฏิบัติการและ HTTP Method ที่ใช้ดังรายการต่อไปนี้

CGI	GET	POST
UNIX	Environment variables	stdin
DOS	Environment variable,%1%	content file
Windows	CGI data file	content file

พิจารณาวินโดว์จะเห็นว่ามีการเปิดไฟล์ ใหม่ทั้งในกรณีของการ GET และ POST เพื่อส่งผ่านตัวแปรและข้อมูลต่างๆ ไปยังซีจีไอโปรแกรมเนื่องจากว่าวินโดว์ไม่สามารถที่จะทำงานในลักษณะของ

Environment Variable ได้ซึ่งข้อมูลและค่าตัวแปรต่าง ๆ ที่จะต้องมีใน Content file และ Data file นั้น ประกอบด้วยสิ่งต่าง ๆ ดังนี้

SERVER_SOFTWARE (ชื่อของโปรแกรม เซิร์ฟเวอร์) จะประกอบไปด้วยชื่อและรุ่นของโปรแกรมที่ทำหน้าที่เป็น เซิร์ฟเวอร์ เช่น NCSA/V.13

SERVER__NAME เป็นชื่อหรือ ไอพีแอดเดรสของเซิร์ฟเวอร์ เช่น www.kmitl.ac.th

GATEWAY_INTERFACE มาตรฐานของซีจีไอที่ เซิร์ฟเวอร์ใช้เช่น CGI/1.1

OUTPUT_FILE เป็นชื่อของแฟ้มข้อมูลของผลลัพธ์ที่ได้จากการประมวลผลซึ่งจะเป็นแฟ้มข้อมูลแบบชั่วคราวเพื่อใช้ในการผ่านผลลัพธ์ไปยังเซิร์ฟเวอร์เท่านั้น

SERVER_PROTOCOL เป็นมาตรฐานโพรโตคอลที่ทางด้านไคลเอนต์ใช้ในการส่งข้อมูลเช่น HTTP/1.0

SERVER_PORT เป็นค่าของหมายเลขพอร์ตของเซิร์ฟเวอร์ที่รับขอติดต่อ จากไคลเอนต์ เช่น 80

REQUEST_METHOD ระบุถึงกรรมวิธีการส่งข้อมูลจากไคลเอนต์มายังเซิร์ฟเวอร์ เช่น GET, POST

PATH_INFO ระบุถึง logical path name ของซีจีไอโปรแกรม

PATH_TRANSLATED ระบุถึง Physical path name ของซีจีไอโปรแกรม

SCRIPT_NAME ชื่อของโปรแกรมหรือ สคริปต์

QUERY_STRING เป็นค่าของตัวแปรหรือข้อมูลที่ผู้ใช้ป้อนเข้าไปใน ฟอร์ม (Form)

REMOTE_HOST เป็นชื่อของเครื่องที่ทำการขอการติดต่อมายังเซิร์ฟเวอร์

REMOTE_ADDR เป็นไอพีแอดเดรสของเครื่องที่ทำการ ขอการติดต่อมายัง เซิร์ฟเวอร์

CONTENT_TYPE ระบุถึงชนิดของข้อมูลที่ถูกส่งมาโดยวิธี POST

CONTENT_LENGTH ระบุถึงขนาดของข้อมูลที่ส่งมาโดยวิธี POST

CONTENT_FILE ระบุถึงชื่อแฟ้มข้อมูลชั่วคราวที่ใช้เก็บข้อมูลที่ส่งมาจากไคลเอนต์โดยวิธี POST

4.2.2 การแยกข้อมูล (Parsing the Data)

การส่งข้อมูลจากไคลเอนต์มายังเซิร์ฟเวอร์จะต้องทำการเปลี่ยนช่องว่าง ให้เป็นเครื่องหมายบวก (+) เสียก่อนและทำการเปลี่ยนเครื่องหมายทางคณิตศาสตร์และอักขระพิเศษ ให้เป็นรหัส ASCII ดังนั้นเมื่อทางด้านเซิร์ฟเวอร์จะทำการส่งผ่านข้อมูลให้กับโปรแกรมซีจีไอหรือสคริปต์ ก็จำเป็นที่จะต้องทำการเป็นเครื่องหมายบวกให้เป็นช่องวางและเปลี่ยนรหัส ASCII ให้เป็นอักขระพิเศษให้หมดก่อนแล้วทำการแยกตัวแปรและค่าของตัวแปรต่าง ๆ

4.2.3 การส่งผลลัพธ์กลับไปยังไคลเอนต์ (Returning Results)

โปรแกรมซีจีไอหรือสคริปต์สามารถที่จะส่งผลลัพธ์กลับไปยังทางไคลเอนต์โดยผ่านทางเซิร์ฟเวอร์ ได้ 3 ลักษณะด้วยกัน

- ส่งข้อมูลกลับไปในรูปของเอกสาร HTML (Return HTML Document) การส่งข้อมูลกลับแบบนี้จะเป็นการที่โปรแกรมซีจีไอหรือสคริปต์ ทำการสร้างเอกสาร HTML ขึ้นมาแล้วทำการส่งกลับไปยังไคลเอนต์โดยลักษณะของข้อมูลจะประกอบด้วยส่วนที่เป็น HTML Header และส่วนที่เป็นเอกสารเช่น

Content-type: text/html

<TITLE>Feedback</TITLE>

<H1>Feedback</H1>

Thank you for your feedback. Your comments have been forward to the appropriate personal.

- ส่งข้อมูลกลับในลักษณะของข้อมูลประเภทอื่นๆ การส่งผลลัพธ์กลับในลักษณะนี้จะเป็นการส่งข้อมูลที่เป็นแฟ้มข้อมูลประเภทต่าง ๆ เช่น เท็กซ์, เสียง, ภาพ และโปรแกรมต่าง ๆ การส่งข้อมูลกลับในลักษณะนี้สามารถทำได้โดยการส่ง HTML Header เพื่อที่จะไปบอกทางด้าน ไคลเอนต์ให้ทราบว่าข้อมูลที่จะทำการส่งต่อไปเป็นข้อมูลชนิดใด เช่น

Content-type: text/plain

Content-type: image/gif

- ส่งผลลัพธ์กลับในรูปลักษณะของ URL การส่งผลลัพธ์ในลักษณะนี้จะเป็นการที่ผลลัพธ์ที่ต้องการจะส่งกลับไปยังไคลเอนต์เป็นเพจอื่นๆ ที่อยู่บน เซิร์ฟเวอร์ เดียวกับโปรแกรมซีจีไอหรืออาจจะเป็น เซิร์ฟเวอร์ อื่น ๆ ก็ได้ โดยการส่ง URL กลับไปยังไคลเอนต์ มีรูปแบบของข้อมูลที่จะส่งกลับดังนี้

Location: protocol://hostname/path_info/

4.3 CGI สคริปต์ทำงานอย่างไร

CGI (Common Gateway Interface) เป็นมาตราฐานที่ให้โปรแกรมเมอร์ใช้เขียนโปรแกรมที่สามารถเข้าไปใช้ข้อมูลบนเซิร์ฟเวอร์ในอินเตอร์เนตอย่างเช่น เว็บเซิร์ฟเวอร์ ได้แล้วส่งข้อมูลนั้นไปให้ผู้ใช้ ซึ่งนับเป็นวิธีการที่จะทำให้เว็บสามารถติดต่อกับฐานข้อมูลนั้นไปให้ผู้ใช้ ซึ่งนับเป็นวิธีการที่จะทำให้เว็บสามารถติดต่อกับฐานข้อมูลและแหล่งข้อมูลภายนอกได้ตัวอย่างเช่น โปรแกรมเมอร์อาจใช้ซีจีไอเขียนแอปพลิเคชั่นในเว็บเพื่อให้ผู้ใช้สามารถค้นหาข้อมูลฐานข้อมูล เช่น หัวข้อข่าวต่างๆ หรือบทความเกี่ยวกับภาพยนตร์แล้วแสดงข้อมูลที่พบในเว็บเพจนี้ในรูปแบบเฮสทีเอ็มเอล นอกจากนี้ยังอาจใช้ซีจีไอในการให้ผู้ใช้กรอกแบบฟอร์มในอินเตอร์เน็ตตัวอย่างเช่น การสมัครเป็นสมาชิกจดหมายข่าวอิเล็กทรอนิกส์และนอกจากนี้โปรแกรมซีจีไอยังสามารถใช้ในการสร้าง Web spider หรือ robots ได้ด้วย ตัวอย่างต่อไปนี้เราจะดูว่าโปรแกรมซีจีไอซึ่งจะทำให้ผู้ใช้สามารถค้นหาข้อมูลจากฐานข้อมูลว่ามีการทำงานอย่างไร

1. ผู้ใช้ซึ่งติดต่อไปยัง เว็บไม่จำเป็นต้องรู้การเขียนโปรแกรมเพื่อเรียกใช้โปรแกรมซีจีไอ แต่โปรแกรมเมอร์จะเป็นผู้เขียนโปรแกรมซีจีไอนั้นเอง โดยสามารถใช้ภาษาคอมพิวเตอร์ต่างๆ ในการเขียนซีจีไอได้หลายภาษาเช่น C, C++, Fortran, Visual Basic และ AppleScript แอปพลิเคชั่นที่เขียน

ขึ้นด้วยภาษาสำหรับโปรแกรมทั่วๆ ไปบางภาษาเช่น ภาษาซี จะต้องผ่านการคอมไพล์ (คือผ่านการแปลงด้วยโปรแกรมที่เรียกว่า คอมไพล์เลอร์) เสียก่อน เพื่อที่จะให้ตัวมันเองทำงานได้ตัว คอมไพล์-เลอร์ นี้จะแปลงแอปพลิเคชั่นในภาษาดังกล่าวให้เป็นภาษาที่ซีจีไอสามารถเข้าใจได้ ส่วนภาษาแบบอื่นๆ ที่เรียกว่า Scripting languages นั้นไม่จำเป็นที่จะต้องผ่านการคอมไพล์ก่อน ซีจีไอสคริปต์นั้นจะง่ายต่อการดีบัก, แก้ไขเปลี่ยนแปลงและบำรุงรักษามากกว่าโปรแกรมที่จะต้องทำการคอมไพล์ก่อน ดังนั้นมันจึงถูกนำมาใช้เขียนโปรแกรมมากกว่าซึ่งภาษาที่นิยมใช้ในการเขียนซีจีไอสคริปต์มากที่สุดก็คือภาษาเพิร์ล

2. หลังจากที่โปรแกรมได้ถูกเขียนขึ้นและคอมไพล์เรียบร้อยแล้ว หรือสคริปต์ได้ถูกเขียนขึ้นแล้วโปรแกรมนี้จะถูกเก็บไว้ในไดเร็คทอรี่พิเศษในเว็บเซิร์ฟเวอร์ เช่น /cgi-bin เป็นต้นซึ่งไดเร็คทอรี่นี้จะเก็บรักษาโปรแกรมซีจีไอไว้ทั้งหมด ทั้งนี้ผู้ที่ดูแลระบบของเว็บเซิร์ฟเวอร์จะเป็นคนกำหนดว่าได-เร็คทอรี่ไหนที่ควรจะใช้เก็บโปรแกรมซีจีไอถ้ามีบางคนเขียนโปรแกรมซีจีไอแล้วใส่ไว้ในไดเร็คทอรี่ที่ไม่ถูกต้องโปรแกรมก็จะไม่สามารถทำงานได้ นี่ถือเป็นการรักษาความปลอดภัยอย่างหนึ่ง เพราะถ้ายอมให้เก็บโปรแกรมซีจีไอไว้ได้ในหลายๆ ไดเร็คทอรี่ก็จะเป็นการยากในการที่จะต้องติดตามดูแลโปรแกรมเหล่านั้นทั้งหมด นอกจากนี้อาจมีบุคคลภายนอกสร้างโปรแกรมซีจีไอเพื่อทำอะไรก็ไม่รู้แล้วเอามาเก็บไว้ได้ซึ่งอาจจะเป็นอันตรายต่อซอฟต์แวร์เดิมที่เก็บไว้แล้วหรืออันตรายต่อทั้งระบบก็ได้

3. หลังจากโปรแกรมซีจีไอถูกเก็บไว้ในไดเร็คทอรี่เฉพาะแล้วลิงก์ที่ชี้ไปยังตัวมันจะถูกเก็บไว้ใน URL ของเว็บเพจนั้น

4. เมื่อคุณเข้าไปใน เว็บไซต์ และคลิกตรง URL โปรแกรมซีจีไอจะถูกเรียกขึ้นมาทำงานเช่น ถ้ามันยอมให้คุณค้นหาข้อมูลในฐานข้อมูลได้มันจะส่งแบบฟอร์มในรูป HTML มาให้คุณกรอกรายละเอียดสิ่งที่คุณต้องการจะค้นหา เมื่อคุณกรอกเสร็จแล้วและคลิกที่ส่งข้อมูลในแบบฟอร์มข้อมูลก็จะถูกส่งไปยังโปรแกรมซีจีไอ

5. โปรแกรมซีจีไอจะติดต่อกับฐานข้อมูลและร้องขอข้อมูลที่ผู้ใช้ต้องการ จากนั้นฐานข้อมูลจะส่งข้อมูลไปในโปรแกรมซีจีไอซึ่งข้อมูลนี้อาจอยู่ในหลายรูปแบบเช่น ข้อความ, รูปภาพ, เสียง,ไฟล์วิดีโอและที่อยู่แบบ URL เป็นต้น

6. เมื่อโปรแกรมซีจีไอได้รับข้อมูลจากฐานข้อมูลแล้วก็จะจัดรูปแบบให้ผู้ใช้สามารถอ่านเข้าใจได้ เช่นจัดให้รูปแบบ HTML เพื่อที่จะสามารถอ่านได้ด้วย โปรแกรมบราวเซอร์แล้วจึงส่งผลที่ได้ในรูปแบบ HTML นั้นมาให้ผู้ใช้ซึ่งอาจเปิดดูได้ด้วย เว็บบราวเซอร์ เช่นเดียวกับเว็บเพจทั่วๆ ไปซึ่งผู้ใช้อาจจะคลิกไปยังลิงค์เพื่อไปเยี่ยมชมไซต์อื่น, สั่งพิมพ์ออกมาหรือดูภาพจากไฟล์กราฟฟิกและมัลติมีเดียได้ เป็นต้น

4.4 ภาษา Perl

Perl ย่อมาจาก Practical Extraction and Report Language ภาษาที่ใช้สำหรับทดลองปฏิบัติงานแยกแยะและทำรายงาน เพริ์ล เป็นภาษาคอมพิวเตอร์แบบถูกแปลในขณะที่ปฏิบัติงาน (Interpreted Language) ซึ่งหมายความว่าเราสามารถเขียนโปรแกรมภาษาเพริ์ลที่เป็นไฟล์ตัวอักษรธรรมดา (Text File) แล้วสามารถนำไปปฏิบัติงานหรือรันได้เลยโดยไม่ต้องมีการแปลหรือคอมไพล์ก่อนการรันโปรแกรม (Compiled Language) เพริ์ลเป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบและพัฒนามาเพื่อใช้ประโยชน์กับจัดการกับไฟล์โดยเฉพาะอย่างยิ่งกับไฟล์อักษร โดยการตรวจตัวอักษรต่าง ๆ ภายในไฟล์ แยกแยะส่วนประกอบตัวอักษรระบบปฏิบัติการยูนิกซ์เป็นต้น เพริ์ลมักจะถูกใช้กับงานของการบริหารระบบคอมพิวเตอร์ เช่น ระบบปฏิบัติการยูนิกซ์ เป็นต้น เพริ์ลได้ถูกออกแบบเพื่อเป็นภาษาที่ใช้กับงานของการบริหารระบบคอมพิวเตอร์มักถูกออกแบบเพื่อเป็นภาษาที่ใช้การทดลองเขียนโปรแกรมเพราะตัวภาษาสามารถเขียนได้หลายรูปแบบสามารถลองผิดลองถูกได้ (ง่ายต่อการใช้งานให้ประสิทธิภาพสูง และสมบูรณ์แบบ) โครงสร้างของเพริ์ลคล้ายกับภาษาซี เช่นการควบคุมทิศทางของการทำงานของโปรแกรม (Program Flow Control) สำหรับคุณสมบัติของการแยกแยะเปรียบเทียบรูปแบบการค้นหาข้อมูลภายในไฟล์ (Patern Matching) เพริ์ลได้แก้ไขพัฒนาจากโปรแกรม sed และ awk

ตัวอย่างการนำเอาภาษาเพริ์ลมาใช้ประโยชน์กับการสร้างโปรแกรม CGI (Common Gateway Interface) ซึ่งทำงานบนเว็บเซิร์ฟเวอร์ หรือเครื่องคอมพิวเตอร์ประสิทธิภาพสูงที่ให้บริการส่งข้อมูลโฮมเพจบนระบบ WWW (World Wide Web) ให้แก่ผู้ขอบริการหรือเว็บไคลเอนต์โดยใช้เว็บบราวเซอร์ HTML สคริปต์ทางด้านไคลเอนต์ (.html) จะทำการรับข้อมูลมีตัวอย่างดังนี้

```
<html><head><title>From Client</title></head><body>
<h2>This is the Guestbook Example:</h2><p>
<from actionhttp://hoohoo.ncsa.uiuc.edu/cg-bin/test-cg imethod="GET">
Your Name:<inpu t type=text" name="txt_box2"><br>
Your E-mail:<inpu t type=text" name="txt_box1"><p>
Your Comment:<br><textarea name="txt_area" cols="30" rows="3"></textarea><p>
<input type="radio" name="rd_box" value="NN" checked>Navigator
<input type="radio" name="rd_box" value="IE">Explorer
<input type="radio" name="rd_box" value="Lx">Lynx
<input type="radio" name="rd_box" value="OT">Other<p>
<input type="checkbox" name="ck_box" value="yes">
Do you want to add your comment in my guesbook?<p>
<input type="submit" name="sm_button" value="Submit">
<input type="reset" name="rs_button" value="Reset">
</form></body></html>
```

และเพริ์ลสคริปต์ test-cgi ที่ทำงานบน เว็บเซิร์ฟเวอร์มีตัวอย่างดังต่อไปนี้#

```
#!/usr/local/bin/perl
print "Content-type:text/html

<html>#สร้างสคริปต์ HTML
<head><title>CGI Server</title>
</head>
<body>";
while($k<y,$value)=each %ENV) { rint "\n$key=$value <br>"; #แสดงพารามิเตอร์ต่าง ๆ
}
print"</body>
</html>";
sub readInput{local($buffer,@pairs,$pair,$name,$value,%FORM);
$ENV{'REQUEST_METHOD'}=~tr/a-z/A-Z/;
if($ENV{'REQUEST_METHOD'}eq"post"){read(STDIN,$buffere,$ENV{'CONTENT_LENGTH'
});}
else
{
$buffer=$ENV{'QUERY_STRING'};
}
@pairs=split(/&/,$buffer);
foreach $pair (@pairs)
{
($name,$value)=split(/=/,$pair);
$value=~tr/+/ /;
$value=~s/%(..)/pack("C",hex($1)/eg;
$FORM{$name}=$value;
}
%FORM;
} #จบสคริปต์ test-cgi
```

จากตัวอย่างเพริ์ลสคริปต์ test-cgi จะสังเกตเห็นได้ว่าโครงสร้างโดยรวมของภาษาเพริ์ลจะคล้ายกับภาษาซี แต่เมื่อลองมองพิจารณาดูรายละเอียดอย่างลึกๆ แล้วจะเห็นได้ว่าภาษาเพริ์ลประกอบไปด้วยสัญลักษณ์คำสั่งแปลก ๆ มากมาย และยากต่อการทำความเข้าใจ ภาษาเพริ์ลเพียงหนึ่งคำสั่งอาจจะใช้คำสั่งของภาษาซีหลายสิบคำสั่งเพื่อทำงานในลักษณะเดียวกัน เป็นต้น

ภาษาเพริ์ลนั้นถือว่าเป็นเครื่องมือประสิทธิภาพสูงสำหรับผู้ดูแลระบบทั้งหลาย (System Administrators) เพื่อจัดการกับคอมพิวเตอร์และระบบเครือข่าย โดยเฉพาะในอินเตอร์เน็ต (Internet) ใช้กันแพร่หลาย

บทที่ 5
การโจมตีเครือข่าย

การโจมตีเครือข่ายหรือการบุกรุกเข้ามาในเครือข่าย มีพื้นฐานและหลักการทำงานคล้ายๆ กัน นั่นคือ อาจเกิดจากความบกพร่องของระบบปฏิบัติการ, ระบบการทำงานของเครือข่าย เช่น โพรโตคอลที่ใช้ในการติดต่อสื่อสาร ตัวอย่างเช่น ทีซีพี, ยูดีพี หรืออาจเกิดจากส่วนต่างๆ ที่เกี่ยวข้อง รวมถึงฮาร์ดแวร์และอื่นๆ ซึ่งสามารถนำมาใช้เป็นการเครื่องมือในการโจมตีเครือข่ายได้ ดังได้ยกเป็นตัวอย่างและศึกษาดังนี้

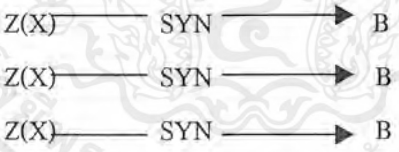
5.1 TCP SYN Flooding

TCP SYN Flooding เป็นการโจมตีที่เกิดจากข้อบกพร่องของโพรโตคอลทีซีพี/ไอพี โดยจะถือเอาความไม่สมบูรณ์ของ three-way handshake เป็นหลัก ดังมีรายละเอียดและขั้นตอนการทำงานดังนี้

5.1.1 การทำงานของ TCP SYN Flooding

-ขั้นตอนที่ 1

โฮสต์ (Host) ที่จะทำการโจมตีจะทำการปลอมหมายเลขไอพีต้นทางเป็นโฮสต์ที่ในขณะนั้นปิดบริการอยู่ไม่สามารถตอบรับแพ็กเกจที่ระบุส่งถึงโฮสต์นั้นได้ เพื่อทำการส่งแพ็กเกจที่มีการกำหนดแฟล็ก SYN ดังรูป



Z(x) หมายถึง โฮสต์ Z ทำการปลอมหมายเลขไอพีต้นทางเป็นโฮสต์ X แล้วทำการส่งแพ็กเกจที่มีการกำหนดแฟล็ก SYN ไปยังโฮสต์เป้าหมาย คือ โฮสต์ B

-ขั้นตอนที่ 2

โฮสต์ B เมื่อได้รับการร้องขอการติดต่อ (ได้รับแพ็กเกจแฟล็ก SYN) ถ้าเป็นบริการที่โฮสต์นั้นให้บริการอยู่นั่นคือมีการเปิดพอร์ตเพื่อรอรับการขอติดต่อ ก็จะทำสร้างแพ็กเกจที่ทำการตอบกลับไปด้วยโดยจะมีการกำหนดแฟล็กเป็น SYN, ACK ไปยังโฮสต์ X เพราะหมายเลขไอพีที่ได้รับจากแพ็กเกจแฟล็ก SYN เป็นไอพีของโฮสต์ X แต่ถ้าโฮสต์ B ไม่เปิดบริการบริการนั้นก็จะส่งแพ็กเกจที่กำหนดแฟล็กเป็น RST ไปยังโฮสต์ X เพื่อยกเลิกการขอติดต่อทำให้การโจมตีไม่มีผล

X ◄——— SYN ,ACK ——— B

X ◄——— SYN ,ACK ——— B

ในขั้นตอนนี้ชั้นทีซีพีจะสร้างแพ็กเกจที่มีแฟล็กเป็น SYN, ACK เพื่อตอบรับแพ็กเกจ SYN เมื่อแพ็กเกจถูกผ่านลงไปยังชั้นของไอพีในตอนแรกจะมีการส่งแพ็กเกจออกไปจริงๆ แต่เนื่องจากโฮสต์ X ไม่เปิดบริการหรือไม่สามารถตอบกลับได้ เราท์เตอร์ที่อยู่ใกล้จะทราบว่าโฮสต์ X ไม่เปิดและเราท์เตอร์จะทำการรายงานข้อผิดพลาดกับมาที่โฮสต์ B ว่าไม่สามารถส่งแพ็กเกจถึงโฮสต์ X ได้ ในชั้นของไอพีจะไม่ทำการส่งแพ็กเกจออกไปอีกแต่ในชั้นของทีซีพีซึ่งมีคิวที่จัดการขอติดต่อที่ยังไม่สมบูรณ์อยู่ (ยังไม่ครบ three-way handshake) ไม่ทราบว่าโฮสต์ X ส่งแพ็กเกจไปไม่ถึง จะคิดว่าเกิดข้อผิดพลาดขึ้น (Error) ก็จะทำการสร้างแพ็กเกจใหม่แล้วส่งลงมายังชั้นของไอพีใหม่ ซึ่งชั้นของไอพีก็จะไม่ทำการส่งแพ็กเกจออกไปเพราะทราบแล้วว่าแพ็กเกจไปไม่ถึงยังโฮสต์นั้น

-ขั้นตอนที่ 3

ปัญหาจะเกิดเมื่อคิวที่ใช้รับการขอติดต่อ (ไม่ครบ three-way handshake) เต็มทีซีพีจะไม่รับการขอติดต่อเข้ามาใหม่ เพราะว่าคิวเต็มทีซีพีก็จะพยายามทำกระบวนการขอติดต่อ (three-way - handshake) ให้สมบูรณ์ นั่นคือรอ ACK จากโฮสต์ X ซึ่งไม่มีวันจะส่งมาถึงโฮสต์ B จะทำการรอจนกว่าจะหมดเวลาที่ตั้งไว้ (time out) แล้วโฮสต์ B จะส่งแพ็กเกจที่กำหนดแฟล็ก RST ออกไปเพื่อยกเลิกการติดต่อนั้น

: B รอ ACK

: B รอ ACK

◄——— RST ——— : B (time out)

เมื่อ time out คิวก็จะว่างลง 1 คิวเพื่อรอรับขอการติดต่อใหม่แพ็กเกจชุดใหม่ก็จะโจมตีเข้ามาอีกทำให้คิวเต็ม

5.1.2 คิวของการรอรับการขอติดต่อ

1. ในชั้นของทีซีพีแต่ละจุดที่มีการให้บริการจะมีการรอรับการขอติดต่อ (Listening) โดยใช้คิวขนาดคงที่รอรับการขอติดต่อในชั้นของทีซีพี

2. ในชั้นของแอปพลิเคชั่นจะกำหนดความยาวของ คิว (backlog) เพื่อรอรับและจัดการการขอการติดต่อซึ่งมีค่าแตกต่างกันไปตามแอปพลิเคชั่น

3. เมื่อการร้องขอการติดต่อเข้ามา (------- SYN ----->) ในชั้นของทีซีพีจะดูค่าของ Listening ถ้าค่ามากกว่า 0 ก็จะยอมให้กระทำกระบวนการติดต่อ (three way handshake) โดยทั่วไปค่านี้จะมีความ

สัมพันธ์กับค่า Backlog และจะลดค่าลงหนึ่งเมื่อมีการย่อมรับ การขอติดต่อได้ถูกสร้างขึ้นแล้วก็จะเพิ่มค่าขึ้นหนึ่งแล้วขอ การ listening ว่าจะยอมรับการติดต่อ หรือไม่

- ค่าของ backlog ถูกกำหนดโดยแอปพลิเคชั่นเพื่อให้ค่ามากสุดของคิวที่ยอมให้ ติดต่อที่จุดนั้น ๆ

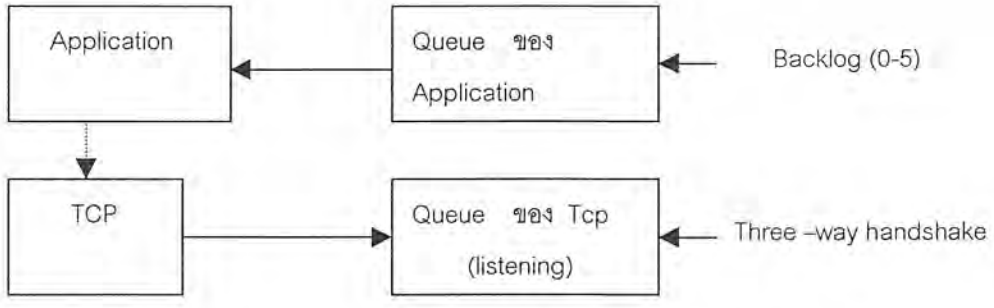
Backlog value	Max # of queued connection BSD	Solaris 2.2
0	1	0
1	2	1
2	4	2
3	5	3
4	7	4
5	8	5

ตารางที่ 5-1 แสดงขนาดของ backlog คิว

4. ค่าของ Backlog จะกำหนด คิวของการขอติดต่อเฉพาะจุดของ (ทีซีพี ⟶ แอปพลิเคชั่น) นั้นเท่านั้นไม่ใช่คิวของระบบ

5. ทีซีพีจะจัดการส่ง SYN, ACK และจะจัดการรอจนกว่าการติดต่อจะถูกจัดตั้งขึ้น แอปพลิเคชั่นที่เซิร์ฟเวอร์จะมองไม่เห็นการขอติดต่อใหม่นี้จนกว่าจะได้รับ ACK ของกระบวนการทำ three-way handshake ถ้าไม่มีคิวสำหรับ listening การขอติดต่อที่เข้ามาใหม่ทีซีพีจะไม่สนใจ SYN ที่เข้ามาและจะไม่ส่งอะไรตอบกลับ (รวมทั้ง RST)

คิวของทีซีพีและ แอปพลิเคชั่น ที่ใช้รับการติดต่อจะแยกกัน โดยคิวของทีซีพีตัวระบบโอเอสจะเป็นตัวควบคุมและจัดการ



รูปที่ 5-1 แสดงการจัดการคิวของทีซีพี และแอปพลิเคชั่น แยกกัน

5.1.3 ผลกระทบจาก TCP SYN Flooding

เป้าหมายของการโจมตีด้วย TCP SYN Flooding นั้นจะทำให้โฮสต์และพอร์ตหรือบริการที่โดนโจมตีจะไม่สามารถตอบรับการร้องขอการติดต่อที่เข้ามาที่พอร์ตหรือบริการนั้นได้แสมือนว่าโฮสต์ที่ถูกโจมตีไม่ได้เปิดให้บริการนั้น

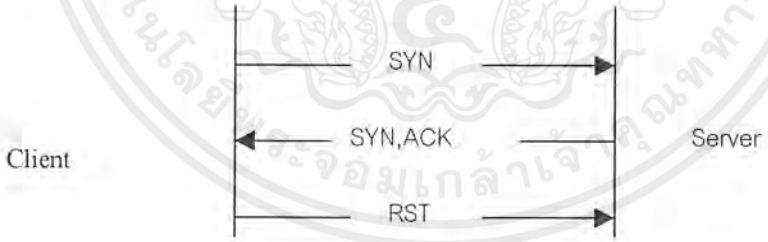
5.2. Scan Port

การสแกนพอร์ตเป็นการหาข้อมูลของโฮสต์ที่ถูกสแกนว่าเปิดให้บริการบริการอะไรบ้าง ยกตัวอย่างเช่น WWW, ftp, E-mail เพื่อที่ผู้ที่ทำการสแกนจะได้ทราบข้อมูลของโฮสต์นั้น โดยข้อมูลที่ได้ไปนี้อาจจะทำให้เกิดผลเสียหายกับโฮสต์ที่ถูกสแกนเช่น อาจมีการสั่งโจมตีบริการที่ให้บริการอยู่ซึ่งอาจจะทำให้บริการนั้นไม่สามารถให้บริการได้และยิ่งกว่านั้นการตรวจสอบหาที่มานั้นทำได้ยาก เช่น TCP SYN Flooding หรืออาจจะทำการบุกรุกเข้ามาในระบบโดยอาศัยข้อมูลที่ได้จากการ สแกนมาเป็นส่วนช่วย ซึ่งไม่เป็นผลดีกับโฮสต์นั้นๆ

5.2.1 การทำงานของ Scan Port

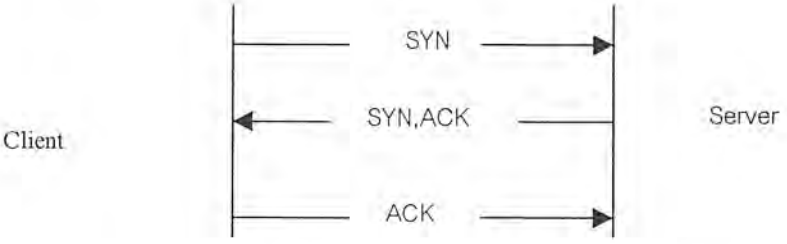
การสแกนพอร์ตแบ่งตามการทำงานออกเป็น 2 ชนิดคือ Full Scan และ Half Scan ซึ่งทั้ง 2 แบบมีความแตกต่างกันในแพ็กเกจที่ 3 นั่นคือ ACK ในแบบ Full Scan และ RST ในแบบ Half Scan ซึ่งจะเป็นตัวบอกว่ามีการติดต่อกันเกิดขึ้นหรือไม่ ดังรูป

-Half Scan



รูปที่ 5.2 แสดงการ Scan Port แบบ Half Scan

-Full Scan



รูปที่ 5.3 แสดงการ Scan Port แบบ Full Scan

การสแกนพอร์ตมีลักษณะเหมือนกับการขอติดต่อขอเข้าใช้บริการต่างๆ ของโฮสต์ที่ให้บริการโดยที่ การขอใช้บริการเหล่านั้นต้องการทราบว่าโฮสต์นั้นเปิดให้บริการอะไรบ้าง โดยจะทำการร้องขอการติดต่อมาที่บริการต่างๆ จนทราบว่าโฮสต์นั้นให้บริการอะไรบ้าง

การสแกนพอร์ตแบบ Full Scan จะเป็นมีการติดต่อกันเกิดขึ้นจริง ในบางโฮสต์อาจจะมีการบันทึกการติดต่อที่เกิดขึ้น (Logfile System) ทำให้การสแกนพอร์ตแบบ Full Scan สามารถตรวจได้ง่าย ส่วนแบบ Half Scan จะไม่มีการติดต่อเกิดขึ้นจริง โฮสต์ส่วนมากจะไม่ทำการบันทึกข้อมูลลงล็อกไฟล์ของระบบ

5.3 IP-Spoofing

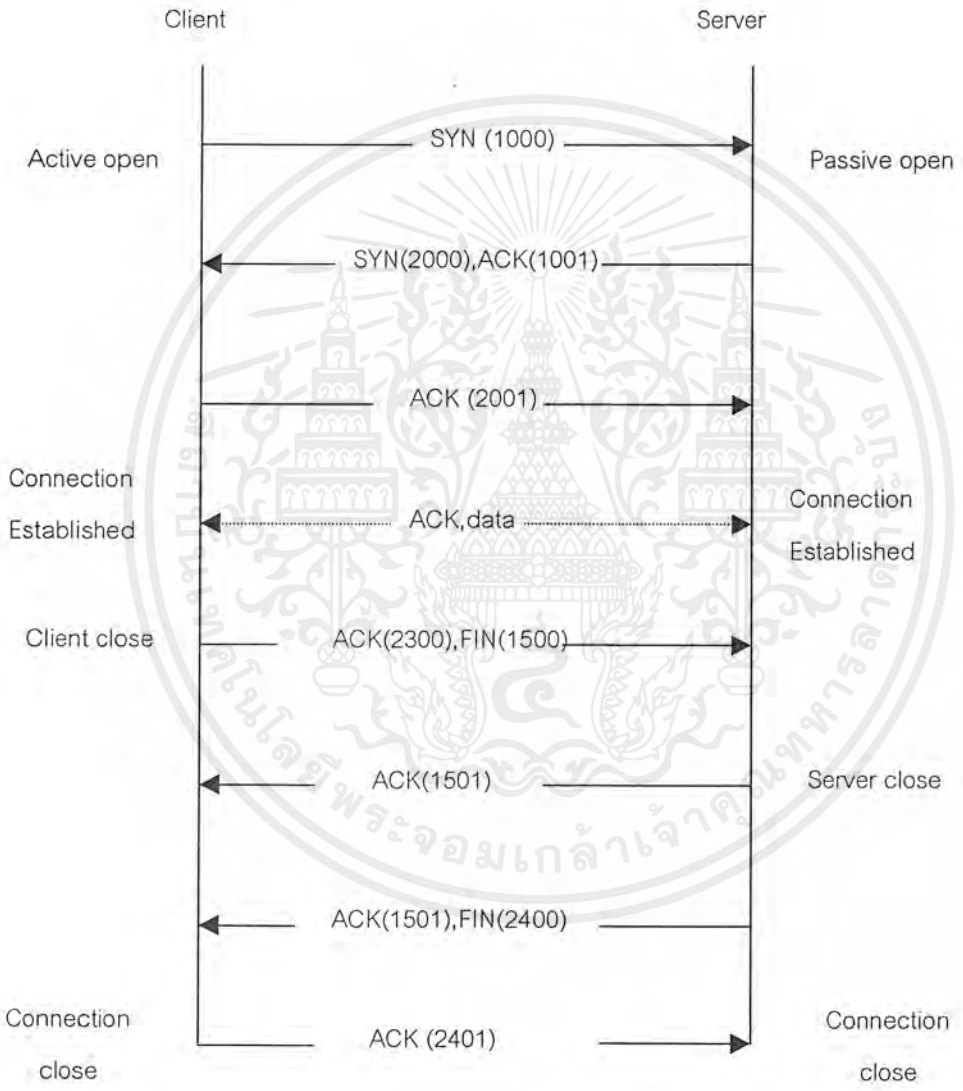
ในการโจมตีเครือข่ายส่วนมากมักจะมีการปลอมหมายเลขไอพีเป็นส่วนประกอบเสมอ ยกตัวอย่างการปลอมไอพีใน TCP SYN Flooding การปลอมไอพีจะทำให้โฮสต์ที่ถูกโจมตีไม่รู้ว่าเป็นการส่งแพ็กเกจมาจากที่ไหน หรือทำให้เข้าใจผิดเพื่อใช้ในการผ่านไฟล์วอล์ล

การปลอมไอพีคือ การที่แพ็กเกจที่ออกจากโฮสต์ที่ถูกกำหนดหมายเลขไอพีหนึ่งเอาไว้ เช่น เป็น 161.246.6.220 โปรแกรมหรือเครื่องมือบางอย่างสามารถจะทำการสร้างแพ็กเกจขึ้นมาโดยยอมให้ทำการกำหนดค่าของหมายเลขไอพีได้ ซึ่งค่าที่กำหนดนี้อาจจะเป็นค่าที่ถูกต้องหรืออาจจะไม่ถูกต้องก็ได้ เช่น เครื่องหมายเลข ไอพี 161.246.6.220 ทำการกำหนดหมายเลขไอพีต้นทางเป็นไอพี 161.246.6.215 ก็จะทำให้ทางด้านโฮสต์ที่ได้รับแพ็กเกจเข้าใจผิดว่าเป็นแพ็กเกจที่มาจากหมายเลขไอพี 161.246.6.215 ซึ่งจริงๆ แล้วเป็นแพ็กเกจที่ส่งออกมาจากเครื่องไอพีหมายเลข 161.246.6.220

บทที่ 6
การออกแบบและการสร้าง

6.1 การบันทึกการติดต่อในเครือข่าย

การบันทึกการติดต่อที่เกิดขึ้นภายในเครือข่ายจำเป็นต้องเข้าใจการทำงานของการส่งแพ็กเกจที่ทำการติดต่อ แอปพลิเคชั่นที่ทำงานโดยใช้ทีซีพีโพรโตคอลจะมีลำดับการติดต่อของแพ็กเกจดังนี้



รูปที่ 6-1 แสดงการติดต่อที่ใช้ทีซีพีโปรโตคอล

การติดต่อจะเริ่มจากแอปพลิเคชั่นที่ฝั่งไคลเอนต์ ทำการส่งแพ็กเกจที่มีการกำหนดแฟล็กให้เป็น SYN ("SYNchronize" หรือ open request ") พร้อมกับส่งค่า Initial Sequence Number (ISN) ซึ่งได้มาจากการสุ่ม (random) ไปยังฝั่งเซิร์ฟเวอร์ เพื่อเป็นการร้องขอการติดต่อ

หลังจากฝั่งเซิร์ฟเวอร์ได้รับแพ็กเกจที่มีแฟล็ก SYN I แล้วเซิร์ฟเวอร์จะทำการตอบกลับด้วยแพ็กเกจ SYN (ISN I), ACK (ISN I+1) ซึ่งมีการสุ่มค่า ISN และเพิ่มแฟล็ก ACK เพื่อตอบรับแพ็กเกจ SYN (ISNI) ด้วย ISNI+1

ฝั่งไคลเอนต์ เมื่อได้รับแพ็กเกจ SYN (ISNI) ACK (ISNI+1) ก็จะทำการตอบรับด้วยการส่ง ACK (ISNI+1) กลับไปที่ฝั่งเซิร์ฟเวอร์

การทำงานของการรับส่ง 3 แพ็กเกจเพื่อทำการเริ่มต้นการติดต่อนี้จะถูกเรียกว่า "three-way handshake"

หลังจากกระบวนการ three-way handshake เสร็จสมบูรณ์จะมีการแลกเปลี่ยนข้อมูลก็จะเริ่มขึ้น

เมื่อการต้องการเลิกติดต่อ ในกรณีนี้ให้ฝั่งไคลเอนต์เป็นฝั่งเริ่มขอจบการติดต่อก่อน ไคลเอนต์จะส่งแพ็กเกจที่มีการกำหนดแฟล็กเป็น FIN พร้อมกับ ACK ดังรูปที่ 6-1 เมื่อฝั่งเซิร์ฟเวอร์ได้รับแพ็กเกจที่มีแฟล็ก FIN เซิร์ฟเวอร์จะทำการตอบด้วยแพ็กเกจ ACK ไปก่อน แล้วตามด้วยแพ็กเกจ ACK FIN เพื่อทำการขอปิดการติดต่อเมื่อฝั่งไคลเอนต์ได้รับแพ็กเกจที่มีแฟล็ก FIN ก็จะทำการตอบกลับด้วย ACK การปิดการติดต่อก็จะสมบูรณ์

ตัวอย่างแพ็กเกจที่ทำการเปิดและปิดการติดต่อที่ได้จาก tcpdump

911482869.041368 161.246.6.81.1268 > 161.246.10.21.23 S 80455712 80455712(0) win 8192 <mss 1460> (DF)

911482869.043840 161.246.10.21.23 > 161.246.6.81.1268 S 3852032629 3852032629(0) ack 80455713 win 8760 <mss 1460> (DF)

911482885.595404 161.246.10.21.23 > 161.246.6.81.1268 FP 3852033645 3852033647(2) ack 80455790 win 8760 (DF)

911482886.741206 161.246.6.81.1268 > 161.246.10.21.23 F 80455790 80455790(0) ack 3852033648 win 7743 (DF)

6.1.1 หลักการและการออกแบบการบันทึกการติดต่อในเครือข่าย

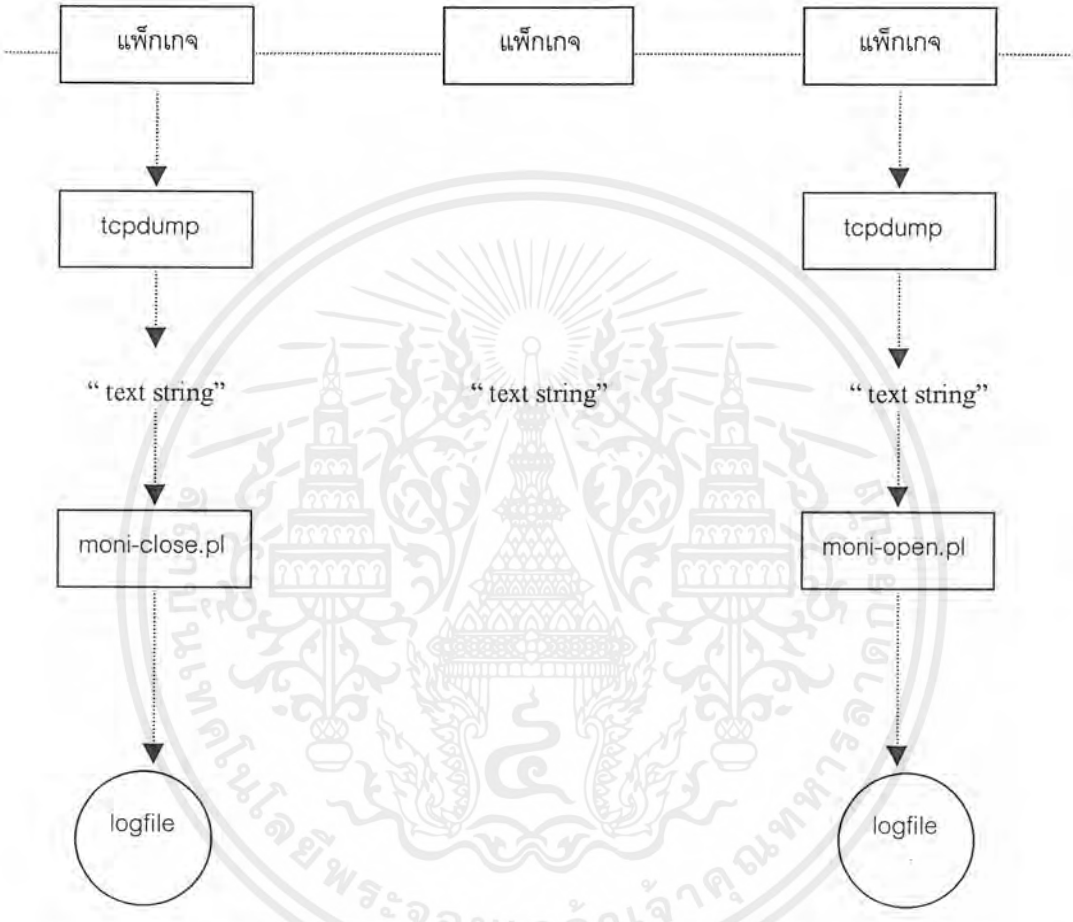
การบันทึกการติดต่อที่เกิดขึ้นภายในเครือข่ายสามารถทราบได้จากแพ็กเกจที่ส่งเพื่อทำการติดต่อกัน โดยแต่ละแพ็กเกจจะมีการกำหนดแฟล็ก และลำดับการส่งแพ็กเกจดังรูปที่ 6-1 ทำให้เราสามารถกำหนดการทำงานของโปรแกรมเพื่อบันทึกการติดต่อตามแฟล็กในแต่ละแพ็กเกจที่ได้รับดังนี้

S	S,ack	. ,ack	F/FP,ack	R
- เก็บการติดต่อใหม่ลงในหน่วยความจำ state = open1 time=time	-ถ้าเป็นการติดต่อที่เข้ามาใหม่ (ไม่พบ S) ให้เก็บการติดต่อลงในหน่วยความจำ state=open2 time=time -ถ้าเป็นการติดต่อเก่า (พบแพ็กเกจ S แล้ว state = open2	- ถ้าเป็นการติดต่อใหม่จะทิ้งแพ็กเกจนี้ไป - ถ้าเป็นการติดต่อเก่า(พบ S,ack แล้ว) state = open3 แล้วทำการบันทึกข้อมูลลงล็อกไฟล์	- ถ้าเป็นการติดต่อใหม่จะทิ้งแพ็กเกจนี้ไป -ถ้าเป็นการติดต่อเก่าทำการเซ็ตเพื่อรอการปิดการจบการติดต่อ เมื่อการติดต่อปิดโดยสมบูรณ์แล้ว บันทึกข้อมูลลงในล็อกไฟล์และลบการติดต่อนั้นออกจากหน่วยความจำ	- ถ้าเป็นการติดต่อใหม่ทิ้งแพ็กเกจนี้ใหม่ - ถ้าเป็นการติดต่อเก่าให้บันทึกลงล็อกไพล์และลบการติดต่อนั้นออกจากหน่วยความจำ

ตารางที่ 6-1 แสดงการทำงานของโปรแกรมบันทึกการติดต่อตามแฟล็กของแพ็กเกจ

การบันทึกการติดต่อจะแยกเป็น 2 ส่วน คือ ส่วนที่ 1 บันทึกการเปิดการติดต่อโดยโปรแกรมที่ทำหน้าที่นี้คือ moni-open.pl และอีกส่วนจะทำการบันทึกการปิดการติดต่อโดยโปรแกรม moni-close.pl จะทำการดูแลในส่วนนี้ ผังการทำงานของส่วนบันทึกการติดต่อจะเป็นดังรูปที่ 6-2

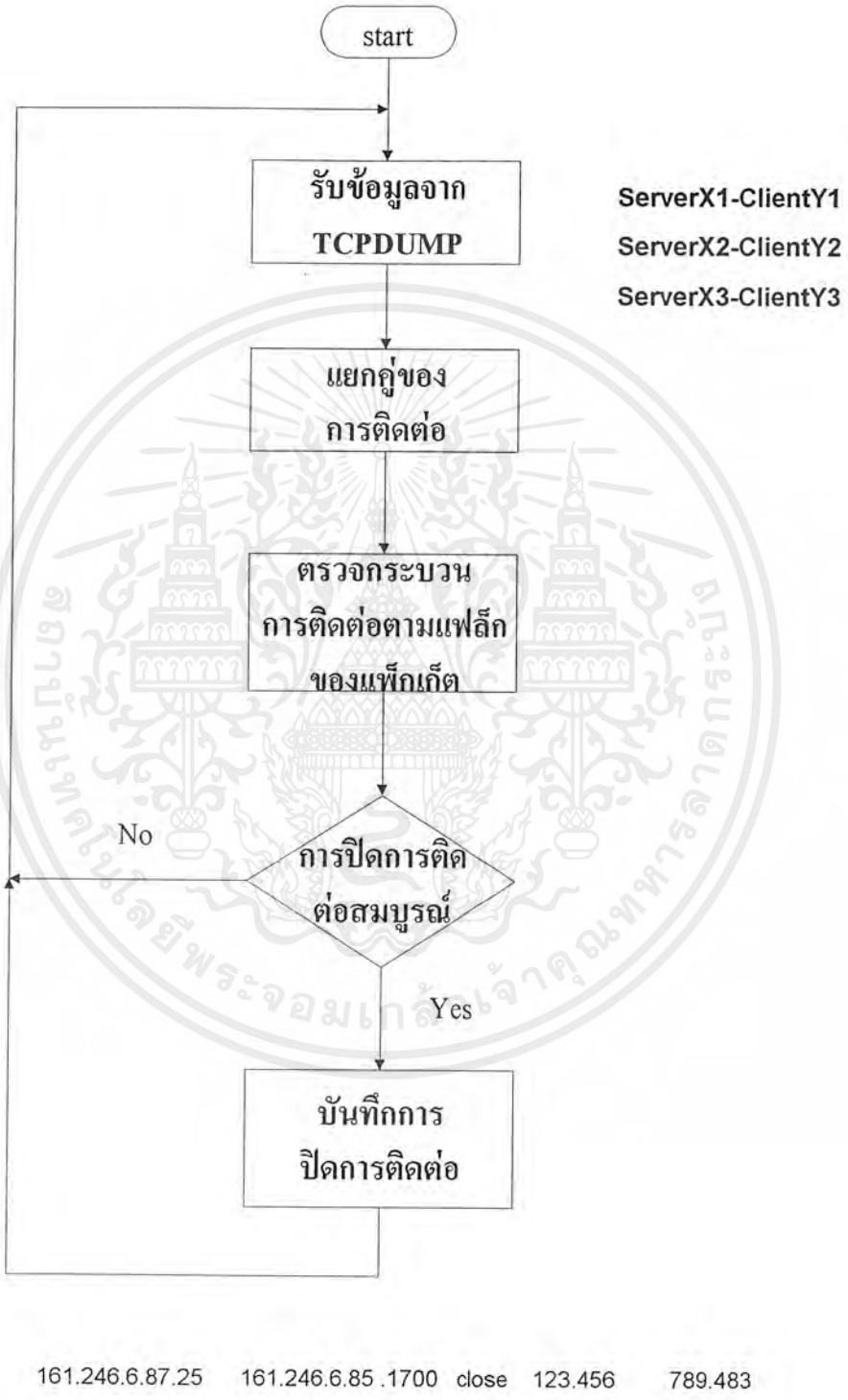
สาเหตุที่แบ่งการบันทึกออกเป็น 2 ส่วนเพราะในเวลาที่ต้องการการสอบถามการติดต่อที่ยังคงเปิดอยู่ในปัจจุบัน จะสามารถตรวจดูได้จากข้อมูลของการเปิดการติดต่อเปรียบเทียบกับข้อมูลของการปิดการติดต่อ ก็จะทำให้ทราบว่าการติดต่อใดเปิดขึ้นแล้วแต่ยังไม่ปิดลง



รูปที่ 6-2 แสดงการทำงานการบันทึกการติดต่อภายเครือข่ายเพื่อบันทึกการเปิดและการปิดการติดต่อ

6.1.2 การทำงานของโปรแกรมที่บันทึกการปิดการติดต่อในเครือข่าย

การทำงานของส่วนนี้จะใช้โปรแกรมที่เขียนด้วยเพริ์ลชื่อ moni-closed.pl ซึ่งมีการทำงานดังรูป 6-3



รูปที่ 6-3 แสดงผังการทำงานของโปรแกรมบันทึกการปิดติดต่อภายเครือข่าย

รายละเอียดเกี่ยวกับโปรแกรม moni-closed.pl มีดังนี้

- ลักษณะ

ทำการบันทึกการติดต่อ ภายในเครือข่ายและบันทึกการติดต่อที่ปิดแล้วเก็บไว้ในไฟล์เพื่อใช้เป็นข้อมูลสำหรับสอบถามในส่วนการแสดงผล

- วิธีการตรวจจับข้อมูล

ดึงโปรแกรม tcpdump –tt –S –n 'tcp' ขึ้นมารันเพื่อป้อนอินพุตให้กับตัวเองโดยจะเฝ้าดูการการติดต่อ ภายในเครือข่ายตั้งแต่มีการขอการติดต่อ (S) จนกระทั่งการติดต่อ นั้นจบไป (F,R) แล้วบันทึกการติดต่อที่ปิดแล้วลงล็อกไฟล์ ชื่อ closed.dat และเมื่อมีการเปลี่ยนวันที่ จะเปลี่ยนชื่อไฟล์ closed.dat ไปเป็นไฟล์ closedxx-xx-xxxx (วัน-เดือน-ปี) และเริ่มทำการเก็บการติดต่อ ของวันใหม่

- ผลของโปรแกรม

การติดต่อ ภายในเครือข่ายจะถูกเขียนลงไฟล์ closed.dat 1 บรรทัดต่อ 1 การติดต่อ ดังตัวอย่าง

161.246.6.87.139 161.246.6.85.1700 close 918.234 567.890

แฟล็กของการปิดการติดต่อมีความหมายดังนี้

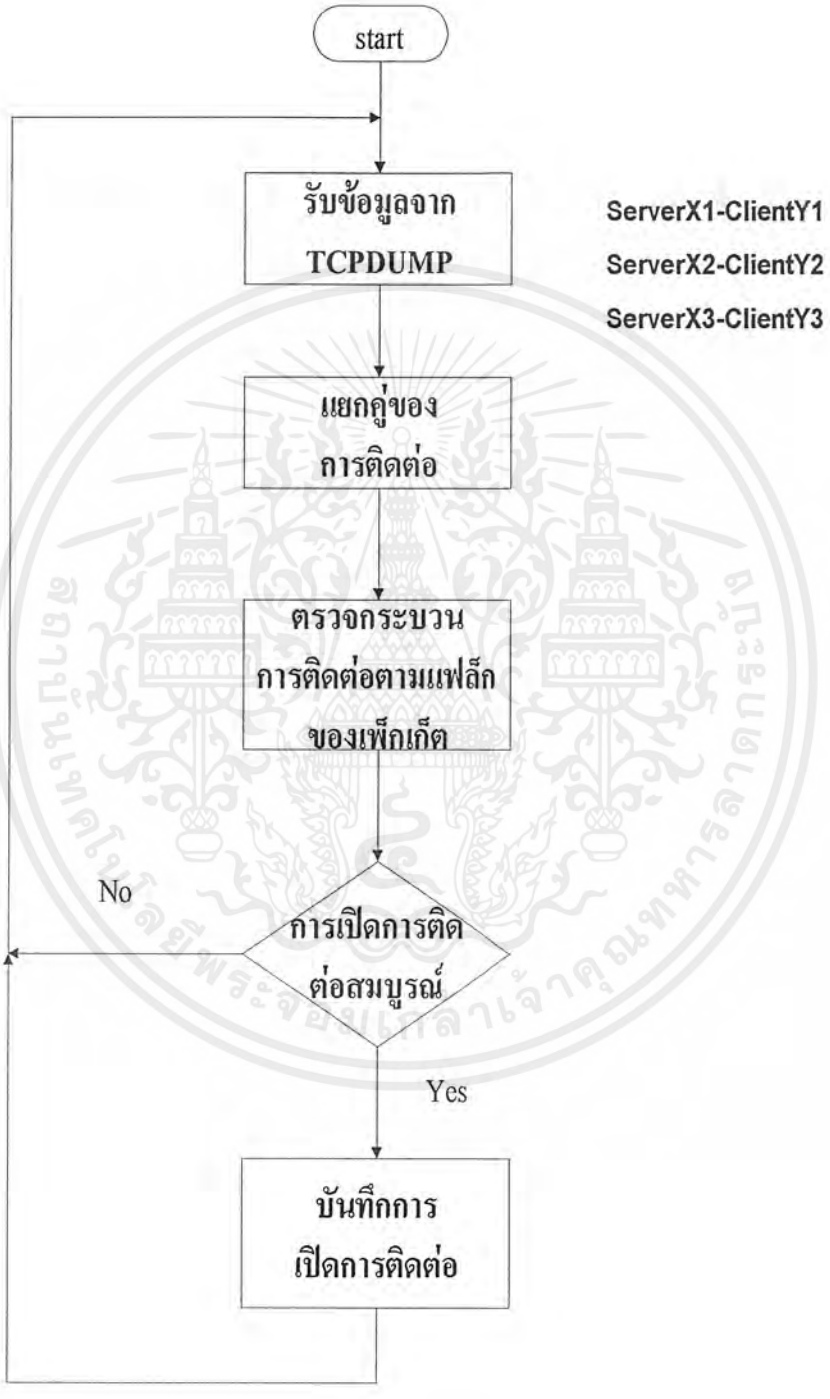
close	การปิดการติดต่อสมบูรณ์
RC_Close	การปิดการติดต่อโดยไคลเอนต์ส่ง RST หลังจาก 3-way handshake เสร็จ
RC_Open	การปิดการติดต่อโดยไคลเอนต์ส่ง RST ก่อนที่ 3-way handshake เสร็จ
RS_Close	การปิดการติดต่อโดยเซิร์ฟเวอร์ส่ง RST หลังจาก 3-way handshake เสร็จ
RS_Open	การปิดการติดต่อโดยเซิร์ฟเวอร์ส่ง RESET ก่อนที่ 3-way handshake เสร็จ

- การเรียกใช้โปรแกรม

มีการส่งรันโปรแกรม " moni-closed.pl"

6.1.3 การทำงานของโปรแกรมที่บันทึกการเปิดการติดต่อในเครือข่าย

การทำงานของส่วนนี้จะใช้โปรแกรมที่เขียนด้วยเพริ์ล ชื่อ moni-open.pl ซึ่งมีการทำงานดังรูป 6-4



161.246.6.87.25 161.246.6.85 .1700 open3 123.456 789.483

รูปที่ 6-4 แสดงผังการทำงานของโปรแกรมบันทึกการเปิดติดต่อภายเครือข่าย

รายละเอียดเกี่ยวกับโปรแกรม moni-open.pl มีดังนี้

- ลักษณะ

ทำการบันทึกการติดต่อ ภายในเครือข่ายตั้งแต่เริ่มจนกระทั่งการเปิดการติดต่อ สมบูรณ์ (three-way handshake สมบูรณ์) แล้วบันทึกการเปิดการติดต่อนั้นลงล็อกไฟล์

- วิธีการตรวจจับ

ดึงโปรแกรม tcpdump –tt –S –n 'tcp' ขึ้นมารันเพื่อป้อนข้อมูล ให้กับตัวเองโดยจะทำการตรวจสอบการติดต่อตาม state 1-3 เพื่อดูการเปิดการติดต่อแล้วบันทึกการเปิดการติดต่อนั้นลงล็อกไฟล์ชื่อ open.dat เมื่อมีการเปลี่ยนแปลงวันที่จะทำการเปลี่ยนชื่อไฟล์ open.dat ไปเป็นไฟล์ open.old และเริ่มทำการบันทึกการเปิดการติดต่อของวันใหม่

- ผลของโปรแกรม

การติดต่อ ภายใน เครือข่ายที่ เปิดการติดต่อ แล้วจะถูกบันทึกลงไฟล์ open.dat 1 บรรทัดต่อ 1 การเปิดการติดต่อ

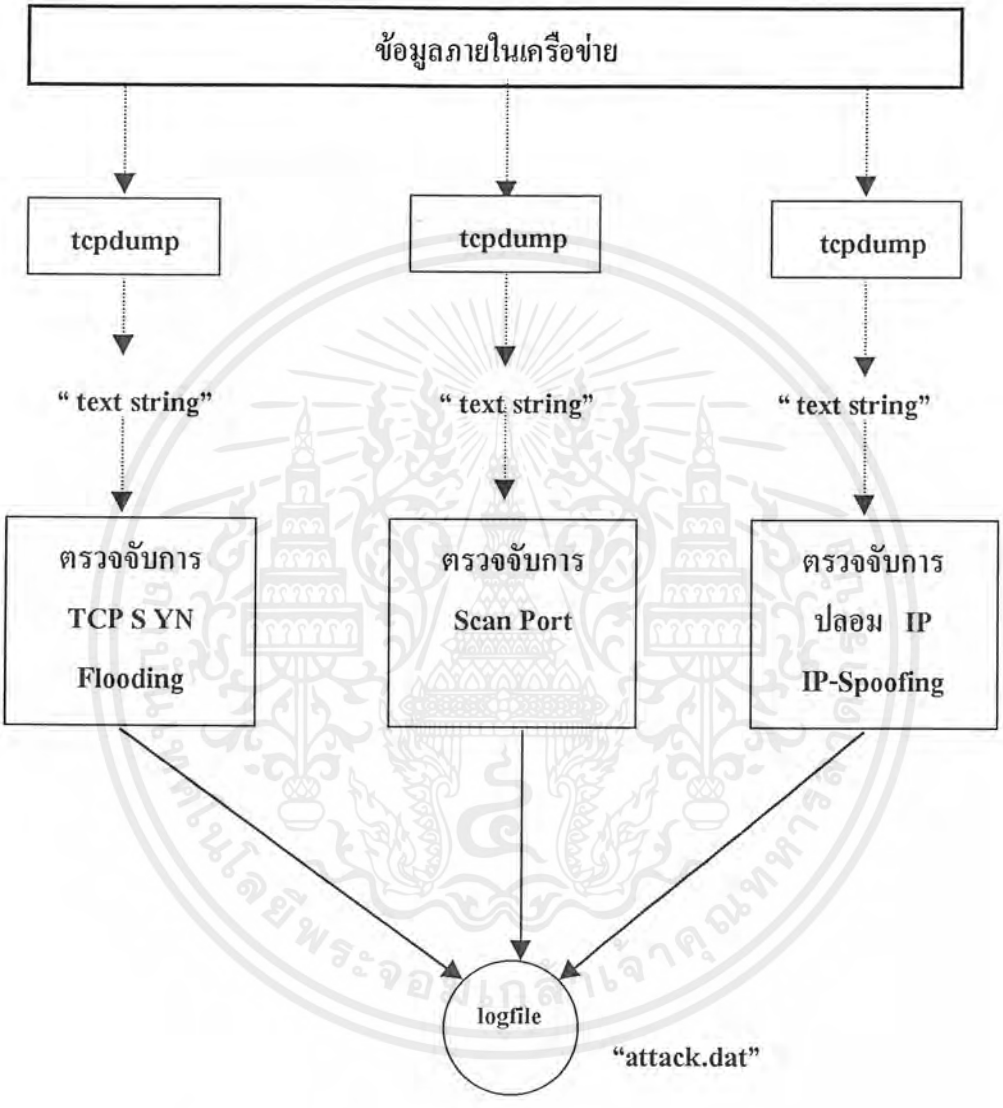
161.246.6.87.139	161.246.6.85.1700	Open3	123.456	789
server-ip.port	client-ip.port	แฟล็ก	เริ่มติดต่อเมื่อ	เปิดการติดต่อเสร็จเมื่อเวลา 789

- การเรียกใช้งานโปรแกรม

มีการสั่งรัน "moni-open.pl" ควรจะอธิบายเรื่องเปิด moni-open.pl ก่อน

6.2 การวิเคราะห์และตรวจจับการบุกรุกภายในเครือข่าย

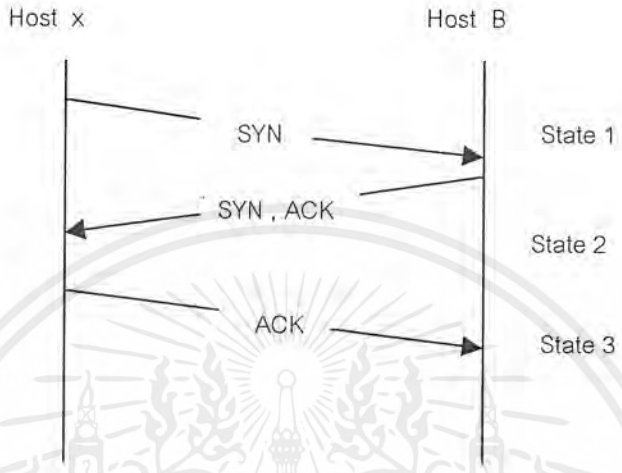
ปริญญานิพนธ์นี้ได้เลือกการทำการตรวจจับ การบุกรุกและสิ่งที่เกี่ยวข้องมา 3 อย่างเพื่อเป็นกรณีการศึกษาคือTCP SYN Flooding, สแกนพอร์ตและการปลอมไอพี (IP-Spoofing) โดยมีรายละเอียดของและผังการทำงานดังนี้



รูปที่ 6-5 แสดงการไหลของข้อมูลในส่วนของการตรวจจับการบุกรุก

6.2.1 การตรวจจับ TCP SYN FLOODING

ในการตรวจสอบหา TCP SYN Flooding จะประกอบด้วยการกระบวนการ three-way handshake ซึ่งเราจะกำหนด State1, State2 และ State3 แทนแพ็กเกจที่กระทำกระบวนการนี้ เพื่อจะใช้ในการตรวจหา TCP SYN Flooding ดังรูป



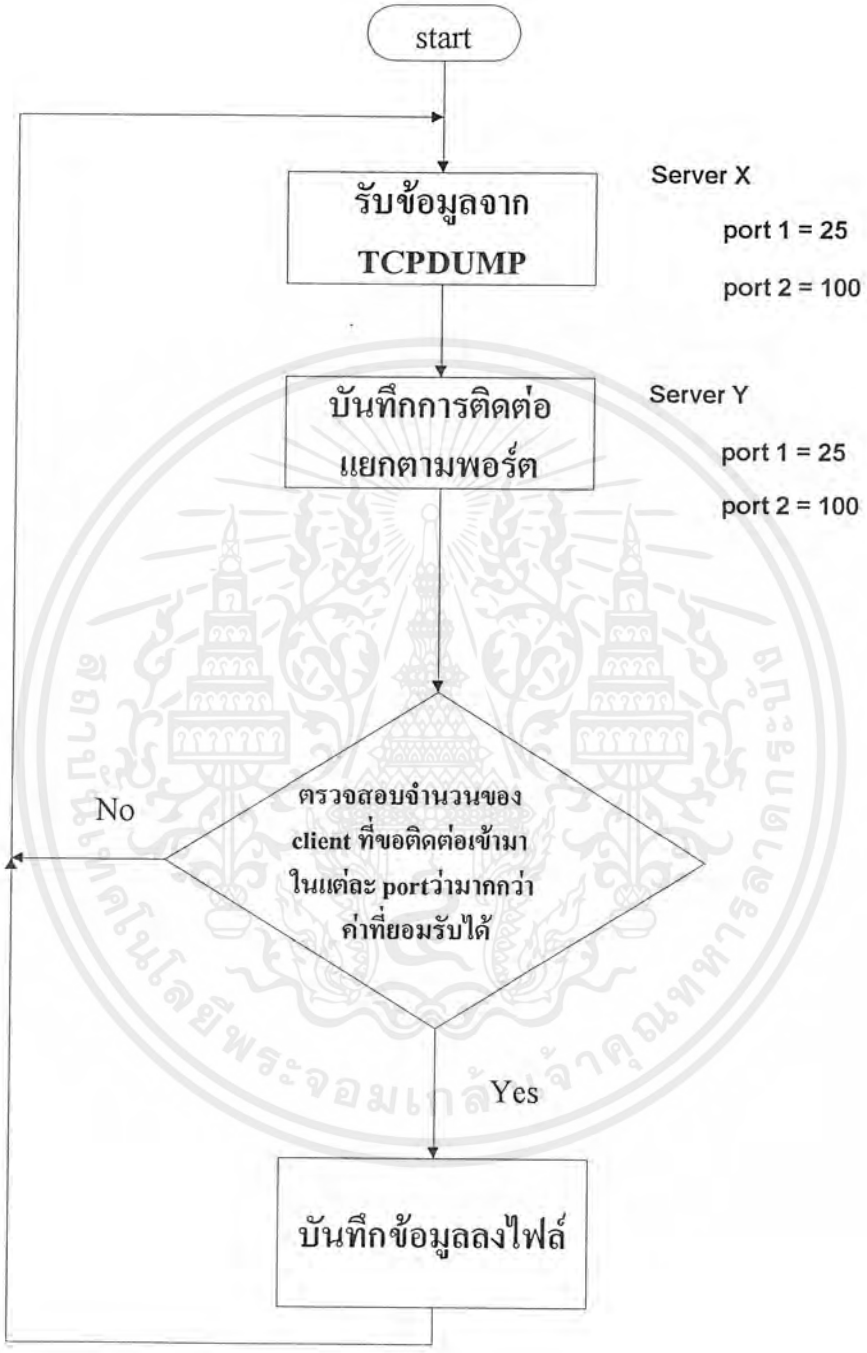
รูปที่ 6-6 แสดงการกำหนด State ที่จะใช้ในโปรแกรม

การจะตัดสินใจว่าเป็นการโจมตีโดยใช้ TCP SYN Flooding จะอาศัยเงื่อนไข 2 ประการดังนี้

1. State1 > max_SYN(100) : มีการส่งแพ็กเกจที่กำหนดแฟล็กเป็น SYN มาที่บริการในบริการหนึ่งเป็นจำนวนมากกว่าค่าที่ยอมรับได้ ก็จะถือว่าบริการนั้นถูกโจมตีโดย TCP SYN Flooding ชนิด MAX SYN เช่นมีการส่งแพ็กเกจที่กำหนดแฟล็กเป็น SYN โฮสต์ 161.246.6.220 บริการเทลเน็ต (พอร์ต 23) มากกว่าค่า max_SYN ก็จะถือว่าโฮสต์ 161.246.6.220 ถูก TCP SYN Flooding โจมตีที่พอร์ต 23 บริการเทลเน็ต

2. State2 > Backlog-Queue(15) : มีการตอบรับการขอติดต่อเข้ามาที่บริการใดๆ มากกว่าค่าที่กำหนดต่อบริการนั้นคือ มีการส่งแพ็กเกจที่กำหนดแฟล็กเป็น SYN, ACK (state) เพื่อตอบรับการขอติดต่อจาก State1 มีค่ามากขึ้นจนเป็นให้คิวที่ใช้ในการดูแลกระบวนการ three-way handshake เพิ่มขึ้นมากเกินกว่าค่าที่ยอมรับได้ก็ จะถือว่าเป็นการโจมตีด้วย TCP SYN Flooding ชนิด Backlog-Queue Full ซึ่งจะทำให้คิวที่ใช้ในการดูแลกระบวนการ three-way handshake ของบริการนั้นเต็ม

การตรวจจับการโจมตีโดยใช้ TCP SYN Flooding จะถูกดูแลโดยโปรแกรม de_synf.pl ดังมีผังการทำงานและรายละเอียดดังนี้



รูปที่ 6-7 แสดงผังการทำงานของโปรแกรมตรวจจับ TCP SYN Flooding

การเก็บข้อมูลที่จะใช้ตรวจสอบจะเก็บข้อมูลตามเซิร์ฟเวอร์ โดยแยกดูตามโฮสต์ที่ทำการร้องขอการติดต่อ ทำการ (ส่งแพ็กเกจ SYN) และที่โฮสต์ทำการตอบรับ (ส่งแพ็กเกจ SYN, ACK) โฮสต์ใดเป็นเซิร์ฟเวอร์

```
server A = {
        Port P1 = {
                State 1 => num.,
                State 2 => num.
            }
        Port  P2 =  { ……..}
        .
        .
        .
        Port Pn =  {……..}
            };
        server B = {…
            };
```

- ลักษณะ

เป็นโปรแกรมที่ทำการตรวจสอบการโจมตีชนิด TCP SYN Flooding

- วิธีการตรวจจับข้อมูล

ทำการตรวจสอบการโจมตี TCP SYN Flooding ภายในเครือข่ายโดยจะนำข้อมูลจาก tcpdump มาตรวจสอบตามเงื่อนไขที่กำหนดไว้ ถ้าพบก็จะบันทึกข้อมูลลงล็อกไฟล์

- ผลของการวิเคราะห์

มีการเขียนข้อมูลลงไฟล์ attack.dat ต่อการตรวจพบ 1 ครั้ง 1 บรรทัด

Wed Feb 10 01:00:11 1999:~TCP SYN Flooding~161.246.10.220:3128:Backlog-Queue Full

Fri Feb 11 01:00:05 1999~TCP SYN Flooding~161.246.10.220:19612:MAX SYN

- การเรียกใช้งานโปรแกรม

มีการสั่ง run "tcpdump –tt –S –n 'tcp' |de_synf.pl"

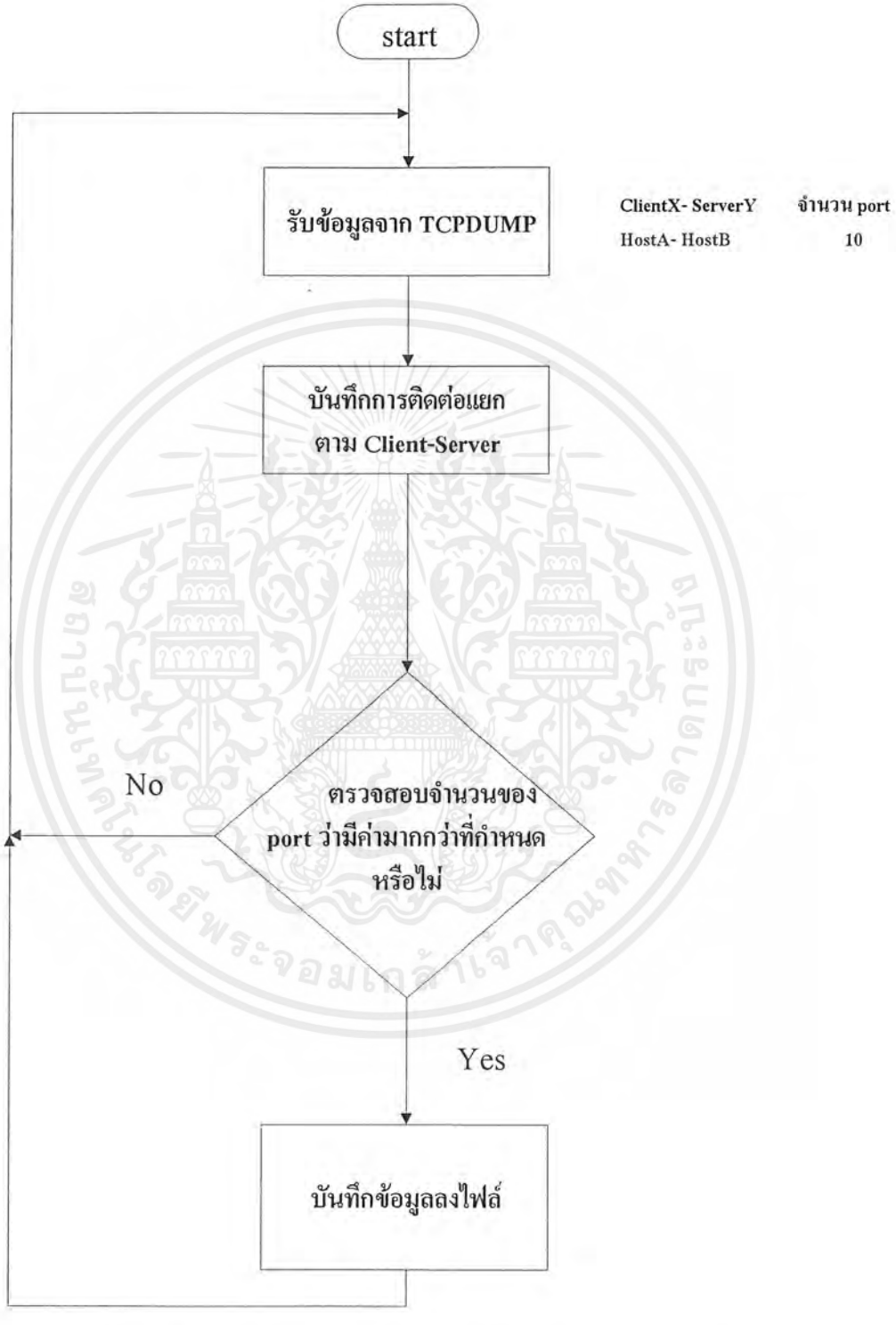
6.2.2 การตรวจจับสแกนพอร์ต

การตรวจจับหาการสแกนพอร์ตจะมีเงื่อนไขดังนี้

1.ไคลเอนต์ใดๆ ก็ตามจะที่ทำการติดต่อกับเซิร์ฟเวอร์ใดเซิร์ฟเวอร์หนึ่งจะต้องไม่มีการติดต่อไปยังพอร์ตที่แตกต่างกันมากกว่าค่าที่ยอมรับได้ (port_Max = 7) โดยเราจะเก็บข้อมูลของไคลเอนต์นั้นต่อเซิร์ฟเวอร์ไว้ ตัวอย่างเช่น ไคลเอนต์ A ทำการติดไปยัง เซิร์ฟเวอร์ B พอร์ตที่ 1 ,พอร์ตที่ 2 , ... จนมีจำนวนพอร์ตที่แตกต่างกันมากกว่า port_Max ก็จะถือว่าไคลเอนต์ A ทำการสแกนพอร์ต

2.เวลาครั้งแรกสุดที่ไคลเอนต์ติดต่อไปยังเซิร์ฟเวอร์ห่างจากครั้งหลังสุดที่ทำให้จำนวนพอร์ตแตกต่างกันมากกว่า port_max จะเป็นตัวบอกชนิดของการสแกนโดยที่ เวลาจะเป็นตัวบอกชนิดของการสแกนช่วงเวลาที่น้อยกว่า allow_Time(7 วินาที) จะถือว่าเป็นการสแกนแบบ Real Time นั่นคือเป็นการขอติดต่อเข้ามาในช่วงเวลาไล่เลี่ยกันมากเช่น ทำการติดต่อยังพอร์ต 1,2,80,25,200,20,21,7,1000 ในช่วงเวลาน้อยกว่า 7 วินาที นั่นคือ ใช้เวลาน้อยเกินไปถ้าจะเป็นการติดต่อเพื่อใช้งานตามปกติ ถ้าหากช่วงเวลามากกว่า low_Time จะถือว่าเป็นการสแกนพอร์ตแบบ Collection ซึ่งแบบนี้จริงแล้วอาจจะไม่เป็นการจงใจจะทำการสแกนแต่ก็มีการติดต่อมายัง เซิร์ฟเวอร์ในจำนวนพอร์ตที่มากกว่าความจำเป็น

การทำงานของโปรแกรมตรวจจับการสแกนพอร์ตจะถูกจัดการโดยใช้โปรแกรม de_full_scan.pl ดังผังการทำงานและรายละเอียดดังนี้



รูปที่ 6-8 แสดงผังการทำงานของโปรแกรมตรวจจับสแกนพอร์ต

-ลักษณะ

เป็นโปรแกรมตรวจสอบการสแกนพอร์ตสามารถตรวจได้ทั้ง FULL SCAN และ HALF-SCAN แล้วทำการรายงานออกทางไฟล์ attack.dat

-วิธีการตรวจจับข้อมูล

ทำการตรวจสอบการโจมตีสแกนพอร์ตภายในเครือข่ายโดยจะนำข้อมูลจาก tcpdump มาตรวจสอบตามเงื่อนไขที่กำหนดไว้ ถ้าพบก็จะบันทึกข้อมูลลงล็อกไฟล์

-ผลของการวิเคราะห์

ข้อมูล 1 บรรทัดต่อการตรวจพบ 1 ครั้ง ในไฟล์ attack.dat

Wed Feb 10 01:00:11 1999~scan port~161.246.6.220:161.246.6.219: Real Time

ขอเข้ามาติดต่อหลายพอร์ตในเวลาใกล้เคียงกัน

Fri Feb 11 01:00:05 1999~sacn Port~161.246.6.220:161.246.6.219:Collection

ขอติดต่อเข้ามาหลายพอร์ต แต่ช่วงเวลาแตกต่างกันมากเป็นแบบสะสมไว้จนมากกว่าค่าที่กำหนด

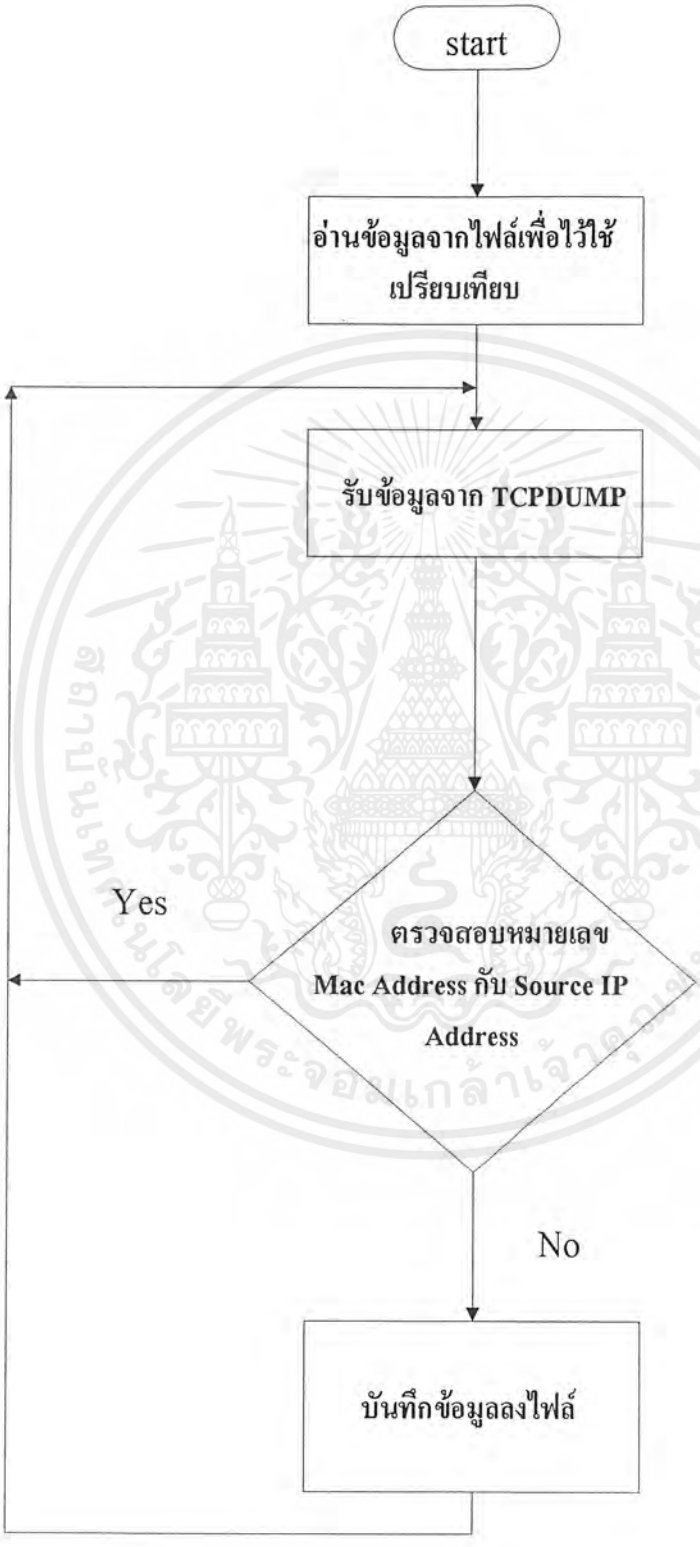
-การเรียกใช้โปรแกรม

มีการสั่งรัน "tcpdump –tt –S –n 'tcp' |de_full_scan.pl"

6.2.3 การตรวจจับการปลอมไอพี (IP-Spoofing)

การตรวจจับการปลอมไอพี จะใช้หลักการที่ว่าเน็ตเวิร์คอินเตอร์เฟสการ์ดจะมีหมายเลขเพียงหมายเลขเดียวและจะไม่ซ้ำกันภายในเครือข่ายเดียวกัน โดยเราจะกำหนดให้เน็ตเวิร์คอินเตอร์เฟสการ์ดจะต้องมีหมายเลขไอพีเพียงหมายเลขไอพีเดียวต่อหนึ่งเน็ตเวิร์คอินเตอร์เฟสการ์ดและจะไม่ซ้ำกัน ถ้าหากเน็ตเวิร์คอินเตอร์เฟสการ์ดทำการส่งแพ็กเกจออกไปด้วยหมายเลขไอพีต้นทางที่ไม่ตรงกับที่กำหนดไว้จะถือว่าเป็นการปลอมหมายเลขไอพี

การทำงานของโปรแกรมตรวจจับการ IP-Spoofing จะถูกจัดการโดยใช้โปรแกรม de_ipspoof.pl ดังมีผังการทำงานและรายละเอียดดังนี้



รูปที่ 6-9 แสดงผังการทำงานของโปรแกรมตรวจจับการปลอมไอพี

-ลักษณะ

ทำการตรวจจับการปลอมหมายเลขไอพีต้นทางไอพีต้นทางภายในเครือข่าย

-วิธีการทำงาน

ทำการรับข้อมูลจาก tcpdump –e –tt –n 'tcp' เพื่อทำการตรวจสอบแอดเดรสของเน็ตเวิร์คอินเตอร์เฟสการ์ดแอดเดรส กับ ไอพี แอดเดรส โดยใช้ข้อมูลอ้างอิงจากไฟล์ ether-ip.dat

(ตัวอย่างข้อมูลในไฟล์ 'ether-ip.dat')

161.246.6.2 0:0:f8:7e:2d:cb

โดยจะใช้แอดเดรสของเน็ตเวิร์คอินเตอร์เฟสการ์ดเป็นตัวกำหนดไอพีแอดเดรส ทำให้ 1 เน็ตเวิร์คอินเตอร์เฟสการ์ดแอดเดรสไม่สามารถมีหลายไอพีแอดเดรสได้

ถ้าแอดเดรสของเน็ตเวิร์คอินเตอร์เฟสการ์ดตรงกับข้อมูลในไฟล์ ether-ip.dat ไอพีแอดเดรสจะต้องตรงกัน ถ้าไม่ตรงกันถือว่าเป็นการปลอมไอพีแอดเดรส

-ผลของโปรแกรม

มีการเขียนข้อมูลลงไฟล์ attack.dat ถ้ามีการตรวจพบการปลอม ไอพีแอดเดรส

Wed Feb 10 01:00:11 1999~TCP IP-spoofing~161.246.6.220:32.99.85.22:0-80-48-ed-97-39

-การเรียกใช้โปรแกรม

มีการสั่งรัน "tcpdump –e –tt –n 'tcp' |de_ipspoof.pl"

6.3 การทำงานของโปรแกรมส่วน CGI

6.3.1 การทำงานของโปรแกรมที่ทำหน้าที่แสดงผลการติดต่อที่ปิดแล้ว

การทำงานในส่วนนี้จะถูกจัดการโดยโปรแกรม connection-off.cgi ดังมีรายละเอียดดังนี้

-ลักษณะ

เป็นเพริ์ลสคริปต์ที่เขียนขึ้นเพื่อรับข้อมูลจากฟอร์มคำถามที่ทำการสอบถาม โดยจะทำการสร้างไดนามิกเว็บเพจเพื่อรายงานผลการสอบถามนั้น การสอบถามจะเป็นการถามถึงการการติดต่อที่ปิดแล้ว

-วิธีการทำงาน

รับข้อมูลจากฟอร์มแล้วนำมาเปรียบเทียบกับข้อมูลในไฟล์ closed.dat ในกรณีที่เป็นวันที่ปัจจุบัน และเปรียบเทียบกับไฟล์ closedxx-xx-xxxx (วัน-เดือน-ปี) ถ้าไม่ใช่วันปัจจุบันถ้าหากไม่มีข้อมูลของวันนั้นก็จะแสดงข้อผิดพลาดบอกทางเว็บ แต่ถ้ามีก็จะทำการรายงานผลของการสอบถามออกทางเว็บ

ข้อมูลที่เป็นอินพุต ประกอบด้วย

-ชื่อ โฮสต์ หรือ ไอพีแอดเดรส

-ชื่อชนิดบริการต่างๆ หรือ หมายเลขพอร์ต

-วันที่ (day) ของนั้นๆ ขอวันที่ต้องการทราบข้อมูล

-เดือน (Mon.) 0-11

-ปี ค.ศ

-ผลของโปรแกรม

เพจที่แสดงทาง เว็บเมื่อมีการกรอกข้อมูลในฟอร์มถูกต้อง เพจแสดงข้อผิดพลาดเมื่อไม่พบข้อมูลหรือไม่พบไฟล์ ในกรณีที่ใช้ ชื่อบริการ เป็น 0 (All Service) จะแสดงข้อมูลของทุกบริการที่มีบนโฮสต์นั้น

-การเรียกใช้โปรแกรม

มีการส่งคำถามจากฟอร์มหรือ การส่งรันโปรแกรม "connection-off.cgi"

6.3.2 การทำงานของโปรแกรมที่ทำหน้าที่แสดงผลการติดต่อที่ยังเปิดใช้อยู่

การทำงานในส่วนนี้จะถูกจัดการโดยโปรแกรม connection-on.cgi ดังมีรายละเอียดดังนี้

-ลักษณะ

เป็นเพริ์ลสคริปต์เพื่อรับข้อมูลการกรอกข้อมูลจากฟอร์ม แล้วทำการแสดงออกทางเว็บโดยจะเป็นการแสดงเฉพาะ การติดต่อที่เปิดใช้งานอยู่เท่านั้น

-วิธีการทำงาน

เมื่อมีการส่งข้อมูลจากฟอร์มคำถาม โปรแกรมจะนำอินพุตจากฟอร์มมาเปรียบกับข้อมูลในหน่วยความจำที่เก็บการการติดต่อที่ยังเปิดใช้งานอยู่ (open แล้วแต่ยังไม่ close) ซึ่งสร้างจากไฟล์ close.dat และ open.dat

(ข้อมูลอินพุตประกอบด้วย)

1. ชื่อโฮสต์ หรือ ไอพีแอดเดรส
2. ชนิดของบริการ หรือ หมายเลขพอร์ต

ในกรณีที่ชนิดของบริการหรือหมายเลขพอร์ตเป็น 0 (All Service) จะทำการแสดงผลของทุกบริการของโฮสต์นั้น

-ผลของโปรแกรม

เพจที่แสดงทาง เว็บ

-การเรียกใช้งานโปรแกรม

มีการส่งข้อมูลจากฟอร์มที่รับคำถามหรือสั่งรันโปรแกรม "connection – on .cgi"

6.3.3 การทำงานของโปรแกรมแสดงผลข้อมูลของการตรวจจับการบุกรุก

การทำงานในส่วนนี้จะถูกจัดการโดยโปรแกรม update_att.cgi ดังมีรายละเอียดดังนี้

-ลักษณะ

เป็นโปรแกรมที่ดูแล เกี่ยวกับการแสดงผลของการตรวจจับการบุกรุก

-วิธีการทำงาน

ทำการแสดงผลการตรวจจับการบุกรุกที่ถูกบันทึกไว้ในไฟล์ attack.dat และยังทำการลิงค์ไปยังไฟล์ attack(หมายเลข).html เพื่อแสดงรายละเอียดของการบุกรุกแต่ละชนิด โดยจะใช้ข้อมูลจากไฟล์ attack.dat จำนวนรายการเชื่อมต่อจะเท่ากับจำนวนของข้อมูลที่ไม่ซ้ำกันในไฟล์ attack .dat

โปรแกรมจะทำการ (update) ข้อมูลใหม่ในทุกๆ 10 วินาทีในกรณีที่ผู้ใช้เปิดเว็บหน้านี้ทิ้งไว้ สำหรับไฟล์ attack X.html (X = หมายเลข) ที่อยู่ในไดเรกทอรี่เก็บ index.html /attack/ จะถูกลบก่อนที่จะมีการสร้างใหม่ทุกๆ ครั้งที่มีการแก้ไขข้อมูล

-ผลของโปรแกรม

เว็บแสดงผลการตรวจจับการบุกรุกที่สามรถลิงค์ไปหารายละเอียดของแต่ละอัน และทำการโหลดข้อมูลใหม่ทุกๆ 10 วินาที

ไฟล์ attack1.html – attack*n*.html ที่แสดงรายละเอียดของการบุกรุก แต่ละชนิด ซึ่งเก็บอยู่ที่ /attack/

attack1.html
attack2.html
attack3.html ตามจำนวนข้อมูลที่ไม่ซ้ำกันในไฟล์ attack.dat
.
.
.
attack*n*.html

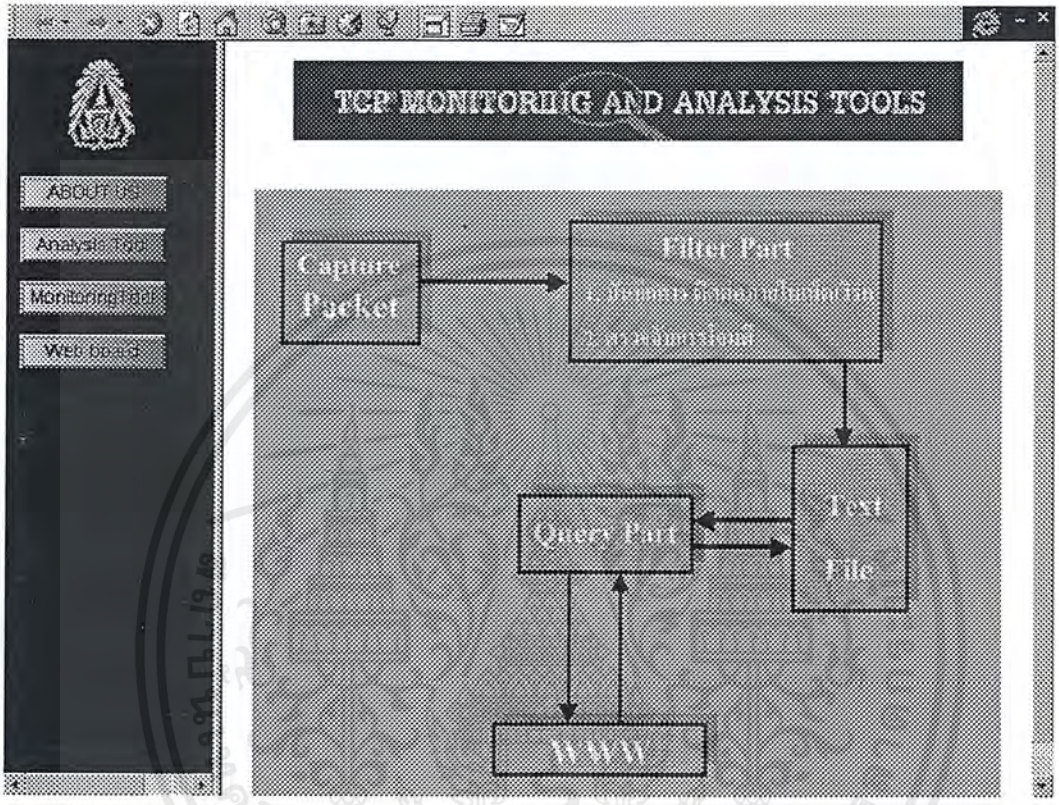
-การเรียกใช้งานโปรแกรม

มีการเปิดเว็บหน้าที่ทำการรายงานผลการตรวจจับการบุกรุกขึ้นมา

บทที่ 7

ตัวอย่างและการใช้งานโปรแกรม

การแสดงผลของปริญญานิพนธ์จะแสดงออกทางเว็บเพจ โดยมีรายละเอียดดังนี้



รูปที่ 7-1 แสดงที่เป็น Home Page ที่แสดงหน้าหลักของงาน

หัวข้อแต่ละรายการสามารถเลือกได้ดังรายละเอียดดังนี้

About us : ทำการเชื่อมต่อไปหาหน้าที่แสดงรายละเอียดอธิบายเกี่ยวกับตัวโครงงานนี้

Attack Tool : ทำการเชื่อมต่อไปหาหน้าที่แสดงผลของการตรวจจับการโจมตีที่ตรวจพบ

Monitoring Tool : ทำการ เชื่อมต่อไปหาหน้าที่แสดงผลของการติดต่อภายในเครือข่าย

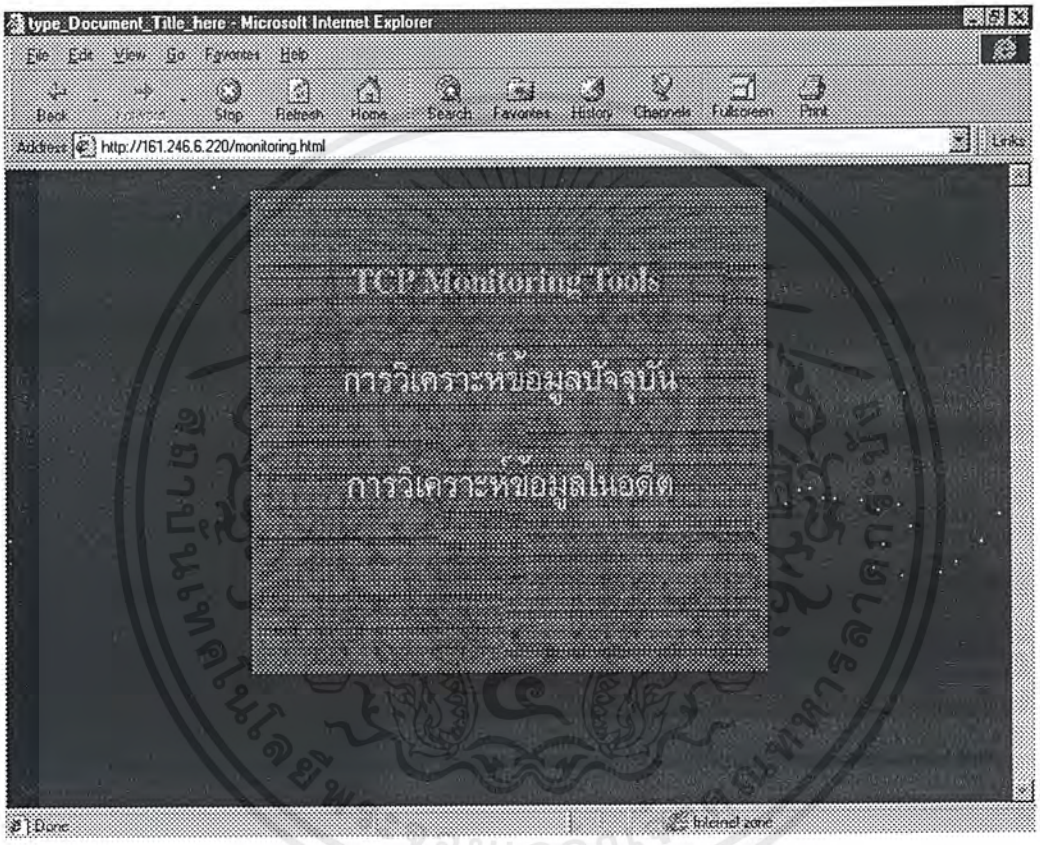
Web board : ทำการ เชื่อมต่อไปหาหน้าที่แสดงการฝากข้อความและคำแนะนำให้กับผู้ดูแล

7.1 การสอบถามข้อมูลของการติดต่อที่เกิดขึ้นในเครือข่าย

การสอบถามข้อมูลของการติดต่อที่เกิดขึ้นในเครือข่ายแบ่งออกเป็น 2 แบบ คือการติดต่อที่ปิดไปแล้ว และการที่ต่อที่ยังใช้อยู่ในปัจจุบัน ณ เวลาที่ถาม โดยการแสดงข้อมูลจะถือเอาโฮสต์ที่ถูกถามเป็นศูนย์กลางของการติดต่อที่เกิดขึ้น โดยสามารถบอกทิศทางทั้งเข้าและออก

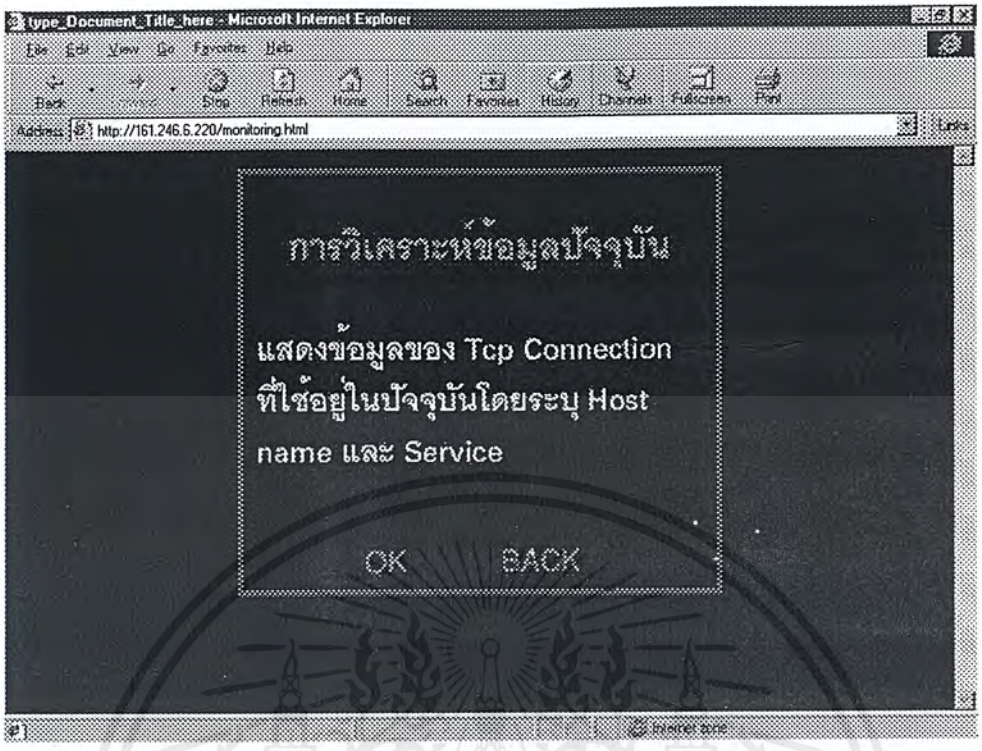
7.1.1 การสอบถามข้อมูลของการติดต่อที่เปิดอยู่

ให้เลือกหัวข้อของ Monitoring Tool เมื่อเลือกแล้วจะได้ผลดังรูปที่ 7-2

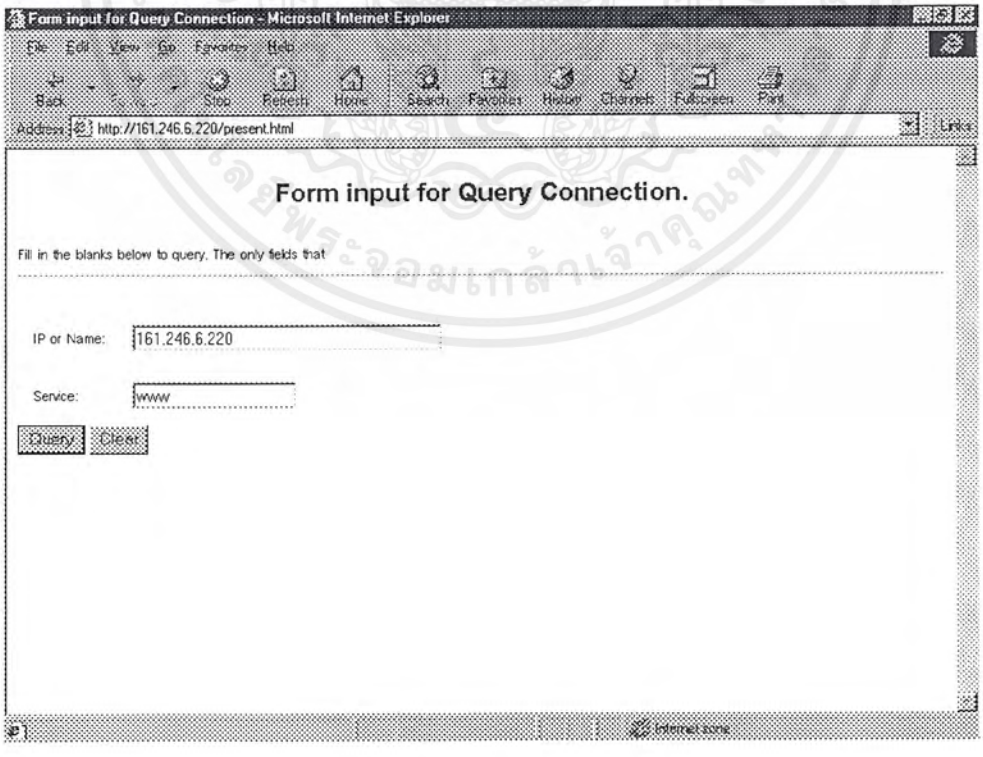


รูปที่ 7-2 แสดงการเลือกเพื่อจะดูการติดต่อภายในเครือข่าย

เลือกการวิเคราะห์ข้อมูลในปัจจุบันเมื่อเลือกแล้วจะ ปรากฏดังรูปที่ 7-3 แล้วทำการเลือก OK เพื่อระบุเครื่องและบริการที่ต้องการดูข้อมูล ดังรูปที่ 7-4

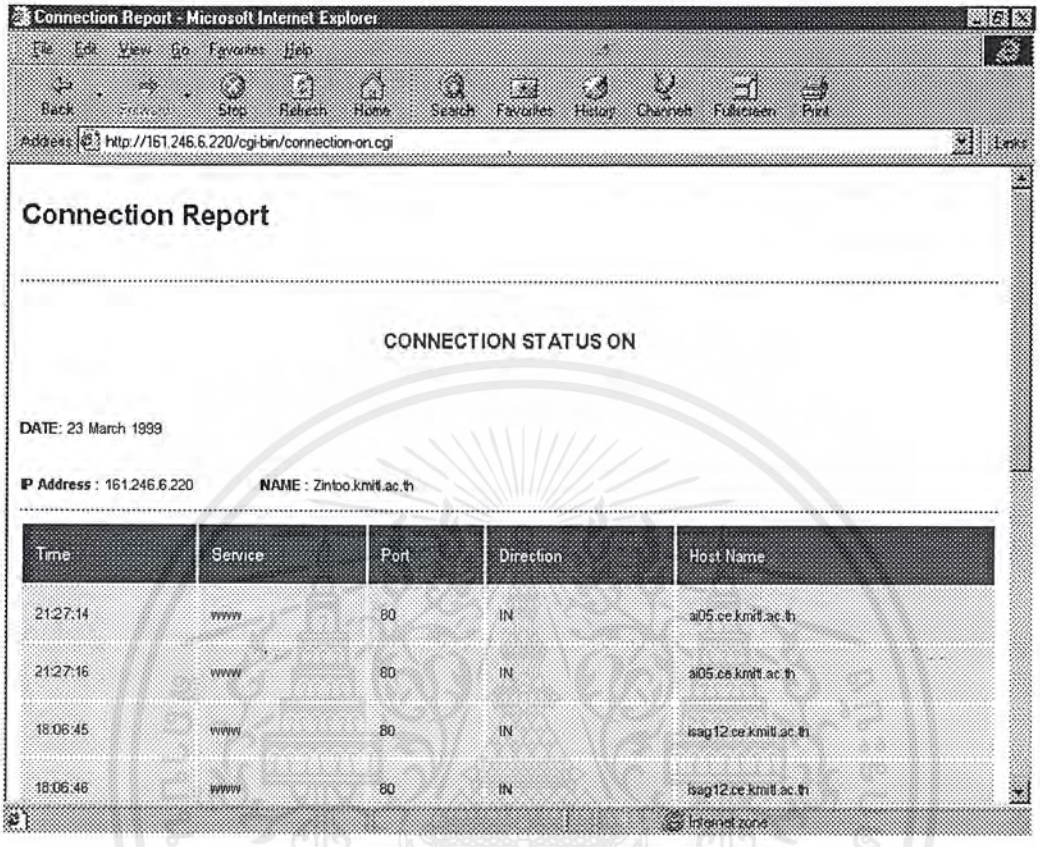


รูปที่ 7-3 แสดงผลของการเลือกดูผลของการติดต่อที่ยังเปิดอยู่ในปัจจุบัน



รูปที่ 7-4 แสดงการระบุข้อมูลเพื่อเลือกโฮสต์และระบุบริการที่ต้องการ

โดยการระบุชื่อเครื่องสามารถระบุเป็นชื่อหรือไอพีแอดเดรส เมื่อผู้ใช้ทำการส่งฟอร์ม เพื่อทำการถามข้อมูล ถ้ามีข้อมูลก็จะ ปรากฏผลดังรูปที่ 7-5

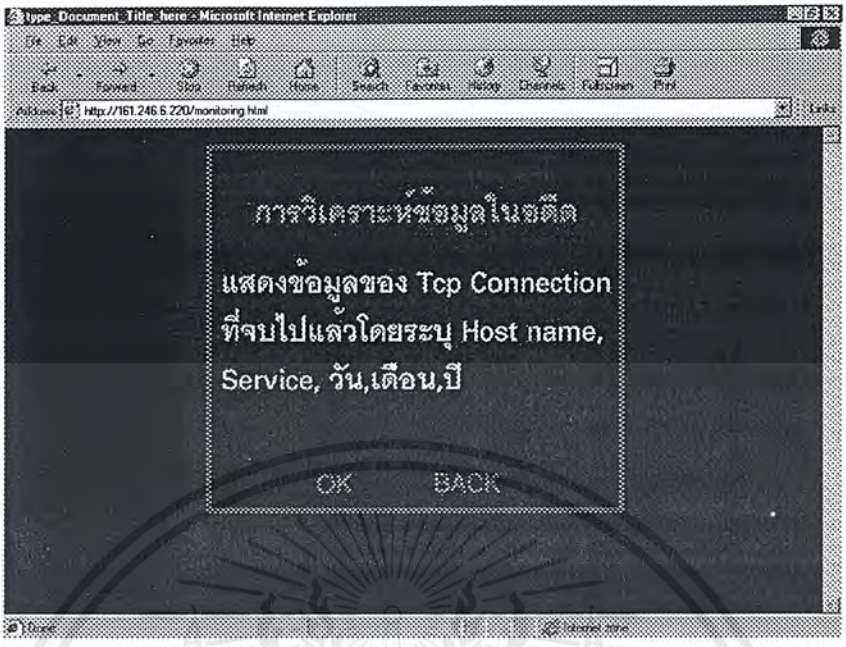


รูปที่ 7-5 แสดงของการติดต่อในปัจจุบันของโฮสต์นั้น

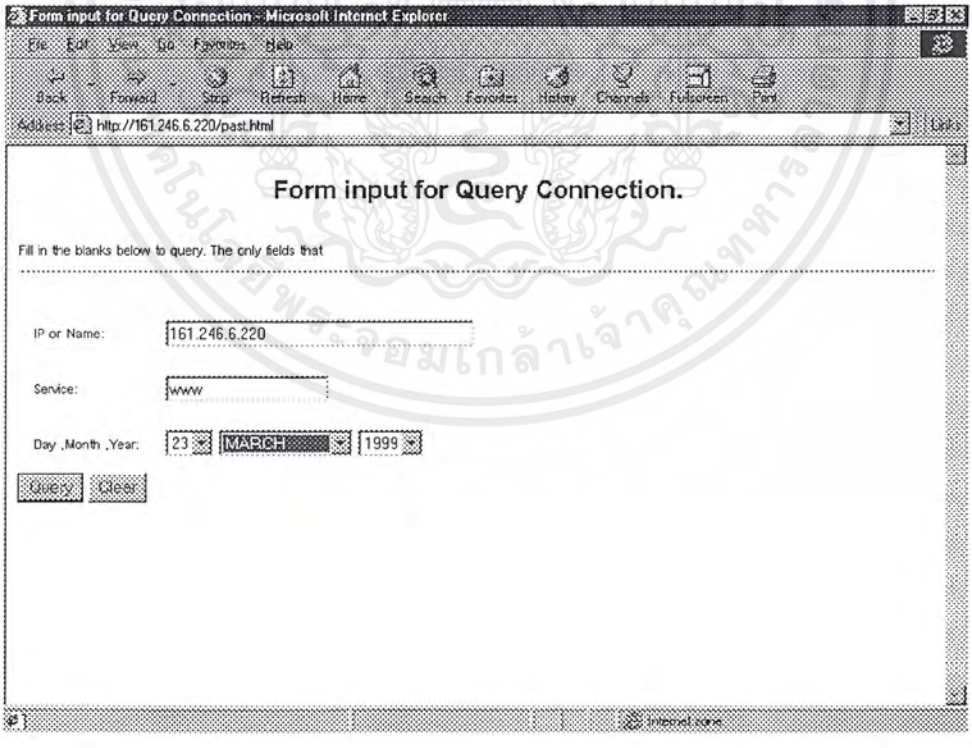
รายละเอียดและความหมายจะกล่าวในภายหลังไปแล้ว

7.1.2 การสอบถามข้อมูลของการติดต่อที่ปิดไปแล้ว

จากรูปที่ 7-2 ให้ทำการเลือกรายการ การวิเคราะห์ข้อมูลในอดีต หลังจากเลือกแล้วจะได้ผลดังรูปที่ 7-6 และทำการเลือก OK เพื่อกำหนดโฮสต์, หมายเลขพอร์ตหรือบริการและวันเดือนปีที่ต้องการดูข้อมูลดังรูปที่ 7-7

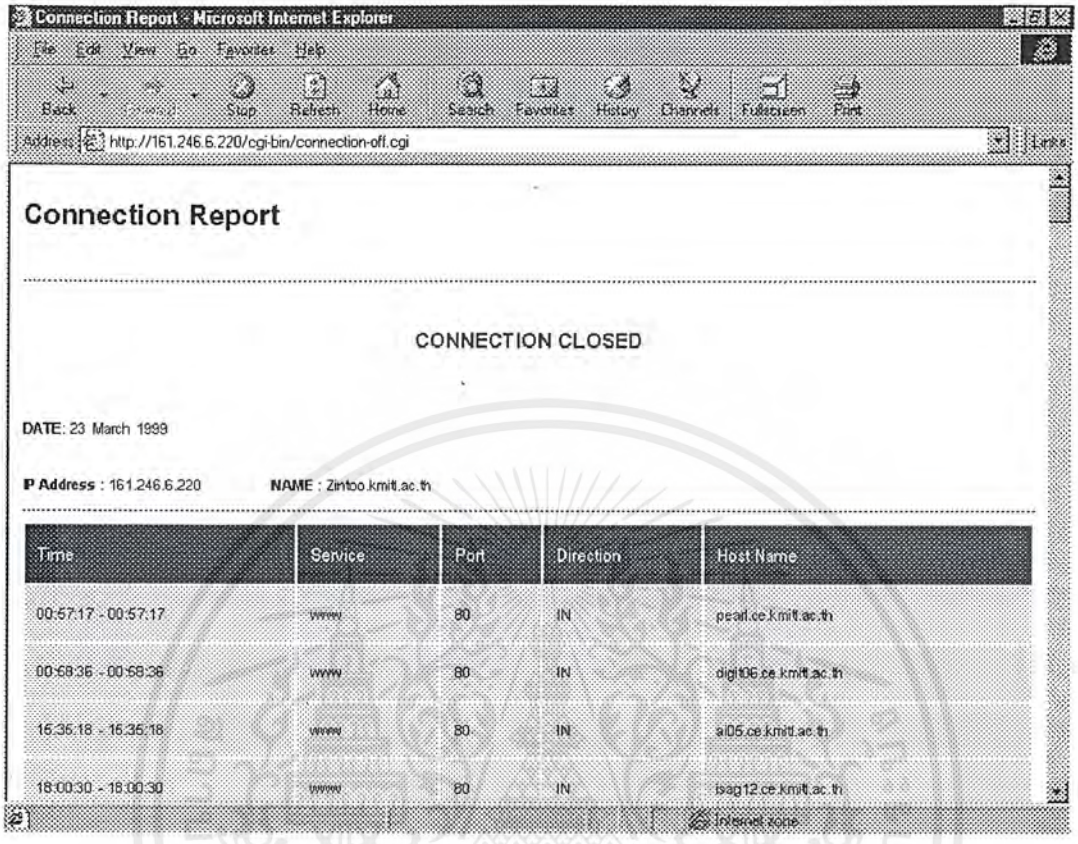


รูปที่ 7-6 แสดงการเลือกการติดต่อที่ปิดไปแล้ว



รูปที่ 7-7 แสดงการกรอกข้อมูลเพื่อทำการสอบถามการติดต่อที่ปิดไปแล้ว

ผลของการสอบถามจะออกมาดังรูปในกรณีที่มีข้อมูลของวันนั้นดังรูปที่ 7-8



รูปที่ 7-8 แสดงผลของการติดต่อที่ปิดไปแล้วจากรูปที่ 7-7

ตัวอย่างผลของการสอบถาม

Connection Report

CONNECTION CLOSED

DATE: 10 February 1999

IP Address : 161.246.6.220 NAME : Zintoo.kmitl.ac.th

Time	Service	Port	Direction	Host Name
20:36:20 – 20:36:21	www	80	IN	Ai05.ce.kmi
20:21:02 – 20:37:01	telnet	23	OUT	Chaokhun.k
20:37:43 – 20:37:43	auth	113	IN	Chaokhun.k
20:37:43 – 20:46:05	telnet	23	OUT	Chaokhun.k
20:52:15 – 20:52:15	www	80	IN	ai05.ce.kmit
20:52:15 – 20:52:15	www	80	IN	ai05.ce.kmit
20:52:15 – 20:52:16	www	80	IN	ai05.ce.kmit

รูปที่ 7-9 แสดงผลของการติดต่อที่ปิดไปแล้ว

ดังมีความหมายดังนี้

Time : แสดงเวลาที่เริ่มต้นและการติดต่อและปิดการติดต่อ

Service : แสดงบริการที่ใช้ในการติดต่อ เช่น Telnet, ftp และอื่นๆ

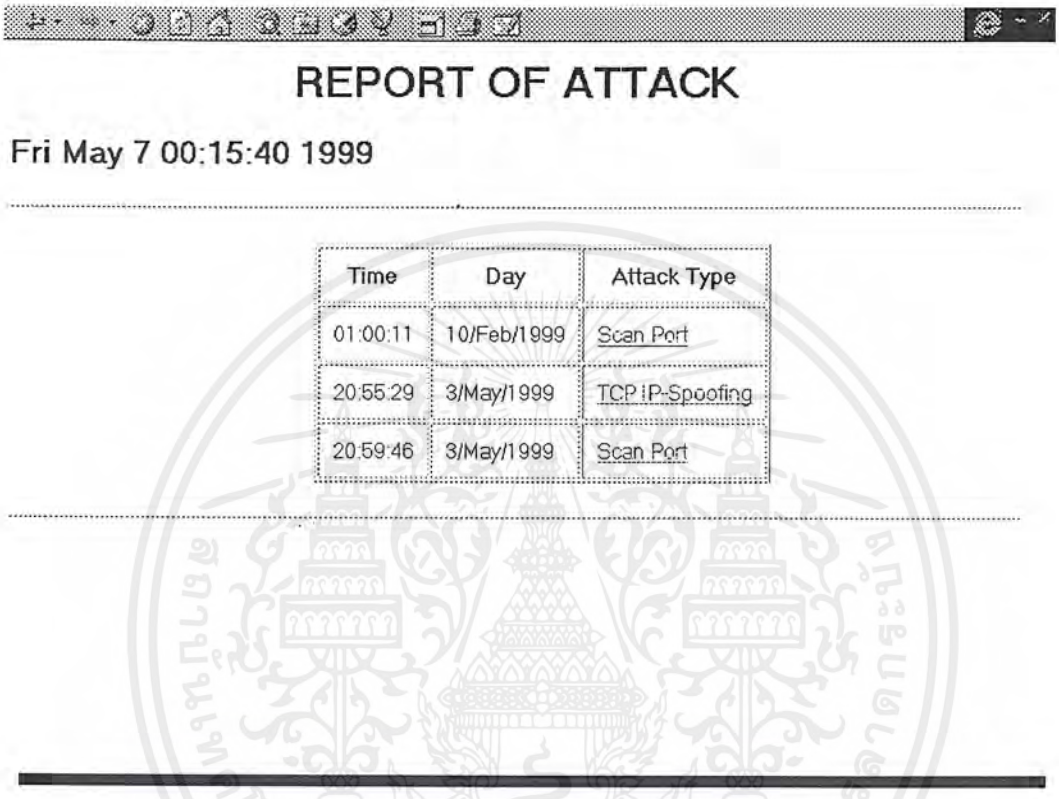
Port: แสดงหมายเลขพอร์ตที่ใช้ในการติดต่อนั้น

Direction: แสดงทิศทางว่าเป็นการขอติดต่อเข้ามา (IN) หรือออกไป (OUT) จากโฮสต์นั้น

Host Name: แสดงโฮสต์ที่ทำการติดต่อด้วย

7.2 การรายงานผลการตรวจจับการบุกรุก

จากรูปที่ 7-1 ให้ทำการเลือกรายการ Analysis Tool จะได้ผลดังรูปที่ 7-10 จากรายการนี้เราให้ทำการปรับปรุงข้อมูลอัตโนมัติจะได้ข้อมูลปัจจุบันและข้อมูลเก่ามาแสดง



รูปที่ 7-10 แสดงรายงานผลการโจมตีที่ตรวจพบซึ่งสามารถเชื่อมต่อไปหารายละเอียดของการโจมตี

แล้วสามารถเข้าไปดูรายละเอียดโดยมีการเชื่อมไว้จะได้ข้อมูลจะปรับปรุงเว็บเพจเองอัตโนมัติทุกๆ 10 วินาทีผลของการรายงานผลการตรวจจับการบุกรุกจะเหมือนดังรูปที่ 7-10 ซึ่งเป็น index ที่สามารถเชื่อมต่อไปหารายละเอียดของการบุกรุกหรือโจมตีที่ตรวจพบได้ ดังมีรายละเอียดของตามแต่ละชนิดดังนี้

7.2.1 รายละเอียดของการตรวจพบการสแกนพอร์ต

Report of Attack

Time :	00:57:34
Date :	11/Feb/1999
Type :	Scan Port
Scan from Host :	161.246.6.2
Scan to Host :	161.246.6.225
Scan Port Type :	Real Time

รูปที่ 7-11 แสดงรายละเอียดของการตรวจพบการ Scan Port

มีความหมายดังนี้

Time: เวลาที่ทำการตรวจพบ

Date: วันที่ทำการตรวจพบ

Type: ชนิดของการบุกรุกที่ตรวจพบ (TCP SYN Flooding, Scan Port หรือ IP-Spoofing)

Scan from Host: โฮสต์ที่ทำการสแกนพอร์ต

Scan to Host: โฮสต์ที่ถูก สแกนพอร์ต

Scan Port Type: Type ประกอบด้วย 2 ชนิด คือ Real Time กับ Collection แบบ Real Time จะเป็นการสแกนมาในช่วงเวลาไล่เลี่ยกัน ส่วนแบบ Collection จะเป็นการสะสมโดยอาจจะใช้เวลานานเท่าใดก็ได้

7.2.2 รายละเอียดของการตรวจพบ TCP SYN Flooding

Report of Attack

Time :	21:45:49
Date :	10/Feb/1999
Type :	TCP SYN Flooding
Flooding to Host :	161.246.6.1
Flooding to Port :	524
Tcp Syn Flooding Type :	Backlog-Queue Full

รูปที่ 7-12 แสดงรายละเอียดของการตรวจพบการ TCP SYN Flooding

มีความหมายดังนี้

Time: เวลาที่ทำการตรวจพบ

Date: วันที่ทำการตรวจพบ

Type: ชนิดของการบุกรุกที่ตรวจพบ (TCP SYN Flooding, Scan Port หรือ IP Spoofing)

Flooding to Host: โฮสต์ที่ถูกโจมตีโดยใช้ TCP SYN Flooding

Flooding to Port: หมายเลขพอร์ตที่ถูกโจมตีโดยใช้ TCP SYN Flooding

Tcp Syn Flooding Type: Type ประกอบด้วย 2 ชนิด คือ MAX SYN กับ Backlog-Queue Full

7.2.3 รายละเอียดของการตรวจพบ IP-Spoofing

Report of Attack

Time :	11:08:19
Date :	11/Feb/1999
Type :	TCP IP-Spoofing
Spoof from IP :	161.246.6.228
Spoof to IP :	161.246.6.224
Mac Address :	0-80-48-ed-97-39

รูปที่ 7-13 แสดงรายละเอียดของการตรวจพบการ IP-Spoofing

มีความหมายดังนี้

Time: เวลาที่ทำการตรวจพบ

Date: วันที่ทำการตรวจพบ

Type: ชนิดของการบุกรุกที่ตรวจพบ (TCP SYN Flooding, Scan Port หรือ IP Spoofing)

Spoof from IP: หมายเลขไอพีที่แท้จริงของเครื่องโฮสต์

Spoof to IP: หมายเลขไอพีที่ถูกปลอมขึ้นมา

Mac Address: หมายเลข Mac แอดเดรส ของเน็ตเวิร์คอินเตอร์เฟสการ์ดที่ทำการปลอมไอพี

บทที่ 8
สรุปและวิจารณ์

ในการทำงานของปริญญานิพนธ์นี้ สามารถที่จะสร้างซอฟต์แวร์ที่สามารถดูแลเครือข่ายที่สามารถบันทึกการติดต่อที่เกิดขึ้นภายในเครือข่ายและตรวจจับการบุกรุกโดยใช้ TCP SYN Flooding การสแกนพอร์ตและการปลอมไอพี (IP-Spoofing) อีกทั้งสามารถแสดงผลออกทางเว็บ ซึ่งสะดวกต่อการดูผลเพราะสามารถดูได้จากเครื่องทั่วๆ ไปที่มีเว็บได้บราวเซอร์

การตรวจจับการบุกรุกหรือการกระทำต่างๆ จำเป็นต้องแยกกันออกไปแต่ละอย่างเป็นอิสระต่อกัน ทำให้สามารถเพิ่มการตรวจการบุกรุกชนิดใหม่เข้าไปได้ง่าย ซึ่งสะดวกนำไปพัฒนาต่อ

ในการดูแลระบบเครือข่ายจะมีการเปลี่ยนแปลงตลอดเวลาการที่จะดูแลระบบให้ปลอดภัยและสามารถใช้งานได้อย่างมีประสิทธิภาพนั้น ผู้ดูแลจะต้องทราบสิ่งใหม่ๆ ที่เกี่ยวข้องตลอดเวลา ทั้งทางด้านเทคโนโลยีของเครือข่าย กระบวนการบุกรุกที่มีการค้นพบและคิดขึ้นใหม่ ดั้งนั้นการตรวจสอบและการดูแลก็จะต้องเปลี่ยนแปลงไปเรื่อยตามความเหมาะสม แต่ถึงอย่างไรหลักการพื้นฐานก็ไม่แตกต่างกันมากการที่เราเข้าใจหลักการพื้นฐานของการดูแลและตรวจจึงเป็นสิ่งจำเป็น ในงานของปริญญานิพนธ์นี้ต้องใช้งานในหลายส่วนประกอบกันเช่น ระบบปฏิบัติการลินุกซ์, ทีซีพีไอพี โพรโตคอล, เพริ์ล, CGI ซึ่งผู้ที่สนใจจะพัฒนาต่อจะต้องศึกษาและทำความเข้าใจทั้งหมด

8.1 ปัญหาและอุปสรรค

- การใช้งานโปรแกรม tcpdumpมีการกำหนดคอนฟิกกูเรชั่นมากมายซึ่งจะต้องกำหนดให้เหมาะสมจึงจะได้ข้อมูลที่ต้องการ
- การใช้งานและเขียนโปรแกรมภาษาเพริ์ลขาดเครื่องมือที่จะมาช่วยในการทดสอบและแก้ไขโปรแกรมทำให้ต้องเสียเวลามาก
- การทำงานบางอย่างต้องใช้สิทธ์ของผู้ดูแลระบบเช่น การรันโปรแกรม tcpdump หากมีข้อผิดพลาดเกิดขึ้นอาจทำให้เครือข่ายมีปัญหาได้

8.2 แนวทางการวิจัยต่อ

- สามารถที่จะเพิ่มการตรวจจับการบุกรุกใหม่ๆ เข้าไปได้
- สามารถเปลี่ยนแปลงการแสดงผลทางเว็บหรืออาจใช้การแสดงผลทางอื่นเพื่อให้ข้อมูลที่ต้องการถึงผู้ดูแลระบบแน่นอน เช่น ทาง ICQ หรือทางจดหมายอิเล็กทรอนิกส์

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดีเนื่องจากได้รับ การแนะนำและสนับสนุนจากบุคคลต่างๆ ซึ่งเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ก็คือท่าน อาจารย์ธนา หงษ์สุวรรณ และอาจารย์สุมณฑา หลิมศิริวงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้คำแนะนำและช่วยเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้คณะผู้จัดทำมีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดู พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ ขอกราบขอบพระคุณมา ณ ที่นี้

และขอขอบคุณ ผู้ดูแลระบบคอมพิวเตอร์ ภาควิศวกรรมคอมพิวเตอร์และเพื่อน ๆ ที่ให้ข้อคิดเห็นและกำลังใจเสมอมา

คณะผู้จัดทำ

บรรณานุกรม

[1] William R. Chewick, Steven M. Bellovin : "Firewalls and Internet Security" : ADDISON-WESLEY PUBLISHING COMPANY Reading, Massachusetts Menlo Park, California New York 1995.

[2] Donald L.Pipkin :"Halting the Hacker": Prentice Hall PTR Upper Saddle River,New Jersey 1997.

[3] W.Ricnard Stevens:"TCP/IP IIlustrated, Volume I: ADDISON-WESLEY PUBLISHING COMPANY Reading, Massachusetts Menlo Park, California New York 1994.

[4] David Medinets:"PERL 5:Que Corporation 201 W. 103rd Street Inndianapolis, Indiana 46260 USA 1996.

[5] Black,Uyless:"Computer Networks-Protocols,Standards,and Interfaces"Prentice – Hall Inc.,Engwood Cliffs,NJ.,1987.

[6] Dah Ming Chiu and Ram Sudama:"NETWORK MONITORING EXPLAINED Design and Application":ELLIS HORWOOD 1992.

[7] Ed Taylor:"Internetworking Handbook",McGraw-HILL,Inc, 1996