

ระบบต่อต้านการโจมตีเพื่อให้บริการแบบกระจาย

DDoS Defending System



นาย ณรงค์ศักดิ์ แสงนวลศรี
นาย ตวงปราชญ์ เสียงสนั่น

เลขหมู่.....
เลขทะเบียน.....46129
วัน, เดือน, ปี 20 ส.ค. 2546

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ระบบต่อต้านการโจมตีเพื่อให้ปิดบริการแบบกระจาย

DDoS Defending System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

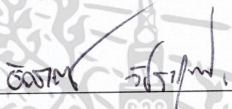
เรื่อง ระบบต่อต้านการโจมตีเพื่อให้ปิดบริการแบบกระจาย

DDoS Defending System

ผู้จัดทำ

1. นาย ณรงค์ศักดิ์ แสงนวลศรี รหัสประจำตัว 42015300
2. นาย ดวงปราชญ์ เสียงสนั่น รหัสประจำตัว 42015304


อาจารย์ที่ปรึกษา
(อาจารย์ ธนา หงษ์สุวรรณ)


อาจารย์ที่ปรึกษา
(อาจารย์ อัครเดช วิษระภูพงษ์)



ระบบต่อต้านการโจมตีเพื่อให้บริการแบบกระจาย

นาย ณรงค์ศักดิ์ แสงนวลศรี 42015300

นาย ดวงปราชญ์ เสียงสนั่น 42015304

อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา

อาจารย์ อัครเดช วัชรระภูพงษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

บทคัดย่อ

ปัญหาเรื่องความปลอดภัยของระบบคอมพิวเตอร์ผ่านเครือข่าย ที่ได้รับความสนใจมาเรื่องหนึ่ง คือ การปิดการให้บริการ (DoS: Denial of Services) ซึ่งผู้โจมตีได้มีการนำเทคโนโลยีแบบกระจาย (Distributed Technology) มาประยุกต์ใช้งาน ทำให้เกิดเป็นแนวทางการโจมตีใหม่ขึ้นมา เรียกว่า การปิดการให้บริการแบบกระจาย (DDoS: Distributed Denial of Services) มาช่วยเพิ่มความสามารถทั้งในด้านของการโจมตีและด้านการติดต่อสื่อสาร ทำให้ปัญหาที่มีอยู่ได้ทวีความรุนแรงมากยิ่งขึ้น และยากต่อการแก้ไข จึงเป็นที่มาของ ระบบต่อต้านการโจมตีเพื่อปฏิเสธการให้บริการแบบกระจาย (DDoS Defending System) เน้นที่การป้องกันเพื่อให้ได้ระบบต้นแบบที่สามารถรองรับการโจมตี และตอบโต้กลับได้อย่างเหมาะสม

DDoS Defending System

Mr. Narongsak Sangnualsri

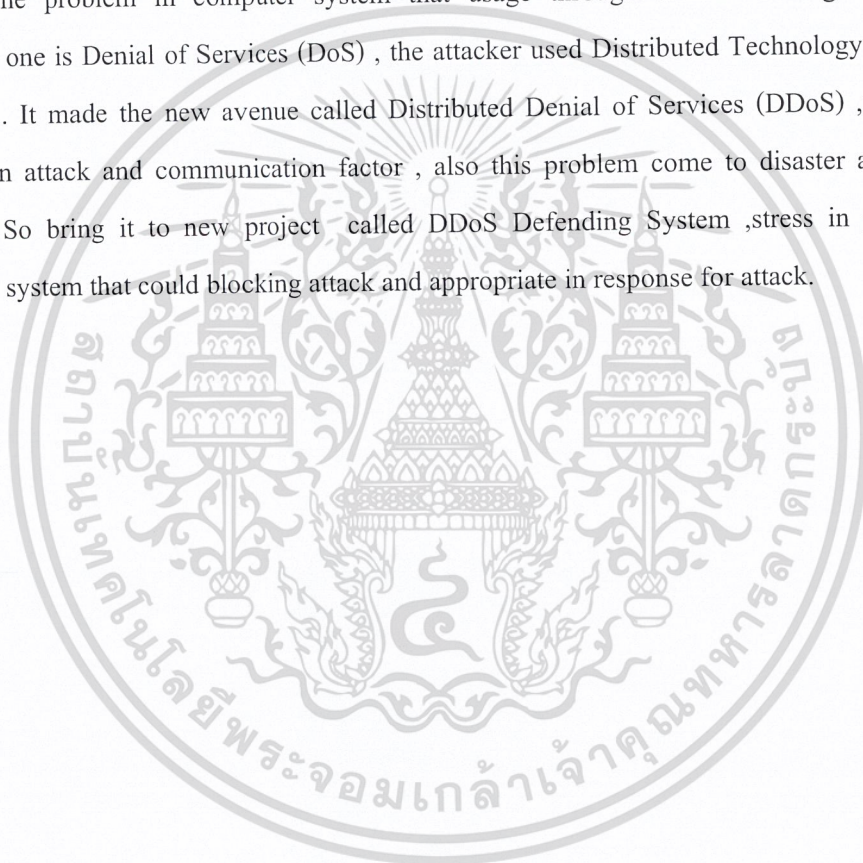
Mr. Tuangprat Saingsanun

Mr. Thana Hongsuwan Advisor

Mr. Akkradach Watcharapupong Advisor

ABSTRACT

The problem in computer system that usage through internetworking system that interested one is Denial of Services (DoS) , the attacker used Distributed Technology for applies in attack . It made the new avenue called Distributed Denial of Services (DDoS) ,increase an abilities in attack and communication factor , also this problem come to disaster and hard to defend . So bring it to new project called DDoS Defending System ,stress in prevent for prototype system that could blocking attack and appropriate in response for attack.



กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับคำแนะนำ และการให้คำปรึกษา จาก อาจารย์ ธนา หงษ์สุวรรณ และ อาจารย์ อัครเดช วัชรภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ ต้องขอขอบพระคุณเป็นอย่างสูง รวมทั้งอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้อบรมสั่งสอนวิชาความรู้แก่คณะผู้จัดทำมาโดยตลอด

และที่ต้องขอขอบพระคุณเป็นอย่างสูงที่สุดสำหรับผู้ที่มีพระคุณต่อคณะผู้จัดทำ นั่นก็คือ บิดา และมารดา ผู้ที่ได้ให้กำเนิด ให้การอบรมเลี้ยงดู และเป็นกำลังใจในทุกๆด้าน ทำให้คณะผู้จัดทำได้มีวันนี้ได้ จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

สุดท้ายนี้ก็ต้องขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้คณะผู้จัดทำมีโอกาสในการศึกษาเล่าเรียนในที่แห่งนี้

คณะผู้จัดทำ



สารบัญ

หน้าที่

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพประกอบ	XIV
สารบัญตาราง	XVI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 ขั้นตอนการดำเนิน	2
บทที่ 2 โพรโตคอลทีซีพี/ไอพี	3
2.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี	3
2.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)	3
2.3 โพรโตคอลสแต็ก	5
2.4 โพรโตคอลทีซีพี (TCP: Transmission Control Protocol)	6
2.5 โพรโตคอลยูดีพี (UDP: User Datagram Protocol)	8
2.6 โพรโตคอลไอพี (IP: Internet Protocol)	9
2.7 โพรโตคอลเออาร์พี (ARP: Address Resolution Protocol)	12
2.8 โพรโตคอลไอซีเอ็มพี (ICMP : Internet Control Message Protocol)	13
2.9 การโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี	14
2.9.1 ความหมายของการโจมตีเพื่อให้ปิดบริการ	14
2.9.2 ประเภทของการโจมตีเพื่อให้ปิดบริการ	14
2.9.2.1 ประเภทอยู่ในชั้นทรานสปอร์ต หรือชั้นอินเทอร์เน็ต	14
2.9.2.1.1 การส่งแพ็กเก็ตจำนวนมาก (Amount of Packets Sending)	14
2.9.2.1.2 ความผิดปกติของแฟร็กเมนต์ (Abnormal Fragmentation)	17
2.9.2.1.3 แบบผสม (Hybrid)	19
2.9.2.2 ประเภทอยู่ในชั้นแอปพลิเคชัน	20
2.9.2.3 โปรแกรมที่ใช้ในการโจมตีเพื่อให้ปิดบริการ	20

สารบัญ (ต่อ)

หน้าที่

บทที่ 3	การปิดการให้บริการแบบกระจาย (Distributed Denial of Service : DDoS)	23
3.1	พื้นฐานของการโจมตีในรูปแบบการปิดการให้บริการ	23
3.2	พื้นฐานของการโจมตีในรูปแบบการปิดการให้บริการแบบกระจาย	24
3.3	โหมดในการโจมตี	25
3.3.1	การใช้งานทรัพยากรที่มีอยู่อย่างจำกัด	25
3.3.1.1	การเชื่อมต่อทางเครือข่าย	25
3.3.1.2	การใช้งานทรัพยากรของคุณในการต่อต้านตนเอง	26
3.3.1.3	การใช้งานช่องว่างการสื่อสารมาก	26
3.3.1.4	การใช้งานทรัพยากรอื่นๆ	26
3.3.2	การทำลาย หรือสลับข้อมูลที่ใช้ในการกำหนดคุณลักษณะ	26
3.3.3	การทำลาย หรือสลับของส่วนประกอบของเครือข่ายทางฟิสิกส์คอล	27
3.4	ปัญหา,แนวทางป้องกัน เพื่อช่วยลดผลกระทบจากการโจมตี	27
3.4.1	ปัญหาการ Spoofing	27
3.4.2	ปัญหาการส่งต่อแพ็กเก็ตแบบบรอดคาสต์ (Broadcast)	28
3.4.3	ปัญหาการลดประสิทธิภาพของระบบเมื่อถูกโจมตี	28
3.4.4	ปัญหาความไม่ปลอดภัยของเครื่อง	28
3.5	โปรแกรมที่ใช้โจมตีเพื่อการปิดการให้บริการแบบกระจาย	29
3.5.1	Trinoo (Wintrino)	29
3.5.1.1	แนะนำ	29
3.5.1.2	เครือข่าย	30
3.5.1.3	การติดต่อสื่อสารผ่านพอร์ต	31
3.5.1.4	การป้องกันด้วยพาสเวิร์ด	31
3.5.1.5	คำสั่ง	31
3.5.1.5.1	คำสั่งเครื่อง Trinoo master	31
3.5.1.5.2	คำสั่งเครื่อง Trinoo เดมอน	32
3.5.1.6	ลักษณะการโจมตี	32
3.5.1.7	การป้องกัน	33
3.5.1.8	Trinoo Algorithm	34

สารบัญ (ต่อ)

หน้าที่

3.5.2 TFN (TFN2K, TFN3K)	38
3.5.2.1 แนะนำ	38
3.5.2.2 เครือข่าย	38
3.5.2.3 การติดต่อสื่อสาร	38
3.5.2.4 การป้องกันพาสเวิร์ด	38
3.5.2.5 คำสั่ง	38
3.5.2.6 ลักษณะการโจมตี	39
3.5.2.7 การป้องกัน	39
3.5.2.8 TFN Algorithm	41
3.5.3 Mstream	48
3.5.3.1 แนะนำ	48
3.5.3.2 เครือข่าย	48
3.5.3.3 การติดต่อสื่อสาร	48
3.5.3.4 คำสั่งของ Mstream	48
3.5.3.5 ลักษณะการโจมตี	49
3.5.3.6 การป้องกัน	49
3.5.3.7 Mstream Algorithm	49
3.5.4 Stacheldraht	54
3.5.4.1 แนะนำ	54
3.5.4.2 เครือข่าย	54
3.5.4.3 การติดต่อสื่อสาร	54
3.5.4.4 การป้องกันพาสเวิร์ด	54
3.5.4.5 คำสั่ง	55
3.5.4.6 ลักษณะการโจมตี	56
3.5.4.7 การป้องกัน	56
3.5.4.8 Stacheldraht Algorithm	56

สารบัญ (ต่อ)

หน้าที่

บทที่ 4	เครือข่ายต้นแบบเพื่อต่อต้านการโจมตีด้วย DDoS	64
4.1	โมเดล 3 – Tier Architecture of DDoS Defending System	64
4.2	คอมโพเนนต์ของ N – Tier Architecture of DDoS Defending System	66
4.3	เราเตอร์ (Router)	66
4.3.1	แนะนำ	66
4.3.2	Routing Algorithm	67
4.3.3	ประเภทของ Routing Algorithm	68
4.3.3.1	Non adaptive algorithm	68
4.3.3.1.1	Shortest Path Routing	68
4.3.3.1.2	Flooding	68
4.3.3.1.3	Flow-Based Routing	68
4.3.3.2	Adaptive algorithm	69
4.3.3.2.1	Distance Vector Routing (Bellman-Ford)	69
4.3.3.2.2	Link State Routing	69
4.3.3.2.3	Hierachical Routing	69
4.3.3.2.4	Broadcast Routing	69
4.3.3.2.5	Multicast Routing	70
4.4	ไฟร์วอลล์ (Firewall)	70
4.4.1	แนะนำ	70
4.4.2	เทคนิคที่ใช้งาน	70
4.4.2.1	Direction Control	70
4.4.2.2	User Control	71
4.4.2.3	Behavior Control	71
4.4.3	ความสามารถของไฟร์วอลล์	71
4.4.4	ข้อจำกัดของไฟร์วอลล์	71
4.4.5	ชนิดของไฟร์วอลล์	71
4.4.5.1	Packet Filter Router	71
4.4.5.2	Application Level Gateway	72

สารบัญ (ต่อ)

หน้าที่

4.4.5.3 Circuit Level Gateway	73
4.4.5.4 Stateful Inspection (Dynamic packet filtering)	73
4.4.5.5 Cache Server	74
4.4.6 ชนิดของพร็อกซีเซิร์ฟเวอร์ (Proxy Servers)	75
4.4.6.1 Generic Firewall Proxies	75
4.4.6.2 Departmental Proxy Server	76
4.4.6.3 Personal Proxy Server	76
4.4.6.4 Specialized Proxy Server	76
4.4.6.4.1 Specialized Proxy Server ระหว่างพร็อกซี (Proxies) กับไคลเอนต์ (Client)	76
4.4.6.4.2 ช่วยเพิ่มประสิทธิภาพในการติดต่อ (Accelerator)	76
4.4.6.5 Reverse Proxy Server	76
4.4.6.6 การออกแบบระบบไฟร์วอลล์ (Firewall Configuration)	77
4.4.6.6.1 Screened Host Firewall (Single Homed Bastion Host)	77
4.4.6.6.2 Screened Host Firewall (Dual Homed Bastion Host)	78
4.4.6.6.3 Screened-subnet Firewall	78
4.4.7 Basic border Firewall	79
4.4.8 เครือข่าย DMZ	79
4.4.9 ดิวอัลไฟร์วอลล์ (Dual Firewall)	80
4.5 ระบบต่อต้านการโจมตี (Intrusion Detection System) และ Network IDS	80
4.5.1 แนะนำ	80
4.5.2 วิธีการตัดสินใจของลักษณะการบุกรุก	81
4.5.2.1 ใช้สถิติในการวิเคราะห์หาความผิดปกติ (Statistical Anomaly Detection)	81
4.5.2.2 การตรวจสอบจากรูปแบบ (Pattern Matching Detection)	81
4.5.3 Realtime หรือ Internet Based	82
4.5.3.1 Event Generator	83
4.5.3.2 Activity Profile	83
4.5.3.3 Rule Set	83
4.5.4 ประเภทของการโจมตี (Class of Attacks)	83

สารบัญ (ต่อ)

หน้าที่

4.5.4.1 ผู้ใช้งานภายใน (Internal Users)	84
4.5.4.1.1 การโจมตีการปิดการให้บริการภายในเครือข่าย (Internal Denial of Service Attack)	84
4.5.4.1.2 การเพิ่มระดับการใช้งานภายใน (Internal Privilege Escalation)	84
4.5.4.1.3 การเพิ่มระดับเป็นผู้ดูแลระบบภายใน (Internal Superuser Privileges)	84
4.5.4.2 ผู้ใช้งานภายนอก (External Users)	84
4.5.4.2.1 การทำความเสียหายการปิดการให้บริการจากภายนอก (External Denial Of Service Threats)	84
4.5.4.2.2 การเพิ่มระดับการใช้งานของผู้ใช้งานภายนอก (External Privilege Escalations)	84
4.5.5 ระดับของแหล่งข้อมูลข่าวสาร (Layer of Information Sources)	85
4.5.5.1 การบริการงานเครือข่าย (Network Services)	85
4.5.5.2 ระบบปฏิบัติการ (Operating System)	85
4.5.5.3 โปรแกรม (Application)	85
4.5.6 IDS ในเชิงพาณิชย์	85
4.5.7 การโจมตีอย่างง่าย หรือ ซับซ้อน (Simple or Complex Attacks)	86
4.5.7.1 ความยากของการตรวจสอบผู้บุกรุก	86
4.5.7.2 Vulnerability Scanners	86
4.5.7.3 ในระดับของ system ที่ต้องตรวจสอบ	86
4.5.7.4 การดักฟังของผู้โจมตี (Sniffing For Intruders)	87
4.5.8 NetRanger พัฒนาโดย WheelGroup และ Cisco	87
4.5.9 ข้อดีของ Network IDS	87
4.5.10 ข้อจำกัดของการดักจับแพ็กเก็ตในเครือข่าย	87
4.5.10.1 ไม่ทราบผลกระทบที่จะเกิดขึ้น (unknown side effects)	88
4.5.10.2 กรณีที่การโจมตีไม่ได้ใช้ในการรับ/ส่ง แพ็กเก็ตในเครือข่าย	88
4.5.11 ความท้าทายของปัญหาของ IDS	89
4.5.12 ปัญหาและการแก้ไข	89
4.5.12.1 เตรียมการป้องกันก่อนโจมตี (Preparation before you're hit)	89
4.5.12.2 ตรวจสอบ (Detection) หรือ ทำให้ค้นสภาพ (discovery) เมื่อเหตุการณ์เกิดขึ้น	90
4.5.12.3 ตอบสนองต่อเหตุการณ์ (Response to the incident)	90
4.5.13 ชนิดของระบบต่อต้านการโจมตี (Intrusion Detection System : IDS)	90

สารบัญ (ต่อ)

หน้าที่

4.6 SSL Architecture	91
4.6.1 แนะนำ	91
4.6.2 แนวคิดหลัก 2 ข้อ	91
4.6.2.1 SSL session	91
4.6.2.2 SSL connection	91
4.6.3 โพรโทคอล Handshake มีพารามิเตอร์ดังนี้	91
4.6.3.1 Session State กำหนดพารามิเตอร์ คือ	91
4.6.3.2 Connection State มีพารามิเตอร์	92
4.6.4 SSL Record Protocol	92
4.6.4.1 Confidentiality	92
4.6.4.2 Message Integrity	92
4.6.5 อธิบายการทำงาน	93
4.6.6 Change Cipher Spec Protocol	95
4.6.7 Alert Protocol	95
4.6.8 Handshake Protocol	96
4.7 คลัสเตอร์ (Cluster)	100
4.7.1 เครื่องคอมพิวเตอร์ประสิทธิภาพสูง (High Performance Computing)	100
4.7.2 องค์ประกอบสำหรับการทำคลัสเตอร์ (Cluster)	101
4.7.3 โครงสร้างของคลัสเตอร์	102
4.7.3.1 Share-Nothing และ Shared-Storage	102
4.7.3.2 Active/Standby และ Active/Active	103
4.7.3.3 การเชื่อมต่อ (Interconnects)	104
4.8 โหลดบาลานซ์ (Load Balance)	104
4.8.1 แนะนำ	104
4.8.2 วิธีการ (Methods)	105
4.8.3 ป้ายหรือตัวแปรที่เกี่ยวข้อง	105
4.8.3.1 Network Address Translation	105
4.8.3.2 Domain Names	106

สารบัญ (ต่อ)

หน้าที่

4.8.3.3 Wire-speed Processing	106
4.8.3.4 Node Operating System Limitations	106
4.8.3.5 Balancing Device Limitations	106
4.8.3.6 Session and nonsession-based Traffic	107
4.8.3.7 Application Dependencies	107
4.8.4 วิธีการ Balancing อย่างง่าย	107
4.8.4.1 Weighting	108
4.8.4.2 Randomization	108
4.8.4.3 Round-Robin	108
4.8.4.4 Hashing	108
4.8.4.5 Least Connections	108
4.8.4.6 Minimum Misses	108
4.8.4.7 Fastest Response	108
4.8.5 วิธีการ Advanced Balancing	109
4.8.5.1 Network Traffic-based Balancing	110
4.8.5.2 Node Traffic-based Balancing	110
4.8.5.3 Node Load-based Balancing	110
4.8.5.4 Load-balancing Domain Name Resolution	110
4.8.5.5 Topology-based Redirection	111
4.8.5.6 Policy-based Redirection	112
4.8.5.7 Bandwidth Allocation Policies	112
4.8.5.8 Administrative Policies	112
4.8.5.9 Security Policies	112
4.8.5.10 Application-specific Redirection	112
4.8.6 ความผิดพลาด (Common Errors)	113
4.8.6.1 Overflow	113
4.8.6.2 Underflow	113
4.8.6.3 Routing Errors	113
4.8.6.4 Induced Network Errorsad distribution	114

สารบัญ (ต่อ)

หน้าที่

4.8.7 Dynamic Load Balancing	114
4.8.7.1 พิจารณาที่กระทำ คือ	114
4.8.7.2 ข้อคิดที่เกี่ยวข้อง	114
4.8.7.3 Dynamic Load Balancing (DLB)	114
4.8.7.4 กลวิธีการทำโหลดบาลานซ์ (Load Balancing Strategies)	115
4.8.7.4.1 Global Strategies	115
4.8.7.4.2 Local Strategies	115
4.8.7.5 ค่าพารามิเตอร์ของโมเดล	116
บทที่ 5 เครื่องมือที่ใช้ในการตรวจสอบโปรแกรมประเภท DDos	117
5.1 DDoSPing	117
5.1.1 ลักษณะการทำงานของโปรแกรม	117
5.1.2 รูปแบบในการใช้งานโปรแกรม	117
5.2 Zombbie Zapper	122
5.2.1 ลักษณะการทำงานของโปรแกรม	122
5.2.2 รูปแบบในการใช้งานโปรแกรม	122
5.3 Find_ddos	124
5.3.1 ลักษณะการทำงานของโปรแกรม	124
5.3.2 รูปแบบในการใช้งานโปรแกรม	124
บทที่ 6 โปรแกรมที่พัฒนาขึ้น	127
6.1 อัลกอริทึมการทำงานของโปรแกรม	127
6.2 ฟังก์ชันต่างๆของโปรแกรม	127
6.3 คุณสมบัติของโปรแกรม	128
6.4 ข้อจำกัดของโปรแกรม	129
6.5 การใช้งานโปรแกรม	129
6.6 จุดเด่นและจุดบกพร่องในการทำงานของโปรแกรม	131
6.6.1 จุดเด่น	131
6.6.2 จุดบกพร่อง	131

สารบัญ (ต่อ)

หน้าที่

บทที่ 7 วิเคราะห์ผลการทดลองและสรุปผล	132
7.1 ปัญหาและอุปสรรค	132
7.2 แนวทางการวิจัยและพัฒนาต่อ	132
7.3 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นกับเครื่องมืออื่นๆที่มีอยู่	132
บรรณานุกรม	134



สารบัญภาพ

หน้าที่

รูปที่ 2-1 แสดงการเปรียบเทียบเลขอร์ของโอเอสไอกับเลขอร์ของทีซีพี/ไอพี	3
รูปที่ 2-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี	5
รูปที่ 2-3 แสดงโพรโตคอลสแต็กของทีซีพี/ไอพี	5
รูปที่ 2-4 แสดงการทำ 3-way Handshake	6
รูปที่ 2-5 แสดงแพ็กเก็ตทีซีพี	8
รูปที่ 2-6 แสดงแพ็กเก็ตยูดีพี	9
รูปที่ 2-7 แสดงการทำแฟร็กเมนต์ชัน	9
รูปที่ 2-8 แสดงการรีแอสเซมเบิล	10
รูปที่ 2-9 แสดงแพ็กเก็ตไอพี	11
รูปที่ 2-10 แสดงเออาร์ทีดาต้าแกรม	12
รูปที่ 2-11 แสดงฟอร์แมตของ ไอซีเอ็มพี	13
รูปที่ 2-12 แสดงการโจมตีด้วย Ping Flood Attack	15
รูปที่ 2-13 แสดง รูปแบบแพ็กเก็ตที่เกิดขึ้นจากการโจมตีจาก Ping Flood Attack	15
รูปที่ 2-14 แสดงการส่งแพ็กเก็ตแบบ SYN Flood	16
รูปที่ 2-15 แสดง แพ็กเก็ตที่เกิดขึ้นจากการโจมตีแบบ SYN Flood	17
รูปที่ 2-16 แสดงการรีแอสเซมบลีแบบปกติ	17
รูปที่ 2-17 แสดงการส่งเฉพาะแพ็กเก็ตสุดท้ายไปยังเป้าหมาย	18
รูปที่ 2-18 แสดงการรีแอสเซมบลีแบบแพ็กเก็ตมีขนาดเหมือนกัน	18
รูปที่ 2-19 แสดงการโจมตีโดยส่งแพ็กเก็ตที่ไม่สามารถรีแอสเซมเบิลได้	19
รูปที่ 2-20 แสดงการโจมตีโดยส่งแพ็กเก็ตแบบวนลูป	19
รูปที่ 2-21 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอ	19
รูปที่ 3-1 แสดงพื้นฐานของการปิดการให้บริการ	23
รูปที่ 3-2 แสดงพื้นฐานของการปิดการให้บริการแบบกระจาย	24
รูปที่ 4-1 แสดง เครือข่ายต้นแบบเพื่อต่อต้านการโจมตีด้วย DDoS	66
รูปที่ 4-2 แสดง Packet-filtering router	72
รูปที่ 4-3 แสดง Application-level gateway	73
รูปที่ 4-4 แสดง Circuit-level gateway	73
รูปที่ 4-5 แสดงการทำงานของ cache server	74

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 4-6 แสดงเมื่อมีการร้องขออบเจ็กต์เดิมจากเครื่องไคลเอ็นต์	75
รูปที่ 4-7 แสดง Screened host firewal system (single-homed bastion host)	77
รูปที่ 4-8 แสดง Screened host firewall system (dual-homed bastion host)	78
รูปที่ 4-9 แสดง Screened-subnet firewal system	78
รูปที่ 4-10 แสดง Basic Border Firewall (Single Layer)	79
รูปที่ 4-11 แสดง Basic Border Firewall with DMZ Network	79
รูปที่ 4-12 แสดง คูอัลไฟร์วอลล์ (Dual Firewall)	80
รูปที่ 4-13 แสดง Generic Intrusion Detection Model	82
รูปที่ 4-14 แสดงระดับของแหล่งข้อมูลข่าวสาร	85
รูปที่ 4-15 แสดง SSL Record Protocol Operation	92
รูปที่ 4-16 แสดง SSL Record Protocol Payload	95
รูปที่ 4-17 แสดง Handshake Protocol Action	97
รูปที่ 4-18 แสดง โครงสร้างคอมพิวเตอร์แบบคลัสเตอร์ (Cluster)	101
รูปที่ 4-19 แสดง Share-nothing clusters	102
รูปที่ 4-20 แสดง Active/standby clusters	103
รูปที่ 4-21 แสดง N-Way clusters	103
รูปที่ 4-22 แสดง Cluster-interconnects	104
รูปที่ 4-23 แสดง Ping triangulation ใน topology-based redirection	111
รูปที่ 4-24 แสดง Centralized vs. distributed strategies	115
รูปที่ 5-1 แสดงรูปแบบของโปรแกรม DDoSPing	118
รูปที่ 5-2 แสดงผลจากการสแกนของโปรแกรม DDoSPing	119
รูปที่ 5-3 แสดงการเลือกทำงานแบบ Windows defaults	120
รูปที่ 5-4 แสดงการเลือกทำงานแบบ Unix defaults	121
รูปที่ 5-5 แสดงการใช้งานของโปรแกรม Zombie Zapper	122
รูปที่ 5-6 แสดงการเลือกใช้งานในแบบ Input IP file	123
รูปที่ 5-7 แสดงการทำงานของโปรแกรม find_ddos	126
รูปที่ 6-1 แสดงการใช้งานโปรแกรม IsagDD	129
รูปที่ 6-2 แสดงตัวอย่างการใช้งานโปรแกรม IsagDD	130
รูปที่ 6-6 แสดงตัวอย่างการเพิ่มเติมออพชั่น -v	130
รูปที่ 6-4 แสดงตัวอย่างการเพิ่มเติมออพชั่น -d	131

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 แสดงการทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี	4
ตารางที่ 2-2 แสดงตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ	20
ตารางที่ 4-1 แสดงชนิดของ subnet	67
ตารางที่ 4-2 แสดงประเภทของการ โจมตี	83
ตารางที่ 4-3 แสดง SSL Protocol	91
ตารางที่ 4-4 เปรียบเทียบการเข้ารหัส	94
ตารางที่ 4-5 แสดง SSL Record Format	94
ตารางที่ 4-6 แสดงประเภทของข้อความ	96
ตารางที่ 4-7 เปรียบเทียบคุณลักษณะของคอมพิวเตอร์แบบขนาน	100
ตารางที่ 4-8 เปรียบเทียบระบบคลัสเตอร์	102
ตารางที่ 4-9 แสดงตัวแปร Optimization Vectors ของวิธีการ Advanced Balancing	109
ตารางที่ 7-1 แสดงการเปรียบเทียบโปรแกรมกับเครื่องมืออื่น	133

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

การให้บริการ และการใช้งานบนระบบอินเทอร์เน็ตนั้น ได้รับความนิยม และมีความต้องการใช้งานเพิ่มมากขึ้นอย่างมากมาย ระบบปฏิบัติการที่ได้รับความนิยมนำมาใช้งาน เช่น ระบบปฏิบัติการวินโดวส์ 95/98 , ระบบปฏิบัติการวินโดวส์เอ็นที (NT workstation/ NT server) , ระบบปฏิบัติการวินโดวส์ 2000 (Professional / Server และ Advance Server) รวมไปถึงระบบปฏิบัติการยูนิกซ์ อาทิ เช่น ซันโซลาริส (Sun Solaris), โอเพ่นบีเอสดี (OpenBSD), ฟรีบีเอสดี (FreeBSD), ลินุกซ์ (Redhat, Slackware, Mandrake ฯลฯ) และระบบปฏิบัติการอื่นๆ ที่ใช้ในการเชื่อมต่อกับเข้ากับระบบอินเทอร์เน็ต

ในระบบคอมพิวเตอร์ที่มีช่องโหว่ และมีจุดอ่อนที่ทำให้แฮกเกอร์ (Hacker) , แครกเกอร์ (Cracker) และผู้ไม่ประสงค์ดีกับระบบ พยายามหาช่องทางที่ว่า เพื่อเจาะเข้าสู่ระบบ หลังจากนั้นจะติดตั้งโปรแกรมที่จะทำหน้าที่ เป็นตัว โจมตี โดยมีการรับคำสั่งจากผู้ควบคุม การโจมตี ไปยังเป้าหมาย ซึ่งตกเป็นเหยื่อที่ได้รับความเสียหาย

เว็บที่ให้บริการที่เคยได้รับผลกระทบจากการโจมตีแบบการปิดการให้บริการแบบกระจาย อาทิ เช่น Yahoo, eBay, eTrade, Amazon, Microsoft และเว็บไซต์อื่นๆ ทำให้สูญเสียผลประโยชน์จากการให้บริการอินเทอร์เน็ตไปเป็นมูลค่ามากมาย ต่างพากันหามาตรการป้องกัน และแก้ไขผลกระทบที่ได้รับ

การโจมตีเพื่อปิดบริการแบบกระจายเป็นปัญหา ที่ยังไม่มีทางแก้ไขได้สำเร็จรูป ทั้งยังทวีความรุนแรงของปัญหามากขึ้นเรื่อยๆ อีกทั้งการแยกความแตกต่างระหว่าง การปิดการให้บริการแบบกระจาย (Distributed Denial of Services) กับการใช้งานปกติ ได้ยากมาก

โครงการนี้จึงมุ่งเน้นที่จะแยกแยะความแตกต่างระหว่าง DDoS กับการใช้งานปกติ และควบคุมหรือตอบโต้ DDoS ที่เกิดขึ้น ได้อย่างเหมาะสม

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อวิเคราะห์ และหาทางแก้ไขปัญหาค้างการโจมตีเพื่อให้บริการแบบกระจาย
2. เพื่อสร้างระบบต้นแบบในการต่อต้านการโจมตีเพื่อให้บริการแบบกระจาย

1.3 ขอบเขตของปริญญานิพนธ์

โครงการนี้ จะเน้นแนวทางแก้ไขปัญหาค้างการโจมตีเพื่อให้บริการแบบกระจาย (Distributed Denial of Services : DDoS) ทั้งยังเน้นไปยังระบบเครือข่ายระบบปฏิบัติการของวินโดวส์ และระบบเครือข่ายระบบปฏิบัติการยูนิกซ์ ซึ่งเป็นที่นิยมใช้เพื่อให้บริการด้านอินเทอร์เน็ตในปัจจุบัน เพื่อให้ได้ระบบที่จะเป็นต้นแบบในการป้องกัน ควบคุม หรือตอบโต้การโจมตีเพื่อให้บริการแบบกระจายได้อย่างเหมาะสม

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษารายละเอียดเกี่ยวกับทีซีพี/ไอพีเบื้องต้น
2. ศึกษาการเขียน โปรแกรมผ่านเครือข่ายแบบกระจาย
3. ศึกษารายละเอียดและการทำงานของ การโจมตีเพื่อการปิดการให้บริการแบบกระจาย
4. วิเคราะห์และรวบรวมแนวทางแก้ไข ปัญหา
5. ออกแบบระบบต้นแบบเพื่อรองรับการโจมตีเพื่อการปิดการให้บริการแบบกระจาย
6. ออกแบบโมเดลของโปรแกรมเพื่อรองรับการโจมตีเพื่อการปิดการให้บริการแบบกระจาย
7. พัฒนาระบบต้นแบบเพื่อรองรับการโจมตีเพื่อการปิดการให้บริการแบบกระจาย



บทที่ 2

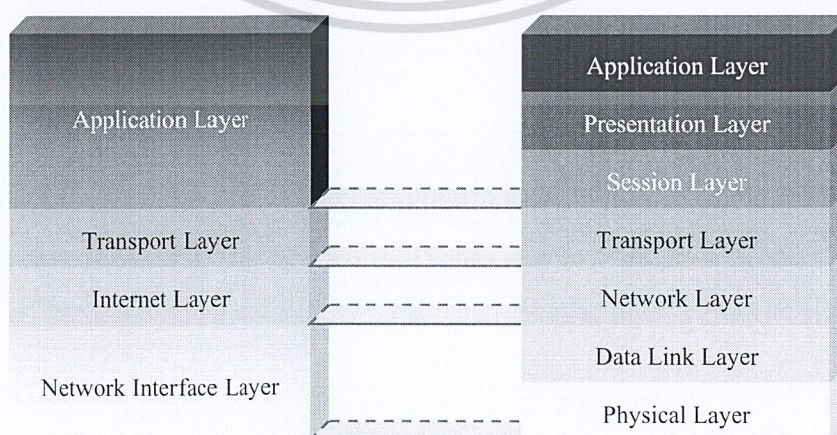
โปรโตคอลทีซีพี/ไอพี

2.1 ความเป็นมาของโปรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นมาตรฐานการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบที่มีขึ้นเมื่อกระทรวงกลาโหมสหรัฐฯ หรือ Department Of Defense (DOD) ทำการทดลองในปี ค.ศ.1969 เชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็นคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกันได้ โครงการนี้มีชื่อว่า Advanced Research Projects Agency Network หรือ ARPANET ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลของ ARPANET ประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ทีซีพี (Transmission Control Protocol หรือ TCP) และ ไอพี (Internet Protocol หรือ IP) ซึ่ง ทีซีพี มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่ง ให้ได้รับข้อมูลถูกต้องครบถ้วน ส่วน ไอพี จะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบที่แอดเดรสของผู้รับ ซึ่งเรียกว่า ไอพีแอดเดรส (IP Address) ต่อมาในปี ค.ศ.1983 ทีซีพี/ไอพี ถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมสหรัฐฯ จึงถือว่า ทีซีพี/ไอพี มีต้นกำเนิดมาจากโครงการ ARPANET นั้นเอง และต่อมาก็ได้ถูกรวมเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์ และก็ได้ถูกใช้กันอย่างแพร่หลายต่อมา ซึ่งปัจจุบันใช้งานอยู่ในแทบทุกเครือข่าย ไม่ว่าจะเป็นเครือข่ายเฉพาะที่หรือเครือข่ายในบริเวณกว้าง ทีซีพี/ไอพีเชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ หรือ อินเทอร์เน็ต (Internet)

2.2 การเชื่อมต่อของโปรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

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

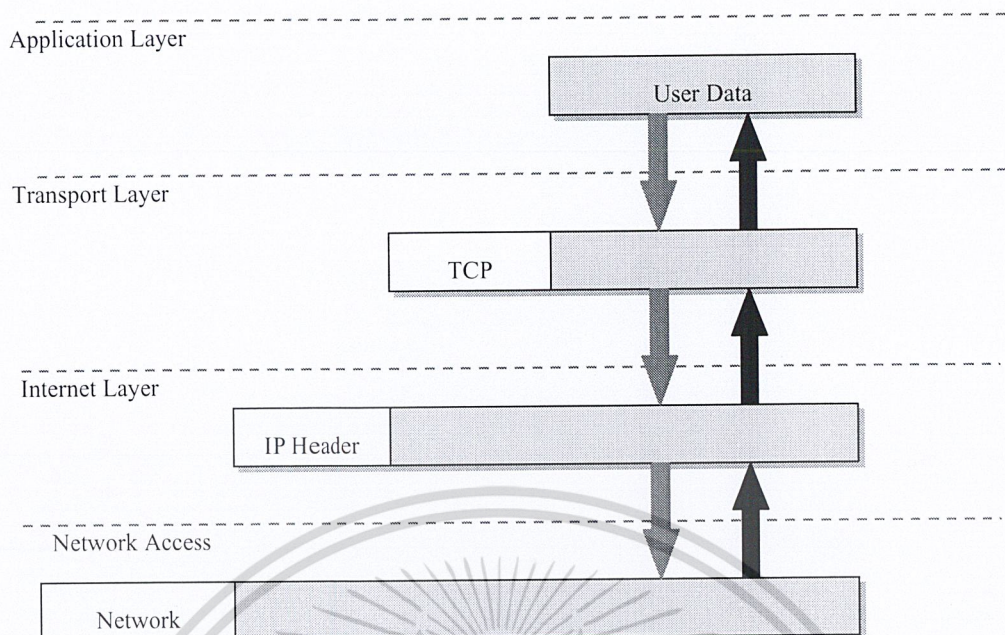


รูปที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอปพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังตารางที่ 2-1

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอปพลิเคชัน (Application Layer)	ชั้นนี้รองรับการทำงานของแอปพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอปพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอปพลิเคชันต่างๆมีการติดต่อกันตามแต่ละโพรโทคอลเฉพาะแล้วแต่แอปพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง
2. ชั้นทรานสปอร์ต (Transport Layer)	มีการสร้างการเชื่อมต่อนี้ระหว่างแอปพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อนี้เพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโทคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโทคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ชั้นนี้มีหน้าที่ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโทคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ตคือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโทคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และ เออาร์พี (Address Resolution Protocol: ARP)
4. ชั้นเน็ตเวิร์กอินเทอร์เฟซ (Network Interface Layer)	ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย

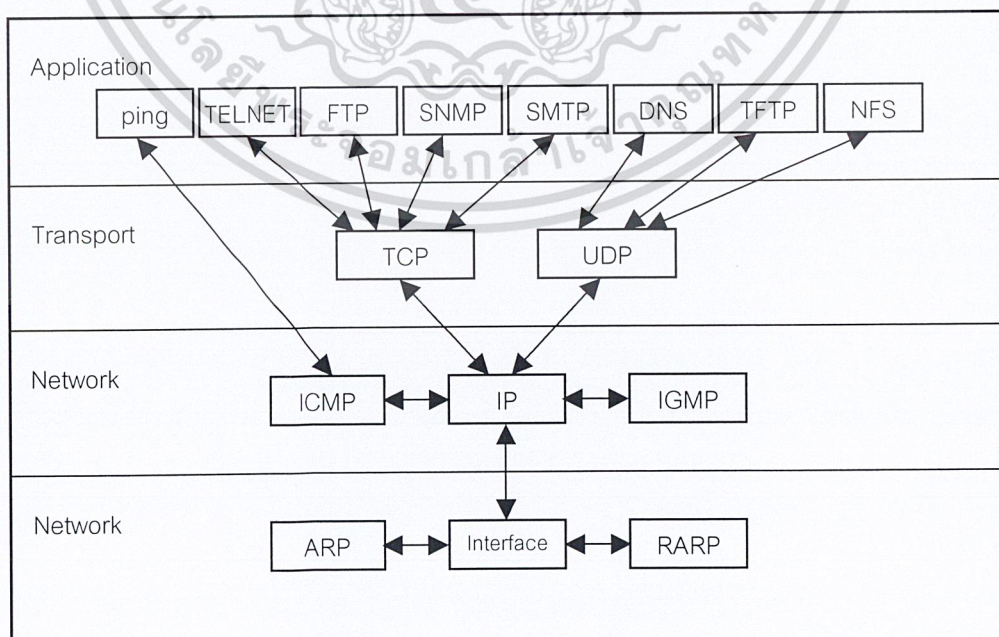
ตารางที่ 2-1 แสดงการทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี



รูปที่ 2-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

2.3 โพรโทคอลสแต็ก

การทำงานตามโปรแกรมประยุกต์หนึ่งๆ ไม่ได้ใช้โพรโทคอลพร้อมกันทั้งหมด หากแต่ใช้เพียงโพรโทคอลที่สัมพันธ์กันไปในแต่ละระดับชั้นของแบบอ้างอิง ตัวอย่างเช่น เทลเน็ต (Telnet) จะอาศัยทีซีพีและไอพี ตามลำดับ การซ้อนทับของโพรโทคอลจากระดับชั้นบนไปชั้นล่างเรียกว่า โพรโทคอลสแต็ก (Protocol Stack) ดังรูปที่ 2-3



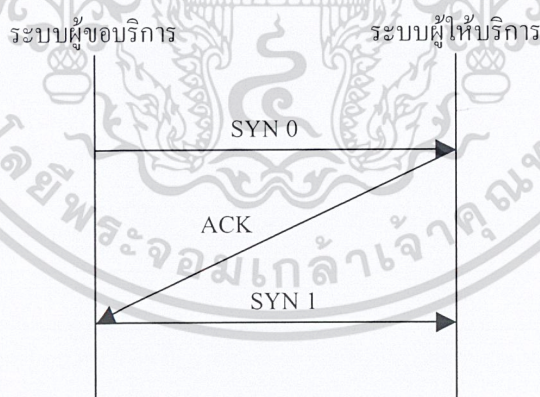
รูปที่ 2-3 แสดงโพรโทคอลสแต็กของทีซีพี/ไอพี

ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวิร์กตามรูปแบบเป็นแกนสำคัญของโปรโตคอลสแต็กเนื่องจากทั้งทีซีพีและยูดีพีต้องใช้ไอพีเพื่อเลือกเส้นทางส่งแพ็กเก็ตในในระดับชั้นเน็ตเวิร์กยังมีไอซีเอ็มพีสนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดพลาดที่เกิดขึ้นเนื่องจากการส่งแพ็กเก็ต และมีไอซีเอ็มพีดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัลติคาสต์ระดับชั้นทรานสปอร์ตมี 2 โปรโตคอลที่สำคัญ คือ ทีซีพีและยูดีพี แอปพลิเคชันจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงานโปรโตคอลระดับล่างถัดจากไอพีได้แก่ โปรโตคอล ระดับเน็ตเวิร์ก อินเทอร์เน็ตซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งาน ในระดับชั้นนี้มี โปรโตคอลในชุดของทีซีพี/ไอพี ทำหน้าที่สนับสนุนการทำงานอยู่สองโปรโตคอล คือ เออาร์พี และ อาร์เออาร์พี ทั้งสองโปรโตคอลทำหน้าที่แปลค่าระหว่างแอดเดรสไอพี กับ ฮาร์ดแวร์แอดเดรส

ในชุดโปรโตคอลทีซีพี/ไอพีนี้ มีโปรโตคอลหลักที่ขอกกล่าวถึง 5 โปรโตคอล ได้แก่ โปรโตคอลทีซีพี โปรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโปรโตคอลไอพี โปรโตคอลเออาร์พี โปรโตคอลไอซีเอ็มพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

2.4 โปรโตคอลทีซีพี (TCP: Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งของโปรโตคอลทีซีพี คือ การทำ “3-way Handshake” ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมาจึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2-4



รูปที่ 2-4 แสดงการทำ 3-way Handshake

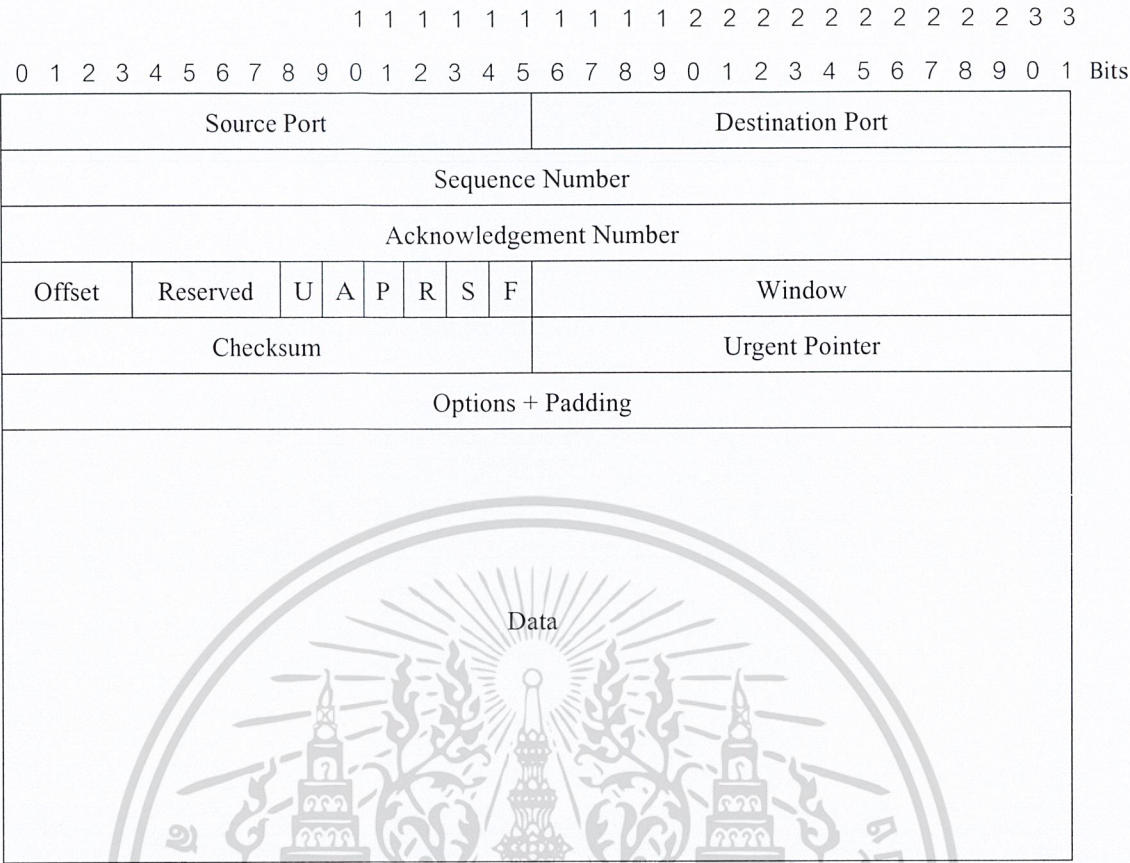
การเชื่อมต่อแบบ 3-way handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่งและฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

ดังนั้น โปรโตคอลทีซีพีจึงเป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ “Connection Oriented” ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

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

ส่วนประกอบของทีซีพีเฮดเดอร์

1. Source Port : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
2. Destination Port : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
3. Sequence Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
4. Acknowledgement Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติ 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 - แสดงเมื่อรีเซตการเชื่อมต่อ
 - SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครไนส์
 - FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว
7. Window : เป็นเลขบอกจำนวนของอ็อกเตต (octet) ของข้อมูล จัดการในส่วนของการ end-to-end flow control
8. Checksum : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล
9. Urgent Pointer : เป็นตัวชี้ตำแหน่งของ Urgent Data
10. Option and Padding : เป็นตัวบอกออฟชั่นของโปรเซสที่ใช้ทีซีพี
11. Data : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)



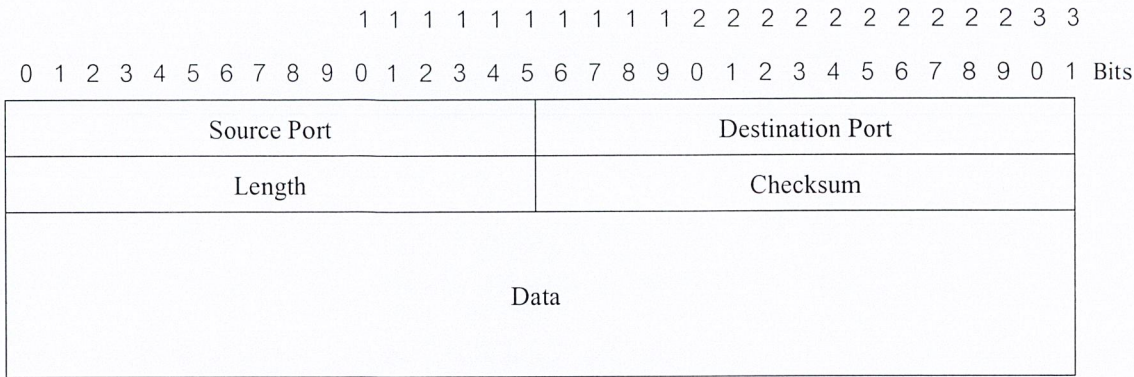
รูปที่ 2-5 แสดงแพ็กเก็ตทีซีพี

2.5 โพรโทคอลยูดีพี (UDP: User Datagram Protocol)

โพรโทคอลยูดีพีเป็นโพรโทคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือ จัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือ ทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโทคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

ส่วนประกอบของ UDP Frame

- 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

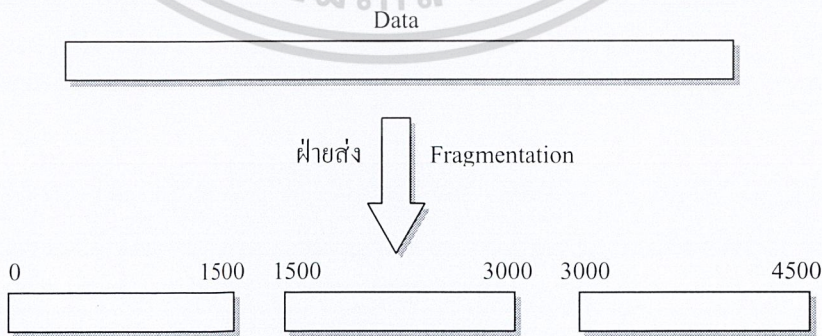


รูปที่ 2-6 แสดงแพ็กเก็ตยูดีพี

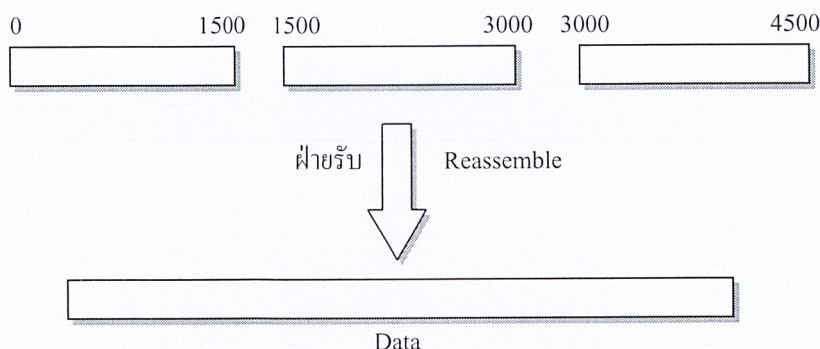
2.6 โพรโทคอลไอพี (IP: Internet Protocol)

โพรโทคอลไอพีเป็นโพรโทคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนที่ซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่

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



รูปที่ 2-7 แสดงการทำแฟร็กเมนเตชัน



รูปที่ 2-8 แสดงการรีแอสเซมเบิล

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 เป็นค่าประจำตัวของไอฟินั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ
6. Flag : เป็นตัวเลข 3 bit บอกลักษณะของแพ็กเก็ตว่ามีการแฟร็กเมนต์หรือไม่
 Bit 0 : สแกนไว้ปกติเป็น 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 โดยกำหนดค่าเป็นจำนวนเรเตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปที่ค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเรเตอร์ 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

1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 Bits

Ver	IHL	Type of Service	Total Length	
Identifier			Flags	Fragment
Time to Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options + Padding				
Data				

รูปที่ 2-9 แสดงแพ็กเก็ตไอพี

2.7 โพรโทคอลเออาร์พี(ARP:Address Resolution Protocol)

โพรโทคอลเออาร์พีเป็นโพรโทคอลที่ออกแบบมาเพื่อใช้ในเครือข่ายที่สนับสนุนการบรอดคาสต์ ถูกเรียกใช้งานโดยโพรโทคอลไอพีเพื่อช่วยแปลงหมายเลขไอพี ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัยการ์ดแลน(LAN card) ติดตั้งอยู่ที่แลนการ์ดนี้จะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใครเพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานใน โพรโทคอล ทีซีพี/ไอพี ก็จะต้องมีการกำหนดหมายเลขแอดเดรสไอพี ประจำตัวเพื่อใช้อ้างอิงกัน และ โพรโทคอลเออาร์พี จะทำหน้าที่แปลงค่าหมายเลขไอพีให้เป็นหมายเลขฮาร์ดแวร์จริงในระดับการทำงานที่ชั้นอินเทอร์เน็ตนี้ ซึ่งกลไกการแปลงนี้เรียกว่า address resolution

hardware		protocol
HLEN	PLEN	operation
Sender HA (octets 0-3)		
Sender HA (octets 4-5)		Sender IA (octets 0-1)
Sender IA (octets 2-3)		Target HA (octets 0-1)
Target HA (octets 2-5)		
Target IA (octets 0-3)		

รูปที่ 2-10 แสดงเออาร์พีตาต้าแกรม

ส่วนประกอบของเออาร์พีตาต้าแกรม

1. Hardware 16 บิต : กำหนดชนิดของฮาร์ดแวร์เครือข่ายที่เออาร์พีทำงานอยู่ ค่าใช้งานมีตัวอย่างดังต่อไปนี้
- 1 อีเทอร์เน็ต

4 โทเค็นริง

5 เคออส(chaos)

6 เครือข่าย IEEE 802

7 อาร์คเน็ต

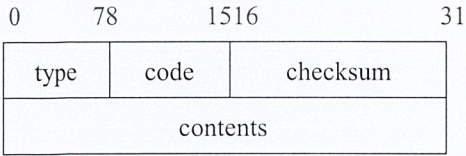
12 โคลัลทอลล์
2. protocol 16 บิต : ชนิดของโพรโทคอลที่ร้องขอใช้เออาร์พี
3. HLEN 8 บิต : ขนาดของฮาร์ดแวร์แอดเดรสเป็นจำนวนไบต์ ค่าปกติที่ใช้งาน คือ 6 ซึ่งเท่ากับขนาด 6 ไบต์ของอีเทอร์เน็ตฮาร์ดแวร์แอดเดรส
4. PLEN 8 บิต : ขนาดของแอดเดรสระดับเน็ตเวิร์กเป็นจำนวนไบต์ ค่าปกติที่ใช้ คือ 4 ซึ่งเท่ากับขนาด 4 ไบต์ของไอพีแอดเดรส

5. Operation 16 บิต : กำหนดรูปแบบการใช้ดาต้าแกรม ค่าในฟิลด์นี้ใช้กำหนดการทำงานของทั้งเออาร์พีและอาร์เออาร์พี ซึ่งมี 4 ค่า คือ
- ARP request (ค่าเท่ากับ 1)
 - ARP reply (ค่าเท่ากับ 2)
 - RARP request (ค่าเท่ากับ 3)
 - RARP reply (ค่าเท่ากับ 4)
6. Address : ฟิลด์แอดเดรสเรียงลำดับจากฮาร์ดแวร์และเน็ตเวิร์กแอดเดรสของสถานีที่ร้องขอ ตามด้วยฮาร์ดแวร์และเน็ตเวิร์กแอดเดรสของสถานีที่ตอบรับ

2.8 โพรโทคอล ไอซีเอ็มพี (ICMP : Internet Control Message Protocol)

หน้าที่หลักของ โพรโทคอล ไอซีเอ็มพี คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้งานทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบ คือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้ โพรโทคอล ไอซีเอ็มพี ยังถูกเรียกใช้งานจากเครื่องเซิร์ฟเวอร์ และ เราเตอร์อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโพรโทคอลไอซีเอ็มพีนั้นจะทำงานควบคู่กับโพรโทคอลไอพีในระดับเดียวกัน และข้อความต่างๆที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูลของไอพี(ไอพิดาต้าแกรม) อีกทีหนึ่ง ข้อความที่โพรโทคอลไอซีเอ็มพีส่งนั้น แบ่งออกได้ 2 แบบ คือ ICMP error message หรือข้อความแจ้งข้อผิดพลาด และ ICMP query หรือข้อความเรียกขอข้อมูลเพิ่มเติม ตัวอย่างการทำงานของ โพรโทคอล ไอซีเอ็มพี เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่เราเตอร์จะส่งข้อความแจ้งเป็น ไอซีเอ็มพี message ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อความก็จะมีส่วนของข้อมูลไอพิดาต้าแกรมที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะได้ทราบว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด

ดังนั้นโพรโทคอล ไอซีเอ็มพี จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบว่าเครื่องเซิร์ฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่ายอินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping มีการเรียกใช้งานโพรโทคอล ไอซีเอ็มพี แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง



รูปที่ 2-11 แสดงฟอร์แมตของ ไอซีเอ็มพี

1. Type ขนาด 8 บิต : กำหนดค่าความผิดพลาดและการรายงานสถานะ การใช้งานในปัจจุบันมีทั้งหมด 15 ประเภท

2. code ขนาด 8 บิต : รหัสความผิดพลาดย่อย
3. Checksum ขนาด 16 บิต : ค่าผลรวมตรวจสอบแบบ 1's complement สำหรับใช้ตรวจสอบความผิดพลาด โดยคำนวณผลรวมของ type,code และ contents
4. Contents ขนาดไม่คงที่ : ฟิลด์นี้ใช้บรรจุข้อมูลข่าวสารเพิ่มเติมเพื่อแจ้งกลับซึ่งจะขึ้นอยู่กับค่า type และ code

2.9 การโจมตีเพื่อให้บริการสำหรับโพรโทคอลสแต็กที่ซีพี/ไอพี

2.9.1 ความหมายของการโจมตีเพื่อให้บริการ

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

2.9.2 ประเภทของการโจมตีเพื่อให้บริการ

ในที่นี้ประเภทของการโจมตีสามารถแบ่งได้ดังต่อไปนี้

2.9.2.1 ประเภทอยู่ในชั้นทรานสปอร์ต หรือชั้นอินเทอร์เน็ต

การโจมตีในระดับชั้นนี้สามารถแบ่งได้เป็น 2 แบบหลักๆ ได้แก่

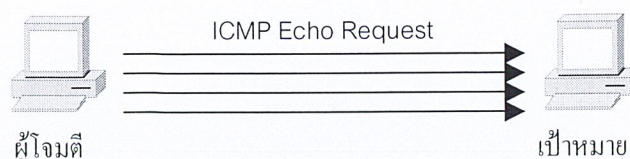
2.9.2.1.1 การส่งแพ็กเก็ตจำนวนมาก (Amount of Packets Sending)

การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตเกิดปริมาณมากเข้าไปยังระบบเป้าหมาย อาจทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่าง หรือไม่สามารถทำงานต่อไปได้ ซึ่งแพ็กเก็ตที่ส่งออกไปนี้สามารถแบ่งออกได้เป็น

(1) แพ็กเก็ตข้อมูล (Data Packets)

การโจมตีวิธีนี้ทำได้โดยการส่งแพ็กเก็ตข้อมูลปริมาณมาก เมื่อข้อมูลเข้ามาสู่เครื่องเป้าหมายก็เก็บไว้ในบัฟเฟอร์ก่อนนำมาประมวลผลอีกครั้ง ดังนั้นหากส่งแพ็กเก็ตเข้ามาเป็นปริมาณมาก อาจทำให้บัฟเฟอร์ของเครื่องเป้าหมายไม่เพียงพอที่จะสามารถรองรับแพ็กเก็ตเหล่านั้นได้ทั้งหมด ซึ่งอาจทำให้เครื่องเป้าหมายให้บริการได้ช้าลง หรือต้องหยุดการให้บริการไปเลย ตัวอย่างการโจมตีประเภทนี้เช่น Ping Flood Attack เป็นต้น

Ping Flood เป็นการโจมตีในยุคแรกๆ ของ DoS หลักการคือส่ง ICMP Echo Request (รูปแบบเดียวกับคำสั่ง Ping) ไปยังเป้าหมายหลายๆ ในระยะเวลาติดต่อกัน ทำให้เป้าหมายต้องคอยตอบ ICMP Echo Reply ตลอดเวลาจนไม่สามารถให้บริการอย่างอื่นได้ ความรุนแรงของการโจมตีขึ้นอยู่กับปริมาณแพ็กเก็ตที่โจมตีไปยังเครื่องเป้าหมาย หากเครื่องที่ทำการโจมตีมีประสิทธิภาพสูงและเครือข่ายมีแบนด์วิดท์มาก อาจส่งผลทำให้เครื่องเป้าหมายหยุดการทำงานลงได้



รูปที่ 2-12 แสดงการโจมตีด้วย Ping Flood Attack

ข้อสังเกตสำหรับการโจมตีประเภทนี้คือ จะปรากฏแพ็กเก็ต ICMP Echo Request และ ICMP Echo Reply ปริมาณมหาศาลซึ่งมีการรับส่งกันระหว่างเครื่องเป้าหมายที่ถูกโจมตีกับเครื่องอื่นๆ ซึ่งอาจมีหรือไม่มีในอินเทอร์เน็ตก็ได้ เนื่องจากกระบวนการสำคัญอย่างหนึ่งของการโจมตีลักษณะนี้คือ ผู้โจมตีต้องปลอมหมายเลขไอพี (IP Spoofing) เสมอ เพื่อป้องกันไม่ให้แพ็กเก็ต ICMP Echo Reply ถูกส่งกลับมายังเครื่องตัวเอง ซึ่งจะทำให้ผู้โจมตีได้รับผลกระทบจากการโจมตีด้วย และการปลอมไอพียังเป็นหลักประกันได้ว่าจะไม่สามารถติดตามได้ว่าผู้ใดเป็นผู้โจมตี รูปแบบแพ็กเก็ตที่เกิดขึ้นจากการโจมตีมีลักษณะดังนี้

```

14:49:43.217137 62.51.12.23      > 10.1.1.10      : icmp: echo request
14:49:43.217175 10.1.1.10      > 62.51.12.23    : icmp: echo reply
14:49:43.217195 62.51.12.23      > 10.1.1.10      : icmp: echo request
14:49:43.217219 10.1.1.10      > 62.51.12.23    : icmp: echo reply
14:49:43.217245 96.141.106.124   > 10.1.1.10      : icmp: echo request
14:49:43.217279 10.1.1.10      > 96.141.10.124  : icmp: echo reply
14:49:43.219017 172.19.251.18    > 10.1.1.10      : icmp: net 162.75.127.79
unreachable
14:49:43.237136 75.126.62.65     > 10.1.1.10      : icmp: echo request
14:49:43.237169 10.1.1.10      > 75.126.62.65   : icmp: echo reply
14:49:43.237193 75.126.62.65     > 10.1.1.10      : icmp: echo request
14:49:43.237216 10.1.1.10      > 75.126.62.65   : icmp: echo reply
14:49:43.237240 218.155.179.58   > 10.1.1.10      : icmp: echo request
14:49:43.237272 10.1.1.10      > 218.155.17.58  : icmp: echo reply

```

รูปที่ 2-13 แสดง รูปแบบแพ็กเก็ตที่เกิดขึ้นจากการโจมตีจาก Ping Flood Attack

นอกจาก Ping Flood Attack จะสร้างความเสียหายแก่เครื่องเป้าหมายแล้วยังสร้างความเสียหายแก่ระบบเครือข่ายของเครื่องเป้าหมายด้วย เพราะการโจมตีวิธีนี้จะสร้างแพ็กเก็ตเป็นจำนวนมากขึ้นใน

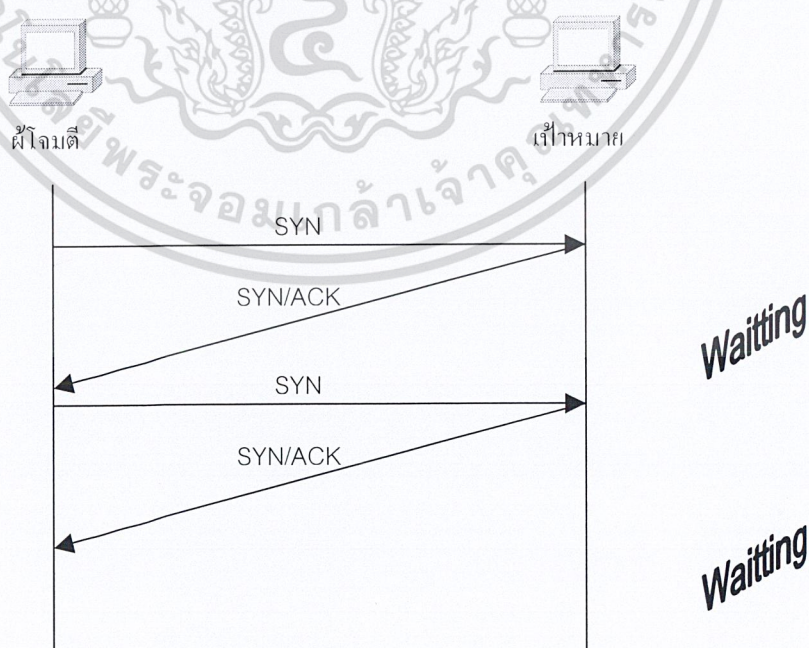
เครือข่ายที่เครื่องเป้าหมายตั้งอยู่ ทำให้ระบบเครือข่ายเกิดความคับคั่งของข้อมูล(Congestion) อาจส่งผลให้เครือข่ายเป็นอัมพาตได้

การป้องกันการการโจมตีลักษณะนี้ทำได้โดยการกำหนดที่อุปกรณ์เราเตอร์หรือไฟร์วอลล์ โดยกำหนดไม่ให้แพ็กเก็ต ICMP Echo เข้ามายังเซิร์ฟเวอร์ แต่อย่างไรก็ตามแพ็กเก็ต ICMP Echo ได้ถูกใช้ในโปรแกรม Ping หากมีการปิดกั้นแพ็กเก็ต ICMP Echo จะทำให้ไม่สามารถตรวจสอบสถานะของเซิร์ฟเวอร์ได้ แนวทางที่ควรปฏิบัติคือกำหนดแบนด์วิดธ์ของแพ็กเก็ต ICMP Echo ให้เหมาะสมในอุปกรณ์เราเตอร์ โดยให้เพียงพอต่อการใช้งานสำหรับการตรวจสอบสถานะของระบบ แต่ไม่มากจนทำให้เกิดการโจมตีได้

(2) แพ็กเก็ตสำหรับการควบคุม (Control Packets)

นอกจากแพ็กเก็ตที่เป็นตัวข้อมูลแล้ว ยังมีแพ็กเก็ตอีกรูปแบบหนึ่งที่สำคัญมากสำหรับการติดต่อสื่อสารบนโปรโตคอลที่ซีพี/ไอพี คือแพ็กเก็ตส่วนการควบคุม ตัวอย่างแพ็กเก็ตประเภทนี้คือสัญญาณ SYN หรือ ACK สำหรับการสถาปนาการเชื่อมต่อ หรือสัญญาณ Fin สำหรับการยกเลิกการเชื่อมต่อ เป็นต้น

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 2-14 ซึ่งการเปิดการเชื่อมต่อรอเอาไว้วันต้องใช้ทรัพยากรของระบบส่วนหนึ่ง และหากมีการส่งสัญญาณในลักษณะนี้มาเรื่อยๆ และทรัพยากรของระบบมีไม่เพียงพอ อาจทำให้ระบบไม่สามารถให้บริการอย่างอื่น หรือให้บริการกับผู้อื่นต่อไปได้



รูปที่ 2-14 แสดงการส่งแพ็กเก็ตแบบ SYN Flood

สำหรับการโจมตีแบบ SYN Flood จะทำให้เกิดแพ็กเก็ตเกิดในระบบเครือข่ายในลักษณะดังรูป

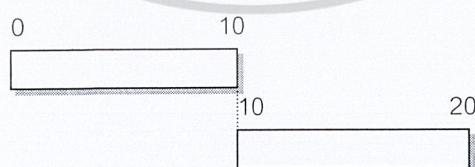
```
10:09:43.137 10.0.0.1 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.139 10.0.0.2 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.143 10.0.0.3 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.149 10.0.0.4 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.158 10.0.0.5 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.165 10.0.0.6 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.177 10.0.0.7 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
10:09:43.185 10.0.0.8 > isag11.ce.kmitl.ac.th.80: S 399259715:399259715(0)
```

รูปที่ 2-15 แสดง แพ็กเก็ตที่เกิดขึ้นจากการโจมตีแบบ SYN Flood

ในปัจจุบันนี้การโจมตีแบบ SYN Flood ยังเป็นการโจมตีที่ได้ผลและหาทางป้องกันได้ยาก เนื่องจากยากที่จะแยกลักษณะของแพ็กเก็ตที่ใช้ในการโจมตีกับแพ็กเก็ตที่ขอเริ่มต้นเชื่อมต่อทั่วไป นอกจากนี้ไฟร์วอลล์หรือเราเตอร์ทั่วไปยังไม่สามารถป้องกันการโจมตีประเภทนี้ได้อย่างสมบูรณ์ หนทางที่เป็นไปได้คือการใช้ระบบตรวจจับผู้บุกรุกทางระบบเครือข่ายทำการตรวจจับการโจมตี เพื่อนำข้อมูลจากการโจมตีกลับไปตั้งค่าอุปกรณ์เราเตอร์หรือไฟร์วอลล์เพื่อป้องกันการโจมตีมายังเซิร์ฟเวอร์

2.9.2.1.2 ความผิดปกติของแฟร็กเมนต์ (Abnormal Fragmentation)

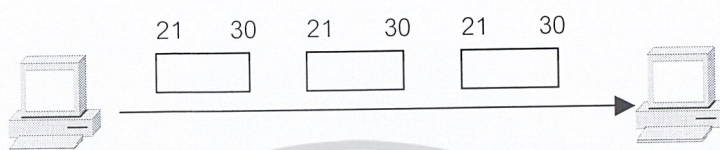
การโจมตีวิธีนี้อาศัยหลักการแฟร็กเมนต์เซชันและรีแอสเซมเบิลที่กล่าวไว้ข้างต้น โดยทำให้แพ็กเก็ตนั้นต้องมีการรีแอสเซมเบิล (กำหนดค่า MF flag = 0) ซึ่งปกติการรีแอสเซมเบิลแพ็กเก็ตทั้งหมดต้องสามารถเชื่อมต่อกันได้สนิท ดังรูปที่ 2-16 แต่แพ็กเก็ตที่ผู้บุกรุกส่งไปมีการแก้ไขข้อมูลในบางฟิลด์ ทำให้เกิดความผิดปกติในกระบวนการรีแอสเซมเบิล ซึ่งการโจมตีในลักษณะนี้ แบ่งได้ดังต่อไปนี้



รูปที่ 2-16 แสดงการรีแอสเซมเบิลแบบปกติ

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ (Abnormal Sequences of Packets Sending)

ปกติการส่งแพ็กเก็ตมักเรียงตามลำดับกันไป หากไม่เรียงลำดับก็ต้องรอจนกว่าแพ็กเก็ตก่อนหน้านี้มาถึง เพื่อเรียงลำดับแพ็กเก็ตที่เครื่องรับ แต่การ โจมตีแบบนี้กลับส่งเฉพาะแพ็กเก็ตสุดท้าย เพื่อให้ระบบเป้าหมายรอแพ็กเก็ตก่อนหน้านี้ และส่งไปเป็นปริมาณมาก ๆ ซึ่งจะส่งผลให้ระบบต้องจองทรัพยากรส่วนหนึ่งเพื่อรองรับแพ็กเก็ตที่ต้องรอเหล่านั้นในปริมาณมาก จนกระทั่งระบบไม่สามารถจัดหาทรัพยากรได้เพียงพอ ส่งผลให้ระบบไม่สามารถให้บริการอย่างอื่นได้



รูปที่ 2-17 แสดงการส่งเฉพาะแพ็กเก็ตสุดท้ายไปยังเป้าหมาย

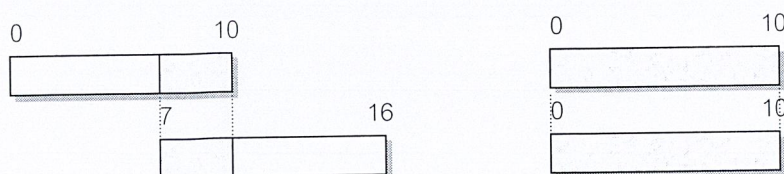
โดยปกติแล้วการ โจมตีในรูปแบบนี้ผู้โจมตีจะแก้ไขข้อมูลในฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพี ซึ่งเป็นส่วนที่แสดงลำดับของข้อมูลหลังจากกระบวนการแฟร็กเมนต์เตชัน โดยแก้ไขให้ส่งแพ็กเก็ตสุดท้ายหรือแพ็กเก็ตหลังๆ เพียงแพ็กเก็ตเดียวเลย ทำให้ระบบเป้าหมายต้องรอแพ็กเก็ตก่อนหน้านี้

(2) การส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน (Overlapped Packets' Size Sending)

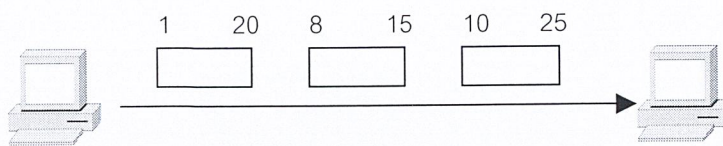
ปกติแพ็กเก็ตที่ส่งมาต้องนำมาต่อกันที่ระบบเป้าหมายได้พอดี แต่การ โจมตีแบบนี้เป็นการส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน หรือซ้อนทับกัน ทำให้ข้อมูลเมื่อมาต่อกันแล้วเกิดความผิดพลาด หรือไม่สามารถเชื่อมต่อกันได้

โดยปกติแล้วการ โจมตีแบบนี้ ผู้บุกรุกสามารถแก้ไขข้อมูลได้ 2 แห่งใหญ่ๆ ได้แก่

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



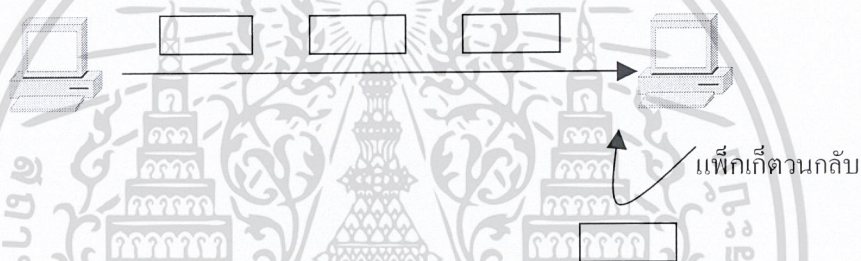
รูปที่ 2-18 แสดงการรีแอสเซมเบิลแบบแพ็กเก็ตมีขนาดเหลื่อมกัน



รูปที่ 2-19 แสดงการโจมตีโดยส่งแพ็กเก็ตที่ไม่สามารถรีแอสเซมเบิลได้

(3) การส่งแพ็กเก็ตแบบวนลูป (Looping)

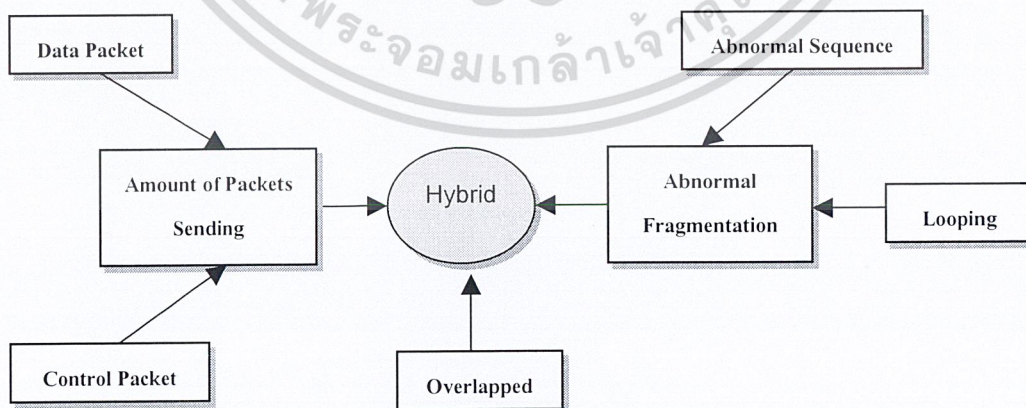
คือ การส่งโดยกำหนดค่าแอดเดรสต้นทาง (Source Address) และแอดเดรสปลายทาง (Destination Address) ให้เหมือนกันทำให้เกิดการรับส่งวนไปวนมาอยู่ที่เครื่องเป้าหมายเอง เช่น LAND ซึ่งเป็นโปรแกรมโจมตีที่มีการกำหนดแอดเดรสต้นทาง และแอดเดรสปลายทางเป็นค่าเดียวกัน คือ เป็นแอดเดรสของเครื่องเป้าหมายนั่นเอง ทำให้เกิดการส่งวนไปวนมาอยู่ที่เครื่องเป้าหมาย



รูปที่ 2-20 แสดงการโจมตีโดยส่งแพ็กเก็ตแบบวนลูป

2.9.2.1.3 แบบผสม (Hybrid)

คือ การโจมตีที่อาศัยวิธีการผสมกันระหว่างสามแบบแรกที่ได้กล่าวมาแล้ว ดังรูปที่ 2-21



รูปที่ 2-21 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กที่ซีพี/ไอพี

2.9.2.2 ประเภทอยู่ในชั้นแอปพลิเคชัน

การโจมตีประเภทอื่นนอกจากที่ได้กล่าวมาแล้วข้างต้น ส่วนใหญ่เกิดจากการใช้จุดอ่อนหรือข้อผิดพลาดของแอปพลิเคชันที่เครื่องเป้าหมายใช้อยู่ในการโจมตีเครื่องเป้าหมายเอง ไม่ว่าจะเป็นจุดอ่อนของระบบปฏิบัติการ หรือข้อผิดพลาดของซอฟต์แวร์ก็ตาม

ในกรณีเช่นนี้เจ้าของเครื่องสามารถแก้ไขได้เอง โดยการนำโปรแกรมแพช (Patch) หรือเซอร์วิสแพ็ค (Service Pack) ต่างๆ มาลงเพื่อแก้ไขข้อผิดพลาดเหล่านี้ หรือหลีกเลี่ยงไปใช้โปรแกรมอื่นที่ไม่เกิดปัญหา ซึ่งการโจมตีในลักษณะนี้ไม่อยู่ในขอบเขตที่ศึกษา

2.9.2.3 โปรแกรมที่ใช้ในการโจมตีเพื่อให้ปิดบริการ

โปรแกรมที่ใช้ในการ โจมตีเพื่อให้ปิดบริการนั้นเกิดขึ้นมาอย่างมากมาย และนับวันจะเพิ่มรูปแบบมากขึ้นเรื่อยๆ แต่โปรแกรมในปัจจุบันยังคงมีรูปแบบการ โจมตีไม่มากไปกว่าที่ได้กล่าวมาแล้ว ซึ่งโปรแกรมต่างๆจะมีตัวอย่างดังตารางที่ 2-1

ชื่อโปรแกรม	ประเภท	โปรโตคอลที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
1. syncclood	ส่งแพ็กเก็ตสำหรับควบคุมในปริมาณมาก	TCP	เป็นการส่งสัญญาณ SYN ไปขอเปิดการเชื่อมต่อ แล้วเมื่อได้รับ ACK ก็ไม่ส่งสัญญาณ SYN กลับไป ดังที่ได้อธิบายไปแล้ว	- Windows 95
2. oshare	แฟร็กเมนต์ชนผิดปกติ (แฟกเก็ตหล่นซ้ำกัน)	IP	เป็นการส่งแพ็กเก็ตที่เหมือนกันมาที่เครื่องเป้าหมาย ทำให้แฟกเก็ตที่ส่งมาซ้อนทับกัน	- Windows 95/98 - Windows NT 4 + Service Pack 5 ลงมา
3. land	แฟร็กเมนต์ชนผิดปกติ (ส่งแพ็กเก็ตแบบวนลูป)	IP	เป็นการส่งแพ็กเก็ตที่มีแอดเดรสต้นทางและปลายทางเป็นค่าเดียวกัน ทำให้เกิดการส่งแบบวนลูป	- Windows 95
4. smurf	แพ็กเก็ตเกิดปริมาณมาก	ICMP	เป็นการส่ง ICMP echo (ping) traffic จำนวนมากโดยการ Broadcast ไปยังเครื่องเป้าหมาย	- Windows 95

ตารางที่ 2-2 แสดงตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ

ชื่อโปรแกรม	ประเภท	โปรโตคอล ที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
5. opentear	แฟร็กเมนต์ชั้น ผิปกติ (ต้องรอ แพ็กเก็ตก่อน หน้า)	UDP	เป็นการสร้างแพ็กเก็ตสุดท้ายมา เพื่อหลอกให้ระบบเป้าหมายรอ แพ็กเก็ตก่อนหน้านี้	- Windows 95
6. pimp	แฟร็กเมนต์ชั้น ผิปกติ (รีแอส เซมเบิลแล้วมี ปัญหา)	IP,IGMP	เป็นการสร้างแพ็กเก็ตที่เมื่อรี แอสเซมเบิลแล้วได้ข้อมูลที่ไม่ มีความหมาย เกิดเป็นขยะใน ระบบ	- Windows 95
7. targa	แฟร็กเมนต์ชั้น ผิปกติ (แพ็ก เก็ตหล่นล้ำกัน)	IP	เป็นการส่งแพ็กเก็ตที่เหมือนกัน มาที่เครื่องเป้าหมายทำให้แพ็ก เก็ตที่ส่งมาซ้อนทับกัน	- Windows 95/98 - Windows NT 4 + Service Pack 5 ลงมา
8. Gin	แพ็กเก็ตสำหรับการ ควบคุมใน ปริมาณมาก	ICMP	เป็นการส่ง ICMP Echo ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98 - Modem
9. raped	แพ็กเก็ตสำหรับการ ควบคุมใน ปริมาณมาก	TCP	เป็นการส่งสัญญาณ SYN ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98
10. stream	แพ็กเก็ตเกิดปริมาณ มาก	TCP	เป็นการส่งแพ็กเก็ต TCP ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 95/98 - Modem
11. moya	แพ็กเก็ตสำหรับการ ควบคุมใน ปริมาณมาก	ICMP	เป็นการส่ง ICMP Echo (Ping) ปริมาณมากไปยังเครื่องเป้าหมาย	- Windows 98
12. Kox	แฟร็กเมนต์ชั้น ผิปกติ (แพ็ก เก็ตหล่นล้ำกัน)	IGMP	ส่งแพ็กเก็ตที่หล่นล้ำกันไปยัง เครื่องเป้าหมายไม่สามารถ ประกอบแพ็กเก็ตเหล่านั้นได้	- Windows 98/98SE - Windows 2000 build 2000

ตารางที่ 2-2 แสดงตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ (ต่อ)

ชื่อโปรแกรม	ประเภท	โปรโตคอล ที่ใช้	ลักษณะการโจมตี	ระบบที่มีปัญหา
13. Sesquipedaliam	แฟร์ริเมนต์ชัน ผิดปกติ	UDP	ส่งแพ็กเก็ต UDP ที่มีแฟร์ริเมนต์ ชันผิดปกติไปยังเครื่องเป้าหมาย	- Windows 98/98SE
14. smurf	แพ็กเก็ตสำหรับ การ ควบคุมใน ปริมาณมาก	ICMP	ส่งแพ็กเก็ต ICMP Echo (Ping) ไปยังเครื่องเป้าหมายเป็น ปริมาณมาก	- Windows 95/98
15. WinArp	อยู่ในชั้นแอปพลิเคชัน	ARP	เป็นการส่งแพ็กเก็ตไปยังเครื่อง ปลายทาง โดยเครื่องปลายทาง ต้องกดปุ่ม “OK” สำหรับทุก แพ็กเก็ตที่เข้ามา	- Windows 95/98 - Windows NT 4.0 ลงมา
16. winnuke	อยู่ในชั้นแอปพลิเคชัน	NetBIOS (Port 139)	เป็นการส่ง OOB (Out Of Band) ดาต้าไปยังพอร์ต 139 (NetBIOS) ทำให้เกิดหน้าจอฟ้า ขาวการเชื่อมต่อกับอินเทอร์เน็ต และทำให้เกิดปัญหาในการติดต่อ บนเครือข่าย	- WFWG 3.11 - Windows 95 - Windows NT 3.51 - Windows NT 4.0 + Service Pack 4 ลงมา
17. jolt2	แฟร์ริเมนต์ชัน ผิดปกติ (แพ็ก เก็ตหลอกลวง กัน)	ICMP	เป็นการส่งแพ็กเก็ตที่เหมือนกัน มาที่เครื่องเป้าหมาย ทำให้แพ็ก เก็ตที่ส่งมาซ้อนทับกัน	- WFWG 3.11 - Windows 95 - Windows NT 3.51 - Windows NT 4.0 + Service Pack 4 ลงมา - Windows 2000

ตารางที่ 2-2 แสดงตัวอย่างโปรแกรมที่โจมตีเพื่อให้ปิดบริการ (ต่อ)

บทที่ 3

การปิดการให้บริการแบบกระจาย

(Distributed Denial of Service : DDoS)

เป็นปัญหาด้านความปลอดภัยของระบบคอมพิวเตอร์ ที่มีการใช้งานการติดต่อสื่อสารทางเครือข่ายคอมพิวเตอร์หลายรูปแบบ ในปัจจุบันกำลังเป็นปัญหาที่ได้รับความสนใจทั้งจากนักคอมพิวเตอร์ , นักวิชาการ , นักวิทยาศาสตร์ , ผู้ประกอบการเกี่ยวกับคอมพิวเตอร์ และผู้สนใจเกี่ยวกับคอมพิวเตอร์โดยทั่วไป เพราะความร้ายแรงของปัญหาไม่ได้จำกัดในรูปแบบของการโจมตี (Intrusion) และไม่ได้ค่อยไปกว่าการโจมตีในรูปแบบอื่นๆเลย อีกทั้งความเสียหายที่เกิดขึ้นมาจากการใช้การโจมตีด้วยทุลแบบนี้ ยิ่งทวีความรุนแรงเพิ่มมากขึ้น และยากต่อการป้องกันจากการโจมตี

ก่อนที่จะเราจะเข้าใจ “การปิดการให้บริการแบบกระจาย” (Distributed Denial Of Service : DDoS) เราจะต้องเข้าใจการทำงานพื้นฐานของ “การปิดการให้บริการ” (Denial Of Services : DoS) เสียก่อน

3.1 พื้นฐานของการโจมตีในรูปแบบการปิดการให้บริการ



รูปที่ 3-1 แสดงพื้นฐานของการปิดการให้บริการ

จากรูปข้างต้น จะเป็นการแสดงการทำงานพื้นฐานของการโจมตีการปิดการให้บริการ โดยมี Attacker คือ ผู้โจมตี และควบคุมการทำงานของ Zombie คือ โปรแกรมที่ทำงานโจมตีไปยังเป้าหมาย Victim คือ เป้าหมายที่ถูกโจมตี

การปิดการให้บริการ (Denial Of Services : DoS)

การโจมตีในรูปแบบนี้ แตกต่างจากการโจมตี (Intrusion) เพราะว่าไม่จำเป็นต้องพยายามเข้าไปในระบบคอมพิวเตอร์ แต่มันสามารถทำให้คอมพิวเตอร์หยุดการทำงานได้ เพราะว่าการโจมตีที่เห็นเป็นส่วนใหญ่ เราพยายามป้องกันการโจมตี โดยพยายามไม่ให้พวกแฮกเกอร์ (Hacker) , แครกเกอร์ (Cracker) และผู้มีเจตนาหรือไม่เจตนาทำความเสียหายให้กับคอมพิวเตอร์ เข้ามาในระบบ ซึ่งแตกต่างจากการปิดการให้บริการสามารถติดต่อเข้ามาได้อย่างง่ายดาย การโจมตีจะพยายามติดตั้งโปรแกรมประเภท รุคคิต

(Rootkit) เป็นโปรแกรมประเภทดักจับพาสเวิร์ด (Password) หรือไม่ก็ใช้การดักจับไอพี (IP Sniffing) หรือวิธีการอื่นๆ

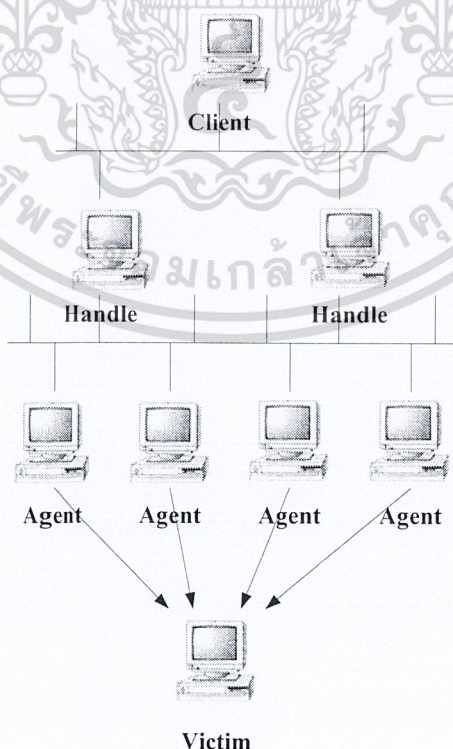
การปิดการให้บริการจะใช้การติดต่อแบบ 3 Tier ซึ่งเมื่อติดต่อจะมีการร้องขอบริการไปยังระบบคอมพิวเตอร์ จะมีการตอบกลับการร้องขอไปยังผู้ร้องขอบริการ และเมื่อผู้ร้องขอได้รับการตอบกลับจะมีการติดต่อกลับไปยังระบบคอมพิวเตอร์นั้น ซึ่งเป็นการเริ่มต้นการติดต่อ ซึ่งแน่นอนว่าระบบคอมพิวเตอร์นั้นจะรอการตอบกลับจากผู้ร้องขอ หรือว่าหมดเวลารอ (Timeout) เราเรียกการติดต่อแบบนี้ว่า “**Half Open**”

ผู้โจมตีที่ใช้การปิดการให้บริการ จะส่งการร้องขอเป็นร้อยเป็นพันครั้งไปยังระบบคอมพิวเตอร์ที่ตกเป็นเหยื่อ (Victim) ซึ่งการเชื่อมการติดต่อจะเกิดขึ้นอย่างมากมาย แน่นอนว่าคอมพิวเตอร์จะมีการรอรับการตอบกลับ ซึ่งทำให้มีการใช้งานทรัพยากรในระบบ (resource) เพิ่มขึ้นเรื่อยๆ ทำให้ประสิทธิภาพของระบบลดต่ำลง จนกระทั่งไม่สามารถรองรับการร้องขอบริการใดๆ ได้

เราเรียกการโจมตีในรูปแบบนี้ว่า “**Asymmetric Attack**” อธิบายได้อย่างง่ายว่า เราสามารถใช้คอมพิวเตอร์ที่มีประสิทธิภาพต่ำ, โมเด็มที่ไม่เร็วมากนัก สามารถที่จะบังคับ หรือควบคุมให้เครื่องคอมพิวเตอร์ที่เร็วกว่า และมีความซับซ้อนมากกว่าได้

ส่วนการใช้ไฟร์วอลล์ (Firewall) จะพยายามป้องกันไม่ให้เกิดการคุกคามเข้ามาในระบบคอมพิวเตอร์ แต่ไม่สามารถใช้ได้กับการปิดการให้บริการ (DoS) เพราะการปิดการให้บริการจะใช้ช่องว่างการสื่อสาร (Bandwidth) เพิ่มขึ้นเรื่อยๆจนกระทั่งระบบคอมพิวเตอร์ไม่สามารถรองรับได้

3.2 พื้นฐานของการโจมตีในรูปแบบการปิดการให้บริการแบบกระจาย



รูปที่ 3-2 แสดงพื้นฐานของการปิดการให้บริการแบบกระจาย

ไคลเอ็นต์ (Client) คือ ผู้ควบคุมการโจมตี **แฮนด์เลอร์ (Handler)** คือ โสสต์ที่มีโปรแกรมพิเศษที่ทำงานบนเครื่อง โดยแต่ละเครื่องสามารถควบคุมเครื่องเอเจนต์ (agents) ได้หลายเครื่อง ส่วน **เอเจนต์ (Agents)** คือ โสสต์ที่มีโปรแกรมพิเศษทำงานบนเครื่อง จะรับผิดชอบการสร้างสตรีมของแพ็กเก็ต ที่ส่งไปยังเครื่องที่ตกเป็นเหยื่อโดยตรง

ส่วนโปรแกรมที่ผู้โจมตีรู้จักกันเป็นอย่างดี คือ Trinoo , TFN , TFN2K และ Stacheldraht และล่าสุดได้มีเพิ่มขึ้นมา คือ TFN3K , Shaft และ Mstream (ปี พ.ศ. 2544) เราจะมาวิเคราะห์แต่ละโปรแกรมโดยละเอียดในลำดับต่อไป

การปิดการให้บริการแบบกระจาย (Distributed Denial Of Services : DDoS)

การทำงานของ การปิดการให้บริการแบบกระจาย (Distributed Denial Of Service : DDoS) มีความแตกต่างจาก การปิดการให้บริการ (Denial Of Service : DoS) เพียงเล็กน้อย

การปิดการให้บริการจะใช้คอมพิวเตอร์เพียงเครื่องเดียว แต่การปิดการให้บริการแบบกระจายจะใช้คอมพิวเตอร์มากกว่านั้น อาจจะเป็นร้อยเป็นพันๆเครื่อง ซึ่งแน่นอนว่า ความรุนแรงจะเพิ่มขึ้นจนทำให้เหยื่อที่ถูกรุกรการโจมตีในรูปแบบนี้ได้รับความเสียหายได้รวดเร็วยิ่งขึ้น

ซึ่งใช้วิธีการแก้ไขแพ็กเก็ต ทีซีพี/ไอพี ที่ใช้งานเพียงเล็กน้อย แต่ทว่าทว่าความรุนแรงได้อย่างมาก เพราะไม่สามารถที่จะป้องกันจากคอมพิวเตอร์จำนวนมากๆได้อย่างแน่นอน

3.3 โหมดในการโจมตี

3.3.1 การใช้งานทรัพยากรที่มีอยู่อย่างจำกัด

3.3.1.1 การเชื่อมต่อทางเครือข่าย

3.3.1.2 การใช้งานทรัพยากรของคุณในการต่อต้านตนเอง

3.3.1.3 การใช้งานช่องว่างการสื่อสารมาก

3.3.1.4 การใช้งานทรัพยากรอื่นๆ

3.3.2 การทำลาย หรือสลับข้อมูลที่ใช้ในการกำหนดคุณลักษณะ

3.3.3 การทำลาย หรือสลับของส่วนประกอบของเครือข่ายทางฟิสิกส์คอล

3.3.1. การใช้งานทรัพยากรที่มีอยู่อย่างจำกัด

คอมพิวเตอร์และเครือข่ายจำเป็นต้องใช้งานทรัพยากรเพื่อปฏิบัติงาน อาทิ เช่น ช่องว่างการสื่อสารเครือข่าย , หน่วยความจำ และ พื้นที่ว่างของดิสก์ , เวลาของซีพียู , โครงสร้างข้อมูล , การเข้าถึงข้อมูลของคอมพิวเตอร์หรือเครือข่าย และส่วนประกอบอื่นๆที่จำเป็น อาทิ เช่น แหล่งจ่ายกระแสไฟฟ้า , การระบายความร้อนด้วยอากาศ หรืออาจจะเป็นน้ำ

3.3.1.1 การเชื่อมต่อทางเครือข่าย

การปิดการให้บริการทุกๆตัวจะมีความทำงานด้วยการต่อต้านการเชื่อมต่อทางเครือข่าย จุดประสงค์เพื่อป้องกัน โสสต์ หรือเครือข่ายไม่สามารถติดต่อกับเครือข่ายอื่นใดได้ ตัวอย่าง คือ การโจมตี

แบบ SYN flood โดยที่ผู้โจมตีจะเริ่มดำเนินการติดต่อไปยังเครื่องของเหยื่อ แต่จะไม่มีมีการติดต่อกลับไปยังเครื่องเหยื่อเพื่อให้การติดต่อไม่สมบูรณ์ ซึ่งทำให้เครื่องเหยื่อต้องสงวนทรัพยากรไว้เพื่อรอการตอบกลับ ซึ่งในลักษณะนี้ เราเรียกว่า “ การเชื่อมต่อแบบ half open ”

3.3.1.2 การใช้งานทรัพยากรของคุณในการต่อต้านตนเอง

ตัวอย่างในกรณีนี้ คือ ผู้โจมตีจะทำการส่งแพ็กเก็ตแบบ ยูดีพี ติดต่อไปยังบริการสะท้อนกลับ จากเครื่องคอมพิวเตอร์เครื่องหนึ่ง ให้เข้าไปใช้บริการของเครื่องอื่น ทำให้เกิดการให้บริการช่องว่างการสื่อสารระหว่างของเครื่องบริการระหว่างกันเอง ยังผลให้เครื่องคอมพิวเตอร์เครื่องอื่นไม่สามารถใช้ช่องว่างการสื่อสารบนเครือข่ายได้เลย

3.3.1.3 การใช้งานช่องว่างการสื่อสารมาก

ผู้โจมตีจะใช้ช่องว่างการสื่อสารที่มีความจำเป็นต้องใช้ในการติดต่อสื่อสารบนเครือข่าย โดยการสร้างแพ็กเก็ตจำนวนมาก และส่งไปยังเครือข่ายคุณโดยตรง เราเรียกแพ็กเก็ตประเภทนี้ว่า “ แพ็กเก็ต ICMP ECHO ” แต่ในความเป็นจริงแล้วมันเป็นอะไรก็ได้ ซึ่งผู้โจมตีไม่จำเป็นต้องปฏิบัติการจากเครื่องโดยตรง พวกเขาอาจจะใช้การติดต่อสัมพันธ์ หรือ การทำงานร่วมกันจากเครื่องอื่นๆจากเครือข่ายใดๆเพื่อให้ได้ผลเช่นเดียวกัน

3.3.1.4 การใช้งานทรัพยากรอื่นๆ

นอกเหนือจากช่องว่างการติดต่อสื่อสารแล้ว ผู้โจมตีอาจจะใช้ทรัพยากรที่ระบบจำเป็นต้องการใช้งาน ตัวอย่าง เช่น ในหลายระบบจำกัดจำนวนโครงสร้างข้อมูลในแต่ละกระบวนการ ผู้โจมตีอาจจะใช้โปรแกรม หรือสคริปต์ต่างๆในการสร้าง และก๊อปปี้ตัวเองไปเรื่อยๆ ทำให้จำนวนงานของระบบเพิ่มขึ้นเรื่อยๆ จนไม่สามารถรองรับงานใดๆได้อีก

ซึ่งในระบบปฏิบัติการใหม่ๆจะมีการจำกัดโควตาการใช้งานทรัพยากรเพื่อป้องกันปัญหาประเภทนี้ แต่ไม่ได้มีในทุกๆระบบปฏิบัติการ หรืออาจจะใช้ทรัพยากรของพื้นที่ดิสก์ ไม่ว่าจะเป็นการสร้างเมล์, หรือใช้พื้นที่ที่มีการแชร์ดิสก์ไว้

ดังนั้นในหลายๆเว็บไซต์ที่จะมีการจำกัดการกำหนด “ ล็อกเอาต์ ” ถ้าล็อกอินผิดพลาดเกินกว่าที่กำหนด หรือเพื่อจุดประสงค์ที่ต้องการพาสเวิร์ดของรูต (Root) หรือผู้ดูแลระบบ (Administrator) ซึ่งจะเกิดความเสียหายให้กับระบบได้มากมาย

ทรัพยากรอื่นๆ ที่มีการจำกัดการใช้งานใดๆ รวมไปถึง พรินเตอร์ , อุปกรณ์ประเภทเทป , การเชื่อมต่อกับเครือข่าย และทรัพยากรใดๆที่มีการจำกัดการใช้งานไว้ในระบบ

3.3.2 การทำลาย หรือสลับข้อมูลที่ใช้ในการกำหนดคุณลักษณะ

การกำหนดไฟล์คุณลักษณะที่เกี่ยวกับระบบคอมพิวเตอร์เป็นสิ่งจำเป็น แต่ถ้าเกิดมีการเปลี่ยนแปลงไฟล์คุณลักษณะใดๆเกิดขึ้น อาจจะนำมาซึ่งความเสียหาย ในจุดนี้ก็เป็นอีกจุดหนึ่งที่ผู้โจมตีนำมาใช้ในการโจมตีระบบ

ตัวอย่าง เช่น ถ้าผู้โจมตีได้ทำการเปลี่ยนแปลงข้อมูลในการหาเส้นทางของเราเตอร์ (Router) ของคุณได้แล้วนั้น ย่อมส่งผลให้ไม่สามารถใช้งานเครือข่ายได้เลย หรือถ้าผู้โจมตีสามารถเปลี่ยนแปลงค่ารีจิสทรี (registry) ของ วินโดวส์เอ็นที ได้แล้วแน่นอนว่าหน้าที่การทำงานย่อมเปลี่ยนแปลงตาม

3.3.3 การทำลาย หรือสลับของส่วนประกอบของเครือข่ายทางฟิสิกส์คอล

จุดประสงค์หลักของการพิจารณาในที่นี้คือ ความปลอดภัยของอุปกรณ์ด้านฮาร์ดแวร์ ซึ่งสามารถช่วยป้องกันผู้โจมตี หรือไม่มีสิทธิใดๆเข้ามาในระบบคอมพิวเตอร์ , เราเตอร์ , อุปกรณ์เชื่อมต่อเครือข่าย (Network wiring closet) , การจัดแบ่งเครือข่ายหลักๆ (Network backbone segments) , สถานีจ่ายกระแสไฟฟ้าและความเย็น รวมถึงอุปกรณ์ส่วนอื่นๆที่ประกอบอยู่ในเครือข่าย

ความปลอดภัยของฮาร์ดแวร์ เหล่านี้จะช่วยในการป้องกันการโจมตีในรูปแบบการปิดการให้บริการได้ ส่วนข้อมูลในการป้องกันเครือข่าย แนะนำให้ติดต่อบริษัทที่เราติดต่อเพื่อช่วยกำหนดค่าคุณลักษณะใดๆ

3.4 ปัญหา ,แนวทางป้องกัน เพื่อช่วยลดผลกระทบจากการโจมตี

3.4.1.ปัญหาการ Spoofing

3.4.2.ปัญหาการส่งต่อแพ็กเก็ตแบบบรอดคาสต์ (Broadcast)

3.4.3.ปัญหาการลดประสิทธิภาพของระบบเมื่อถูกโจมตี

3.4.4.ปัญหาความปลอดภัยของเครื่อง

3.4.1.ปัญหาการ Spoofing

ผู้โจมตีมักจะใช้วิธีการซ่อนแหล่งที่อยู่ของไอพีที่ใช้ในการติดต่อสื่อสารระหว่างการโจมตี ทำให้ไม่สามารถติดตามเพื่อหาแหล่งที่มา หรือเครื่องที่ใช้ในการ โจมตีได้ และบางครั้งก็ใช้เครื่องคอมพิวเตอร์ของผู้ที่ไม่รู้จักป้องกัน ซึ่งตกเป็นเหยื่อในการใช้เพื่อการโจมตี

แนวทางแก้ไข

3.4.1.1. ผู้ใช้งาน และ ผู้ให้บริการอินเทอร์เน็ต (ISPs : Internet Service Providers) จะให้ผู้ให้บริการสามารถเข้าใจงาน หรือใช้บริการจะต้องมีแหล่งที่มาของผู้ใช้บริการ แน่แน่นอนว่าไม่สามารถป้องกันการซ่อนไอพีได้ แต่อย่างน้อยก็ช่วยในการติดตามเพื่อหาแหล่งที่มาของการโจมตีได้ และป้องกันไม่ให้มีการใช้งานแบบ “Unroutable Address ” (รายละเอียดศึกษาได้จาก RFC 1918) ส่งออกจากเครือข่ายเหล่านั้น หรือรู้จักกันในชื่อว่า “Egress Filtering” จุดประสงค์เพื่อไม่ให้เครื่อง หรือเครือข่ายของเราเป็นที่ใช้ในการโจมตีเครื่อง หรือเครือข่ายอื่นๆ และยอมรับเฉพาะการเข้าใช้งานแบบแสดงตน (Authorize) รู้จักในชื่อว่า “Ingress Filtering ” ซึ่งสามารถกำหนดได้จากผู้ให้บริการอินเทอร์เน็ต

3.4.1.2. การหมุนโทรศัพท์ของผู้ใช้งาน นี่ก็เป็นอีกแนวทางหนึ่งที่ใช้ในการโจมตี โดยจะต้องหาเครื่องมือเพื่อใช้ในการป้องกัน การซ่อนการหมุนโทรศัพท์เพื่อใช้งานของผู้ใช้

3.4.1.3. ปฏิเสธการติดต่อ จากแหล่งที่อยู่ไอพีแบบไพรเวต (Privated) และ Reserved

- 10.0.0.0/8	- RFC 1918 Private Network
- 127.0.0.0/8	- Loopback
- 169.254.0.0/16	- Link Local Networks
- 172.16.0.0/12	- RFC 1918 Private Network
- 192.0.2.0/24	- TEST-NET
- 192.168.0.0/16	- RFC 1918 Private Network
- 224.0.0.0/4	- Class D Multicast
- 240.0.0.0/5	- Class E Reserved
- 248.0.0.0/5	- Unallocated
- 255.255.255.255/32	- Broadcast

3.4.2 ปัญหาการส่งต่อแพ็กเก็ตแบบบรอดคาสต์ (Broadcast)

เกิดจากการที่ผู้โจมตี ส่งแพ็กเก็ตจำนวนมาก (Flooding Packets) จากเครื่องที่ใช้ในการโจมตีไปยังเครื่องที่ตกเป็นเหยื่อ การโจมตีรูปแบบนี้ เรียกว่า “Smurf Attack”

แนวทางแก้ไข

3.4.2.1 ในบางระบบจะมีการกำหนดให้สามารถทำการส่งแพ็กเก็ตแบบบรอดคาสต์ และ มัลติคาสต์ เพื่อให้สามารถส่งแพ็กเก็ตต่อไปได้ ก็ให้ปิดการทำงานในรูปแบบนี้อาทิ เช่น แพ็กเก็ต แบบ ICMP Echo Reply

3.4.2.2 กำหนดการทำงานของฮาร์ดแวร์ เช่น เราเตอร์เพื่อปิดการส่งต่อไอพีแพ็กเก็ตแบบบรอดคาสต์โดยตรง (รายละเอียดใน RFC 2644)

3.4.2.3 ปิดการบริการการตอบกลับ และบริการที่ไม่จำเป็นออกไป

3.4.3 ปัญหาการลดประสิทธิภาพของระบบเมื่อถูกโจมตี

บางครั้งในระบบไม่มีการตอบสนอง หรือรู้จักป้องกัน และเมื่อถูกโจมตีแล้วก็ไม่รู้ว่าจะทำอย่างไร เพื่อตอบสนองต่อการ โจมตีที่เกิดขึ้น

แนวทางแก้ไข

3.4.3.1 ต้องมีการกำหนดนโยบาย และทีมงานที่จะรับผิดชอบต่อการทำงานด้านความปลอดภัย

3.4.3.2 ผู้ให้บริการอินเทอร์เน็ต จะต้องมีการกำหนดวิธีการตอบสนองต่อการ โจมตี เมื่อมีการตรวจพบว่าการ โจมตีเกิดขึ้นกับระบบอย่างรวดเร็ว

3.4.3.3 ต้องมีการรายงานเกี่ยวกับการ โจมตี เพื่อหาการป้องกัน

3.4.4 ปัญหาความปลอดภัยของเครื่อง

ในหลายระบบถูกเข้าควบคุม และตกเป็นเหยื่อสำหรับใช้โจมตีการปิดการให้บริการแบบกระจาย ซึ่งทำให้ระบบตกเป็นเครื่องมือที่จะนำมาใช้เป็นเครือข่ายในการโจมตีได้

แนวทางแก้ไข

3.4.4.1 ให้ผู้ใช้งานตรวจสอบโปรแกรมที่ติดตั้งอยู่ในระบบ และเมื่อพบโปรแกรมการ โจมตีการปิดการให้บริการแบบกระจาย ก็ทำการคืนค่าสถานะเดิมที่ดีก่อนหน้านี้

3.4.4.2 ให้ติดตั้งไฟร์วอลล์ (Firewall) เพื่อกำหนด และจำกัดการติดต่อสื่อสารอย่างเหมาะสม

3.4.4.3 ทุกเครื่องในระบบ ,เราเตอร์ (Router) หรืออุปกรณ์ที่ติดต่อกับอินเทอร์เน็ต ให้ตรวจสอบการป้องกันความปลอดภัย และมีการปรับปรุงดูแลอยู่เสมอ

3.4.4.4 กำหนดรายชื่อรูปแบบการโจมตี และหาทางกำหนด Priorities โดยผู้ดูแลระบบ

3.4.4.5 ปิดบริการที่ไม่จำเป็น และจำกัดการเข้าใช้งานบริการ

3.4.4.6 ขอความร่วมมือทั้งผู้ใช้งาน และผู้ให้บริการเพื่อหาช่องโหว่ และการกำหนดเพื่อความปลอดภัยร่วมกัน

3.4.4.7 กำหนดระดับการรักษาความปลอดภัยของโปรแกรมที่เข้าใช้งานให้สูงขึ้น

3.4.4.8 ให้ผู้ดูแลระบบ ติดตั้งไฟร์วอลล์ (Firewall) , ระบบตรวจสอบการโจมตี , โปรแกรมตรวจสอบไวรัส เพื่อตรวจสอบการทำงานของระบบ

3.4.4.9 มีการทดสอบ และเพิ่มทักษะของผู้ดูแลระบบ และผู้ควบคุมเครือข่าย

3.5 โปรแกรมที่ใช้โจมตีเพื่อการปิดการให้บริการแบบกระจาย

3.5.1 Trinoo (Wintrinoo)

3.5.2 TFN (TFN2K, TFN3K)

3.5.3 Mstream

3.5.4 Stacheldraht

3.5.1 Trinoo (Wintrinoo)

3.5.1.1 แนะนำ

โปรแกรม Trinoo หรือ Trin00 เป็นโปรแกรมซึ่งทำงานในรูปแบบ Master / Slave ซึ่งเป็นเครื่องมือที่พัฒนาการโจมตีการปิดการให้บริการแบบกระจายบนเครือข่ายคอมพิวเตอร์ เดิมพบในระบบ Solaris 2.x ทำงานกับแพ็กเก็ตประเภท UDP, มีการจำกัดการใช้งานจากคำสั่งระยะไกล และคอยดักจับ (Sniffer) ไฟล์พาสเวิร์ด และคั่นค่าล็อกไฟล์

จากการสังเกตพบว่า การโจมตีแบบนี้ จะมีการติดตั้งโปรแกรม และมีการตรวจสอบเครื่องมือ , ล็อกไฟล์ที่ใช้งาน

การแก้ไขโค้ด (Source Code) ของโปรแกรม จะทำให้รูปแบบของรายละเอียดในการวิเคราะห์นี้เปลี่ยนแปลงไปจากเดิม อาทิ เช่น มีการเปลี่ยนแปลงของพรมท (Prompt) , พาสเวิร์ด (password) , คำสั่ง (Command) , หมายเลขพอร์ต ที่ซีพี/ยูดีพี , วิธีการที่ช่วยในการโจมตี , ลายเซ็น (Signature) และคุณลักษณะอื่นๆ

ซึ่งทำงานบนระบบ Solaris 2.5.1 และ Red Hat Linux 6.0 เครือข่าย Trinoo จะมีการติดตั้งบนเครื่องของระบบอินเทอร์เน็ตซึ่งมีการใช้งาน เพื่อแสวงหาผลประโยชน์ทางการค้า การเข้าใช้งานโดยการติดตั้งโปรแกรมประเภท Back Door ซึ่งเป็น เดมอน (Daemon) ของ Trinoo

จากผลกระทบในช่วงเดือน ตุลาคม ปี ค.ศ. 1999 มีระบบอย่างน้อย 227 ระบบ ซึ่งใน 114 ระบบ เป็นเว็บไซต์ให้บริการอินเทอร์เน็ต โดยมีการส่งแพ็กเก็ตแบบ Flooding ไปยังระบบคอมพิวเตอร์ของ มหาวิทยาลัยมินนิโซต้า ทำให้ไม่สามารถใช้งานเครือข่าย และหยุดการทำงานไปกว่าสองวัน

แนวทางการโจมตี

1. มีการขโมยบัญชีรายชื่อด้วยวิธีการ ใช้เครื่องมือการค้นหา, เครื่องมือใช้ในการ โจมตี, Root kits และการดักจับ (Sniffer) , โปรแกรม Trinoo มาสเตอร์ และ Trinoo เดมอน
2. ตรวจสอบการป้องกันเครือข่ายของเป้าหมาย ว่ามีการให้บริการ อาทิ เช่น Wu-Ftpd , บริการ RPC (Remote Procedure Call) สำหรับ cmd , statd , ttdbserver , amd ฯลฯ ที่ให้บริการของ Solaris 2.x และ Linux
3. แสดงรายชื่อระบบที่ไม่มีการป้องกัน โดยใช้สคริปต์, ติดตั้งคำสั่ง shell ที่ทำงานโดยบัญชีรายชื่อของรูท การทำงานของพอร์ต ที่ซีพี (ปกติ เป็น 1524/tcp เป็นพอร์ตให้บริการ “ingreslock”) เพื่อยืนยันการติดต่อผ่านพอร์ตผลทำให้ได้รายชื่อของระบบนี้ พร้อมที่จะติดตั้ง Back Door , sniffers หรือ โปรแกรม Trinoo มาสเตอร์ หรือ Trinoo เดมอน
4. ทำให้พร้อมที่จะออกแบบ โครงสร้างของเครือข่าย Trinoo และ ทำการ pre-compile ของไฟล์ Trinoo เดมอน และส่งไปยังบัญชีรายชื่อบนอินเทอร์เน็ต
5. สคริปต์นี้จะทำงานบนระบบ และเริ่มกระบวนการติดตั้ง ,ทำงานแต่ละอย่างแบบ มัลติทาสกิง (Multitasking) สคริปต์ที่ใช้ในนี้คือ “netcat” (“nc”) คอยดักฟังพอร์ต 1524/tcp หรืออีกวิธีคือใช้ “c” จะมีการใช้ xterm


```
cd /var/adm/.1
PATH = .: $ PATH
export PATH
xterm 1>/dev/null 2>&1
```
6. ออปชัน อีกอย่าง คือ ใช้รูตคิต (root kit) ที่ติดตั้งในระบบเพื่อซ่อนการทำงานของโปรแกรม , ไฟล์การเชื่อมต่อของเครือข่าย เป็นส่วนสำคัญในการควบคุมการทำงาน

3.5.1.2 เครือข่าย

เครือข่าย : attacker(s) → master(s) → daemon(s) → victime(s)

โดยเครื่อง Trinoo มาสเตอร์ (“master.c”) และ Trinoo เดมอน (“nc.c”) โดยผู้โจมตีจะควบคุมเครื่อง Trinoo มาสเตอร์ เหล่านั้น ซึ่งแต่ละเครื่องจะควบคุม Trinoo เดมอนได้หลายเครื่อง ในโค้ดจะเป็น “Bcast” หรือ โฮสต์ “broadcast” และ Trinoo เดมอนจะใช้การติดต่อโดยใช้แพ็กเก็ต หรือ โจมตีไปยังระบบของระบบที่ตกเป็นเหยื่อ ติดต่อไปยังเครื่อง Trinoo มาสเตอร์ จะใช้ผ่าน เทลเน็ต และมีการตรวจสอบพาสเวิร์ดเพื่อป้องกันผู้ดูแลระบบ หรือ กลุ่มแฮกเกอร์อื่นๆ เข้าควบคุมเครือข่าย Trinoo แทน

3.5.1.3 การติดต่อสื่อสารผ่านพอร์ต

Attacker to Master (s)	:	27665/tcp
Master to daemon (s)	:	27444/udp
Daemon to Master (s)	:	31335/udp

การควบคุมระยะไกลไปยังเครื่อง Trinoo มาสเตอร์ จะใช้พอร์ต 27665/tcp หลังจากนั้นจะมีการตรวจสอบพาสเวิร์ด (ปกติเป็น “betaalmostdone”) ถ้าตรวจสอบสิทธิ์ไม่ถูกต้องจะมีการติดต่อกลับไปยังโฮสต์ของที่อยู่ไอพีแอดเดรส (IP address) ในเวอร์ชันนี้ยังมีบั๊กอยู่ แต่ก็มีคำตอบกลับไปยังจริง การติดต่อจากเครื่อง Trinoo มาสเตอร์ กลับไปยังเครื่อง Trinoo เดมอนใช้พอร์ต 27444/udp โดยใช้

Arg1 password arg2

การติดต่อจากเครื่อง Trinoo เดมอน ไปยังเครื่อง Trinoo มาสเตอร์ใช้พอร์ต 31335/udp

เมื่อเครื่อง Trinoo เดมอนทำงานจะส่งค่า “*HELLO*” ไปยังเครื่อง Trinoo มาสเตอร์ซึ่งจะเก็บรายชื่อเครื่อง Trinoo เดมอน ที่ทำงานเพื่อใช้ควบคุม (การดักจับแพ็กเก็ตใช้ “sniffit”)

ถ้าเครื่อง Trinoo มาสเตอร์ส่งคำสั่ง “png” ไปยังเครื่อง Trinoo เดมอน ผ่านพอร์ต 27444/udp เครื่อง Trinoo เดมอน จะตอบกลับด้วยค่า “PONG” ผ่านพอร์ต 31335/udp

3.5.1.4 การป้องกันด้วยพาสเวิร์ด

“144adsl”	พาสเวิร์ดเครื่อง Trinoo เดมอน
“gOrave”	เครื่อง Trinoo มาสเตอร์ตอนเริ่มต้นการติดต่อผ่านพอร์ต “??”
“betaalmostdone”	พาสเวิร์ดระหว่างการเชื่อมต่อเครื่อง Trinoo มาสเตอร์
“killme”	พาสเวิร์ด Trinoo มาสเตอร์เพื่อควบคุมคำสั่ง “mdie”

3.5.1.5 คำสั่ง

3.5.1.5.1 คำสั่งเครื่อง Trinoo master

die	ปิดการทำงานโปรแกรมเครื่อง Trinoo มาสเตอร์
quit	ออกจากเครื่อง Trinoo มาสเตอร์
mtimer N	กำหนดเวลา N วินาที มีค่าระหว่าง 1-1999 วินาที ถ้า N<1 จะมีค่าเป็น 300 วินาที ถ้า N > 2000 จะมีค่าเป็น 500
dos IP	คำสั่งให้โจมตีการปิดการให้บริการเฉพาะไอพี ที่กำหนด (คำสั่ง “aaa 144adsl IP”) ซึ่งส่งไปยังเครื่องโฮสต์ Bcast (คือ เครื่อง Trinoo เดมอน)
mdie pass	ถ้าพาสเวิร์ด (ปกติเป็น “die 144adsl”) ถูกต้อง จะสั่งให้หยุดการทำงานโฮสต์ Bcast
mping	ส่งคำสั่ง PING (“png 144adsl”) ไปยังเครื่องโฮสต์ Bcast ที่ทำงาน

mdos <ip1:ip2:ip3>	สั่งให้ทำงานการปิดการให้บริการแบบ Multiple ไปยังโฮสต์ Bcast
info	แสดงข้อมูลเวอร์ชัน และข้อมูลการคอมไพล์
msize	กำหนดขนาดของบัฟเฟอร์สำหรับแพ็กเก็ตระหว่างการโจมตีการปิดการให้บริการ
nslookup host	เพื่อดูการทำงานของเครื่อง Trinoo มาสเตอร์
killdead	พยายามกำจัดเครื่อง Trinoo เดมอน ที่ทำงานอยู่ โดยจะตอบกลับว่า “*HELLO*” เพื่อจะได้จัดเตรียมการเริ่มต้นการทำงานใหม่
usebackup	สลับการทำงานไปยังไฟล์แบ็กอัพ เครื่องโฮสต์ Bcast หลังจากใช้คำสั่ง “killdead”
bcast	แสดงรายชื่อเครื่องโฮสต์ Bcast ที่ทำงานอยู่
help [cmd]	แสดง, อธิบายการทำงานของคำสั่ง “cmd” ที่ต้องการ
mstop	พยายามยุติการทำงานการโจมตีการปิดการให้บริการ

3.5.1.5.2 คำสั่งเครื่อง Trinoo เดมอน

aaa pass IP	ส่งแพ็กเก็ตยูดีพีอย่างสุ่ม (พอร์ต 0-65534) ไปยังไอพีที่ปิดการให้บริการเป็นระยะเวลา 1-1999 วินาที (ค่าปกติเป็น 120 วินาที) กำหนดได้จากคำสั่ง “bbb” และกำหนดขนาดแพ็กเก็ตได้จากคำสั่ง “rsz” (ค่าขนาดปกติเป็น 1,000 ไบต์)
bbb pass N	กำหนดเวลาในการโจมตีการปิดการให้บริการ (วินาที)
shi pass	ส่งคำ “*HELLO*” กลับไปยังเครื่อง Trinoo มาสเตอร์ผ่านพอร์ต 31335/udp
png pass	ส่งคำ “PONG” กลับไปยังเครื่อง Trinoo มาสเตอร์ผ่านพอร์ต 31335/udp
dle pass	ปิดการทำงานของเครื่อง Trinoo เดมอน
rsz N	กำหนดขนาดของบัฟเฟอร์สำหรับการโจมตีเพื่อปิดการให้บริการขนาด N ไบต์ (ใช้ฟังก์ชัน malloc() ในการกำหนดขนาด แต่ในขณะโจมตี ไม่สามารถเปลี่ยนแปลงขนาดของบัฟเฟอร์ได้)
xyz pass 123:ip1:ip2:ip3	คล้ายคำสั่ง “aaa” แต่ส่งไปยังไอพี แบบ Multiple

3.5.1.6 ลักษณะการโจมตี

วิธีการติดตั้งโปรแกรม Trinoo เดมอน ใช้เวลาเพียงเล็กน้อย บางระบบจะอยู่ใน

****/usr/sbin/rpc.listen

โปรแกรม Trinoo มาสเตอร์จะสร้างไฟล์ (ปกติชื่อว่าไฟล์ “...”) ซึ่งมีรายชื่อของเครื่องโฮสต์ Bcast อยู่ ถ้ามีการใช้คำสั่ง “killdead” และคำสั่ง “shi” จะส่งรายชื่อเครื่อง Trinoo เดมอนในไฟล์ “...” มี

การส่งแพ็กเก็ตส่งคำ “*HELLO*” กลับไปยังเครื่อง Trinoo มาสเตอร์เพื่อสร้างเป็นไฟล์แบ็กอัพ (ปกติชื่อว่า “...-b”) ทุกครั้งที่เครื่อง Trinoo เดมอนจะตอบกลับด้วย คำ “*HELLO*” กลับมา

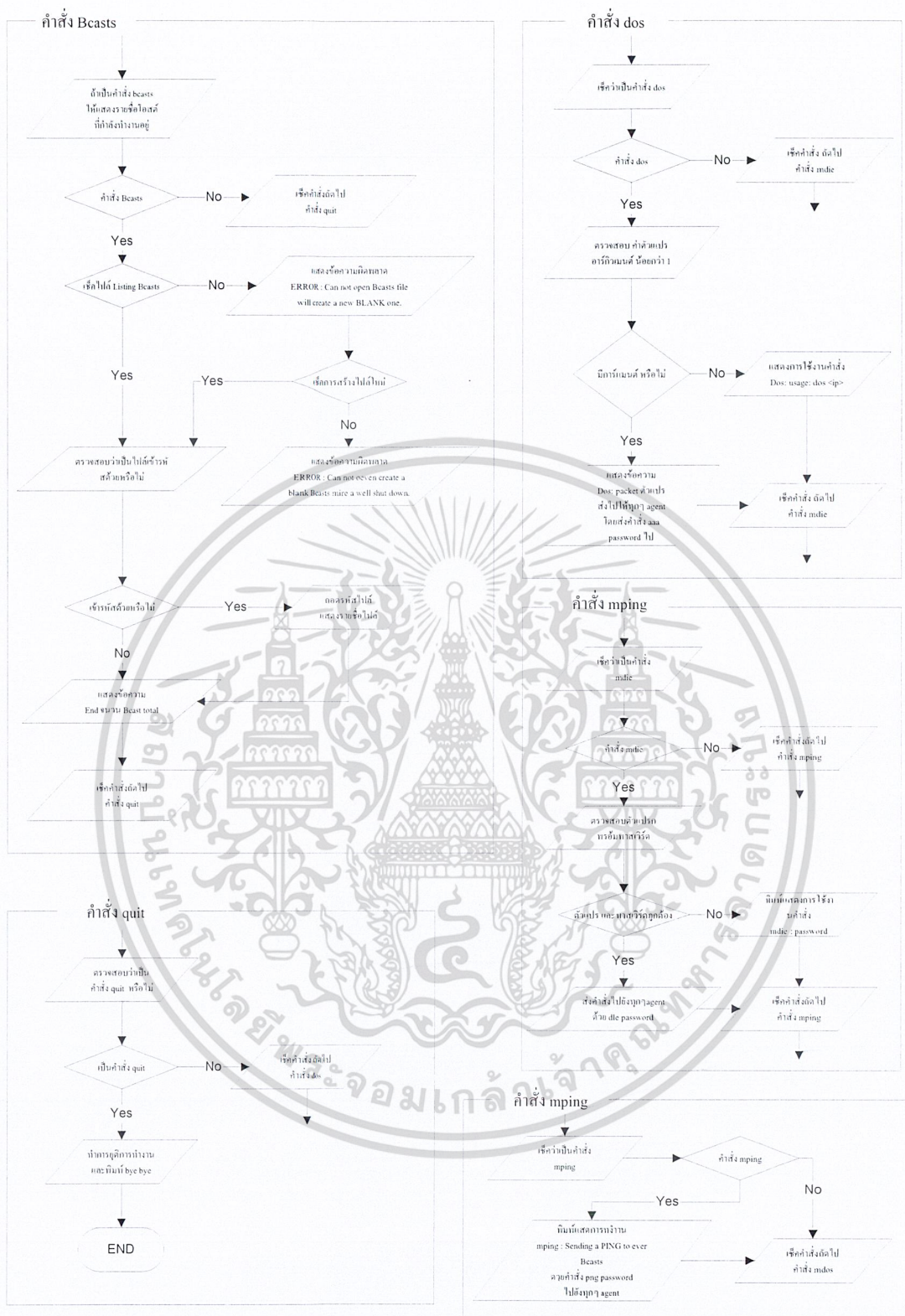
3.5.1.7 การป้องกัน

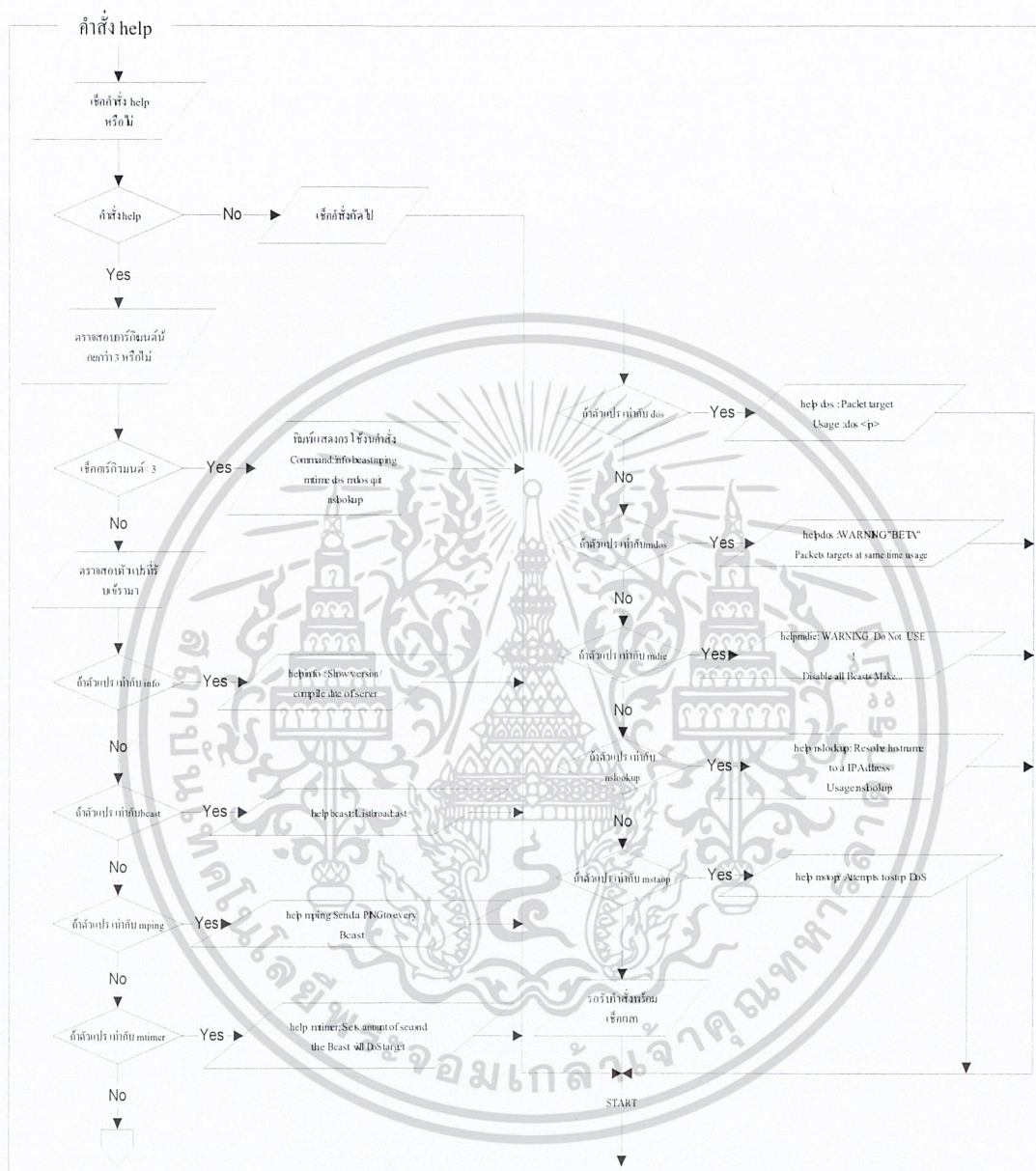
สิ่งแรกที่ต้องทำ คือ การป้องกันการโจมตี หรือว่าตกเป็นเครื่องมือของ Trinoo มาสเตอร์/เดมอน ในความเป็นจริงแล้ว ในทุกระบบจะมี patch ที่คอยอุดช่องโหว่ของระบบ, การป้องกันความปลอดภัย, การดูการทำงาน (monitor), ระบบต่อต้านการ โจมตี (IDS: Intrusion Detection System) และ กำแพงไฟ (firewall) ที่คอยตรวจสอบและปฏิเสธแพ็กเก็ตที่ผิดปกติ

โปรแกรมมีการใช้งานพอร์ตสูงของยูทิลิตี้ทั้งการสื่อสารและการโจมตี เป็นการยากที่จะแยกความแตกต่างกัน ระหว่างการใช้งานปกติ กับการใช้งานเพื่อการโจมตี

วิธีที่ง่ายกว่า คือการใช้การตรวจสอบหาโปรแกรม Trinoo มาสเตอร์/เดมอน ซึ่งอาจจะใช้วิธีการดูการทำงานของแพ็กเก็ตที่ใช้งานร่วมกันของอีเทอร์เน็ต (Ethernet) และดูการติดต่อระหว่างเครื่อง Trinoo มาสเตอร์ และเครื่อง Trinoo เดมอนซึ่งการทำงานจะเกิดขึ้นในช่วงของการ โจมตีเท่านั้นเอง







3.5.2 TFN (TFN2K, TFN3K)

3.5.2.1 แนะนำ

TFN (Tribe Flood Network) จะมีการทำงานคล้ายกับ Trinoo จะมีโครงสร้างโปรแกรมไคลเอนต์ และเดมอนมีการปรับปรุงให้สามารถใช้ ICMP Flood, SYN Flood, UDP Flood และการโจมตีแบบ Smurf ซึ่งมีการทำงานในพอร์ตที่ซีพี TFN เดมอน เดิมพบในระบบ Solaris 2.x ซึ่งมีการใช้ buffer overrun ในบริการ RPC

ตัวเดมอนจะมีการทำงานแบบเก่าด้วยการดักจับระยะไกลหรือ การควบคุมคำสั่งเซิร์ฟเวอร์ระยะไกลซึ่งการทำการแก้ไขข้อผิดพลาดจะทำให้การวิเคราะห์นั้นเปลี่ยนแปลงไป

ถ้าเดมอนและไคลเอนต์จะคอมไพล์และทำงานบน Red Hat 6.0

3.5.2.2 เครือข่าย

attacker (s) → client (s) → daemon(s) → victim(s)

เครือข่าย TFN จะมีโปรแกรมไคลเอนต์ ("tribe.c") และ tribe เดมอน ("td.c")

3.5.2.3 การติดต่อสื่อสาร

การควบคุมระยะไกลของเครือข่าย TFN จะใช้บรรทัดคำสั่งให้โปรแกรมไคลเอนต์ซึ่งใช้วิธีการเชื่อมต่อได้หลายแบบ (อาทิเช่น remote shell bound ของ พอร์ตที่ซีพี, เซิร์ฟเวอร์ระยะไกล ไคลเอนต์/เซิร์ฟเวอร์ บนพื้นฐานยูดีพี, เซิร์ฟเวอร์ระยะไกล ไคลเอนต์/เซิร์ฟเวอร์บน ICMP เช่น LOKI , การเข้าถึง terminal แบบ SSH, หรือปกติใช้ "telnet" ติดต่อพอร์ตที่ซีพี)

ไม่ต้องการพาสเวิร์ดในการสั่งให้ไคลเอนต์ทำงาน ดังนั้นมันจำเป็นที่จะต้องมียาของเดมอนเก็บไว้ในไฟล์ "iplist"

การติดต่อสื่อสารจาก TFN ไคลเอนต์ไปยังเดมอนจะทำงานด้วยแพ็กเก็ต ICMP_ECHOREPLY ซึ่งไม่จำเป็นต้องใช้การติดต่อโดยพื้นฐานของทีซีพี หรือ ยูดีพี ระหว่างไคลเอนต์ กับ เดมอนตลอดเวลาทางเครือข่ายจะใช้เครื่องในการตรวจหาซึ่งจะไม่สามารถตรวจหรือแสดงข้อมูลที่ใส่แพ็กเก็ต ICMP หรือ วิเคราะห์ฟิลด์ของ ICMP ได้ ดังนั้นจึงเป็นยากที่จะตรวจสอบจากการติดต่อสื่อสารระหว่างไคลเอนต์ กับ เดมอน

3.5.2.4 การป้องกันพาสเวิร์ด

ขณะที่ไคลเอนต์ที่ไม่มีการป้องกันด้วยพาสเวิร์ด โดยถ้าส่งไปยังเดมอนจะถูกส่งไปในรูปแบบของจำนวนเลขไบนารีขนาด 16 บิต ในฟิลด์ id ของแพ็กเก็ต ICMP_ECHOREPLY (โดยจะเป็นเลขลำดับค่าคงที่ 0x0000, ซึ่งเสมือนเป็นการตอบสนองการเริ่มแพ็กเก็ตถูกส่งออกไปด้วยคำสั่ง "ping")

ค่าต่างๆจะถูกกำหนดหรือเปลี่ยนแปลงเพื่อรันโปรแกรม โดยกำหนดไว้ในไฟล์ "config.h" ซึ่งเพื่อป้องกันไม่ให้คนอื่นที่รู้จักค่าต่างๆที่กำหนดไว้ จึงมีการเปลี่ยนแปลงตำแหน่งเหล่านั้นได้เอง

3.5.2.5 คำสั่ง

ถ้าคุณเรียกการทำงานของ TFN โดยไม่ได้ใส่ตัวเลือก หรือ อาร์กิวเมนต์ใดๆ ตัว TFN จะแสดงหน้าจอช่วยเหลือการใช้คำสั่งเองโดยจะมี

usage: ./tfn [ip] [port]

ประกอบไปด้วยรายชื่อของโฮสต์ที่พร้อมจะทำงานโดย

-1 จะเป็นชนิดของ spoofmask

-2 กำหนดขนาดแพ็กเก็ต

0 สำหรับ หยุค/สถานะ

1 สำหรับ udp

2 สำหรับ syn

3 สำหรับ icmp

4 สำหรับ bind a rootshell

5 สำหรับ smurf

ip คือ เป้าหมาย

ips เป็นการกระจาย

[ip] เป้าหมายของไอพีที่สามารถใช้กันด้วย @ สำหรับต้องการมากกว่าหนึ่งไอพีได้

[port] จะกำหนดหาร syn flood หรือ ถ้าเป็น 0 เป็นการสุ่ม (random)

3.5.2.6 ลักษณะการโจมตี

คล้ายกันกับ Trinoo ซึ่งลักษณะวิธีการที่ใช้ในการติดตั้ง โคลเอนด์/เดมอน จะเหมือนกันกับการติดตั้งโปรแกรมบนระบบปฏิบัติการยูนิกซ์ด้วยซอฟต์แวร์มาตรฐานในการติดตั้งโปรแกรมและไฟล์ โดยที่ โคลเอนด์ และ เดมอน จะต้องทำงานโดยรู้ตัวทั้งคู่จะเป็นซ็อกเก็ต (socket) AF_INET ในโหมด SOCK_RAW

โปรแกรมโคลเอนด์ต้องการไฟล์ "iplist" โดยปัจจุบันการติดตั้ง TFN เดมอนจะมีการเพิ่มสตริงเพื่อบอกว่ามีการใช้การเข้ารหัส (encryption) แบบ Blowfish ไว้ในไฟล์ iplist ด้วย ทำให้การที่จะตรวจสอบเดมอนต้องเพิ่มภาระหนักขึ้นไปอีก โดย TFN โคลเอนด์จะส่งคำสั่งไปยังเดมอนโดยใช้แพ็กเก็ต ICMP_ECHOREPLY แทนการใช้แพ็กเก็ต ICMP_ECHO ทั้งนี้เพื่อเป็นการป้องกันเคอร์เนล (kernel) บนระบบที่เดมอนอยู่ตอบกลับด้วยแพ็กเก็ต ICMP_ECHOREPLY และ เดมอนจะตอบสนองกลับไปยังโคลเอนด์โดยใช้แพ็กเก็ต ICMP_ECHOREPLY

โดยในฟิลด์ id ของ ICMP_ECHOREPLY จะประกอบด้วยค่าตั้ง (ค่า 16 บิต จะแปลงเป็นไบนารีของเครือข่ายซึ่งเช็ค โดยใช้ htons() ในโค้ด) และหลายๆ อาร์กิวเมนต์จะให้อยู่ในรูปของแอสกี ในฟิลด์ข้อมูลของแพ็กเก็ต

3.5.2.7 การป้องกัน

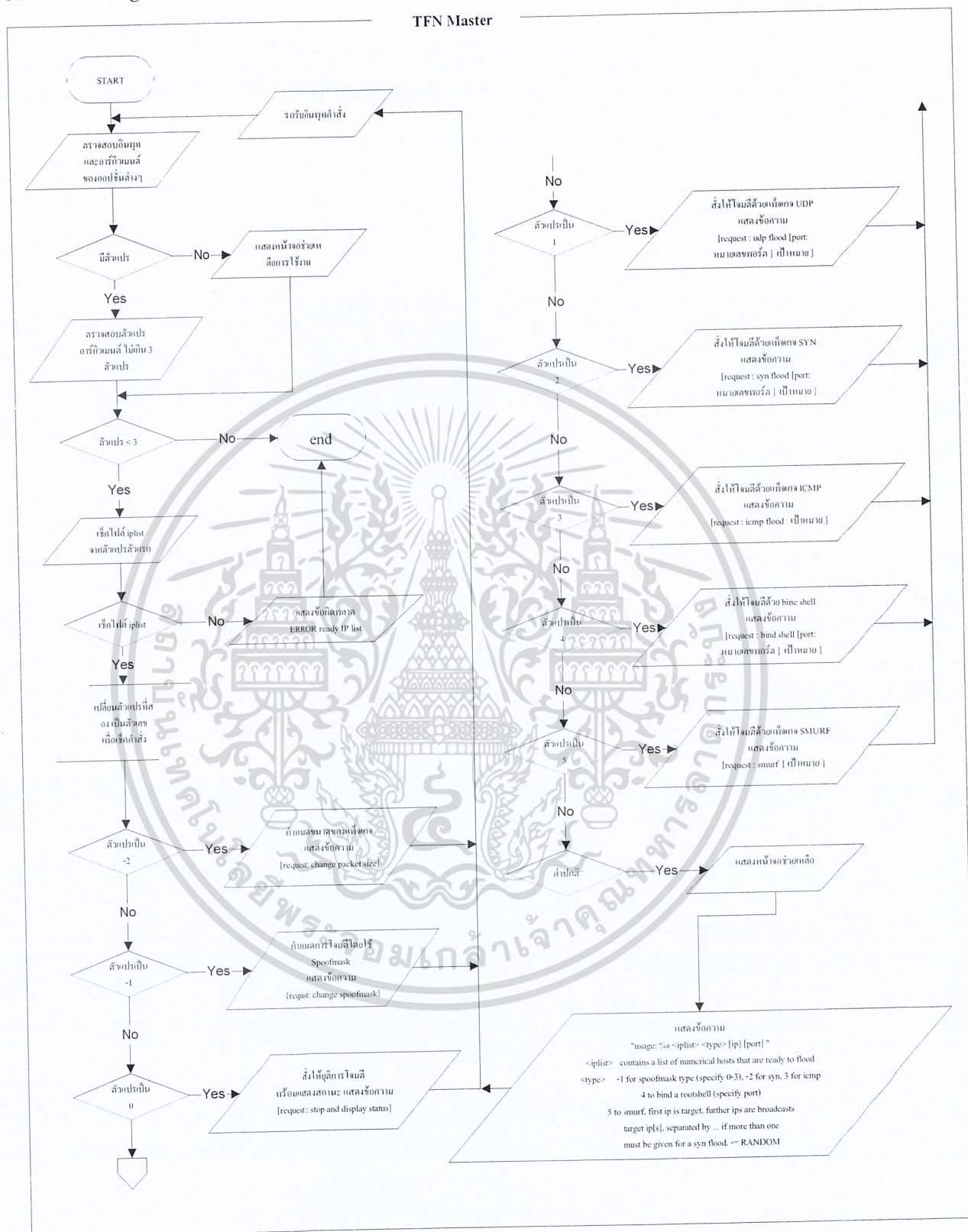
เนื่องจากโปรแกรมใช้แพ็กเก็ต ICMP_ECHOREPLY ในการติดต่อสื่อสาร มันเป็นการยากที่จะหลีกเลี่ยงการป้องกันโดยจะไม่หยุดการทำงานของโปรแกรมอินเทอร์เนตที่ใช้ ICMP การทำงานโดยจะหยุดการทำงานในรูปแบบนี้ จำเป็นที่จะต้องสังเกตและเรียนรู้ความแตกต่างของการใช้งานปกติระหว่าง ICMP_ECHO และ ICMP_ECHOREPLY

ข้อด้วย

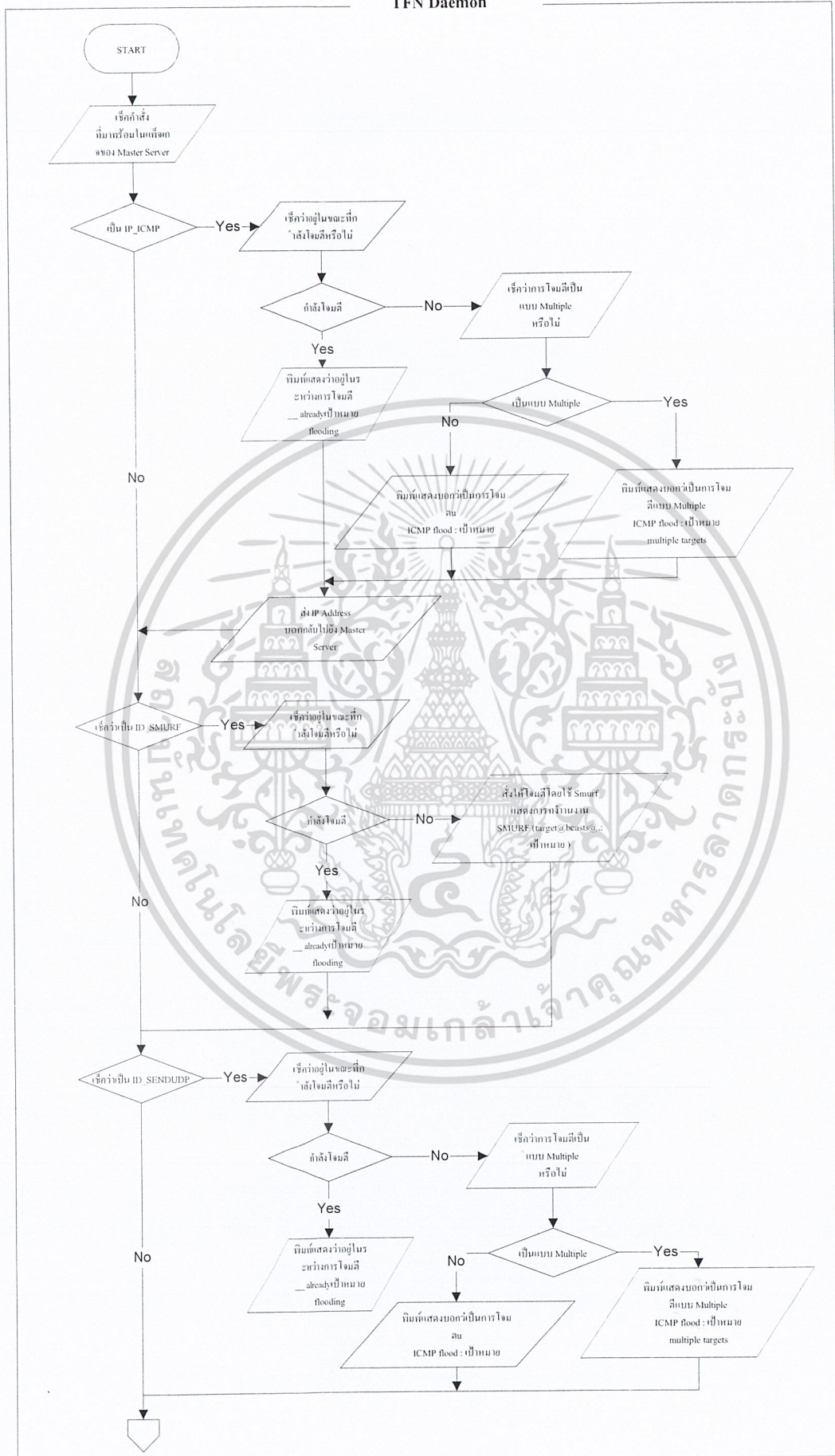
ถ้ามีการแก้ไขหรือเปลี่ยนแปลงโค้ดของ ไคลเอนต์ และ เดมอน จำเป็นที่จะต้องตรวจสอบสตริงในโปรแกรมไบนารีแทน อาทิเช่น โปรแกรม string หรือมีการใช้คำสั่ง “rtp” ของ Berkeley ในการแสดงคูการติดต่อสื่อสาร “rtp” (514/tcp) จากระบบบนเครือข่ายได้จะรวดเร็วกว่า และตั้งแต่ที่ TFN ใช้แพ็กเก็ต ICMP แทนแล้วทำให้การตรวจสอบการกระทำต่างๆยากขึ้นโดยโปรแกรม “ngrep” ไม่สามารถตรวจสอบได้ อีกอย่างหนึ่งคือ TFN ไม่มีการตรวจสอบสิทธิการเข้าใช้งาน



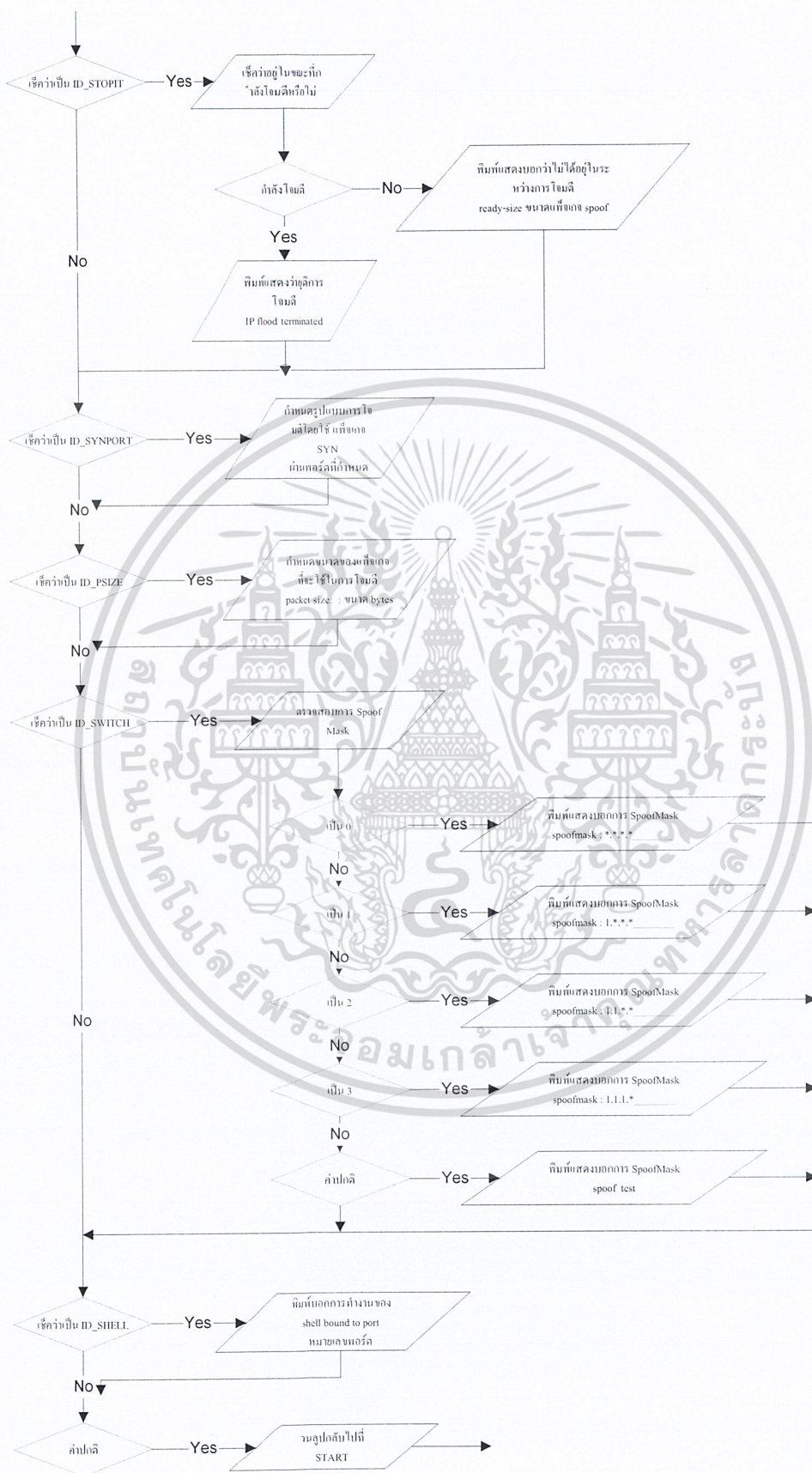
3.5.2.8 TFN Algorithm



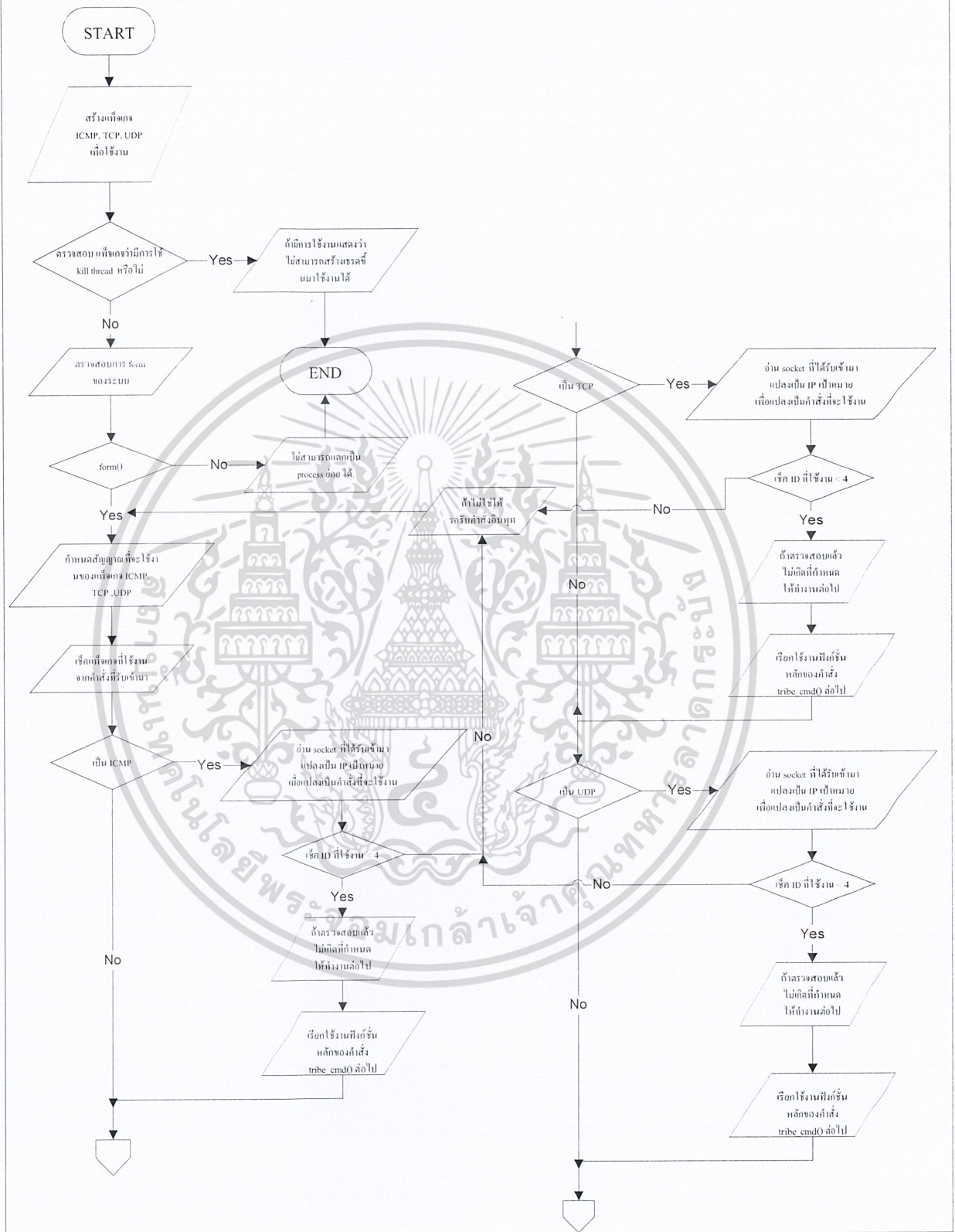
TFN Daemon



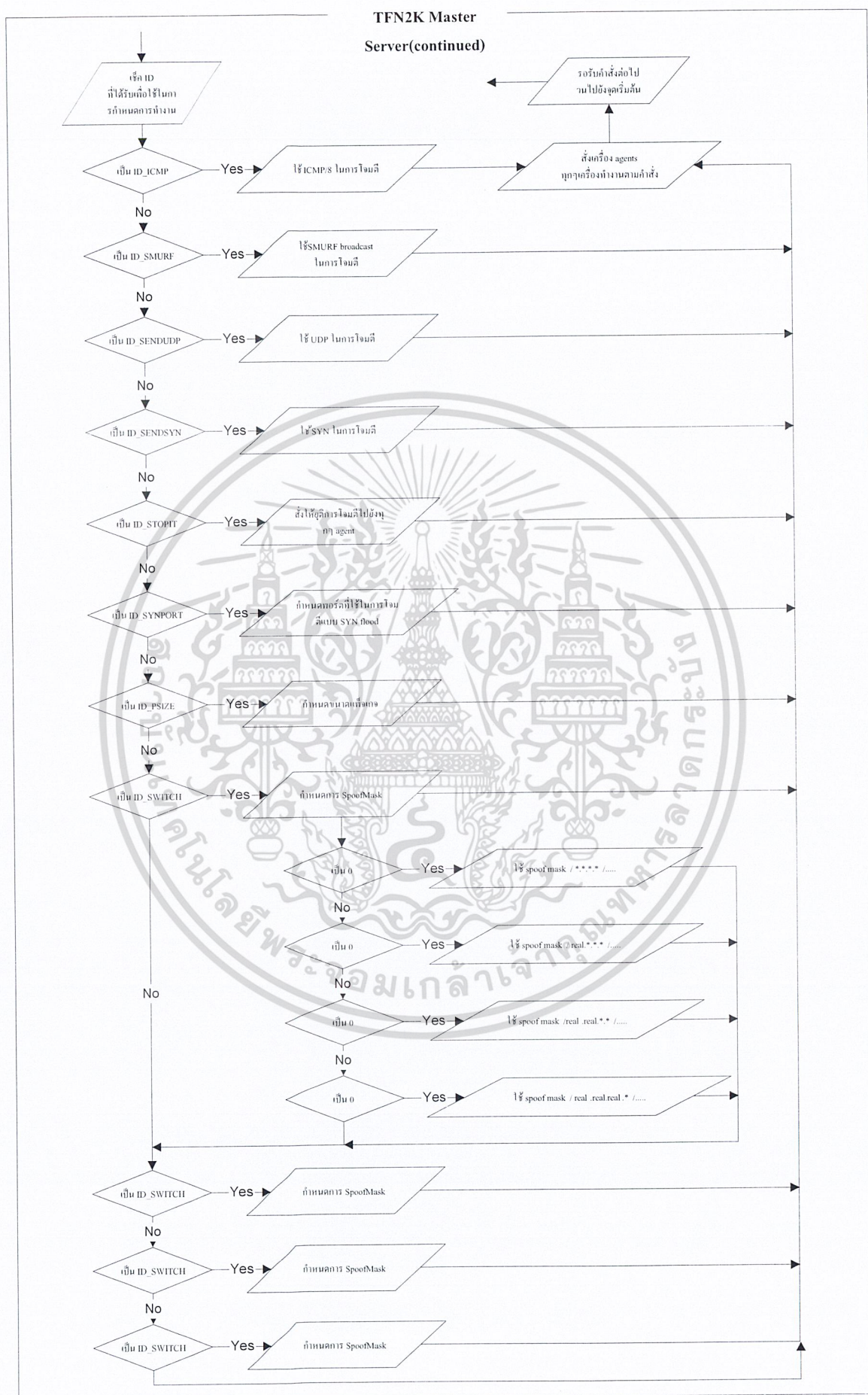
TFN Daemon (continue)



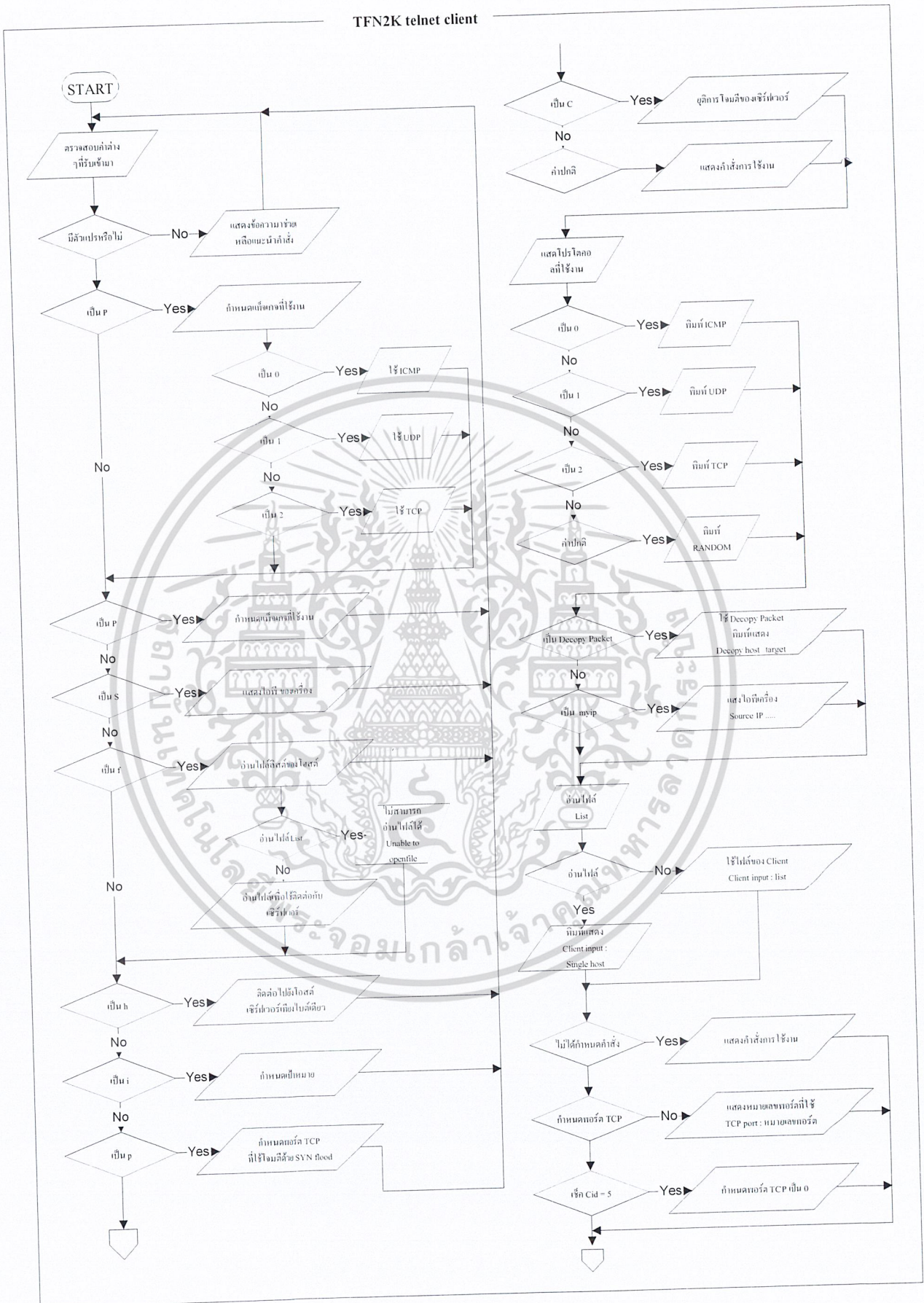
TFN2K Server



**TFN2K Master
Server(continued)**



TFN2K telnet client



3.5.3 Mstream

3.5.3.1 แนะนำ

Mstream เป็นทูลอีกตัวหนึ่งอยู่ในกลุ่ม DDoS โดยตัวเอนต์ตรวจพบในเดือนตุลาคม , 2000 สำหรับ RFC 2267 egress filtering ซึ่งเป็นมาตรฐานที่มีขนาดแพ็กเก็ตที่ถูกส่งออกจากเครือข่าย ซึ่งเป็นการเพิ่มความคับคั่งให้กับเครือข่าย ทำให้ระบบลดประสิทธิภาพลง โดยช่องว่างการสื่อสาร (bandwidth) จะถูกใช้งานมากขึ้น แต่การกำหนดพอร์ตสามารถทำได้โดยกำหนดค่าคุณลักษณะในเราเตอร์ (router) หรือ ผู้ให้บริการอินเทอร์เน็ต (ISP)

3.5.3.2 เครือข่าย

client (s) → handler (s) → agent (s) → victim (s)

เครือข่าย Mstream จะคล้ายกับ trinoos และ shaft โดยใช้แฮนด์เลอร์ (“master.c”) , และกลุ่มของเอเจนต์ (“server.c”) ผู้โจมตีติดต่อกับแฮนด์เลอร์โดยไม่ได้มีการเข้ารหัสผ่านที่พิเศษเฉพาะแฮนด์เลอร์กับเอเจนต์จะใช้ยูดีพี

3.5.3.3 การติดต่อสื่อสาร

Attacker to handler (s)	:	6723/tcp (ใน source code)
		15104/tcp (ในระบบทั่วไป)
		และ 12754/tcp (ในการค้นหาใหม่)
Agent to Handler (s)	:	9325/udp (ใน source code)
		6333/udp (ในระบบทั่วไป)
Handler to agent (s)	:	7933/udp (ใน source code)
		10498/udp (ในระบบทั่วไป)

การควบคุม mstream handler ผ่านที่ซีพีติดต่อกับพอร์ต 6723/tcp , 15104/tcp , 12754/tcp อื่นๆ ซึ่งทำให้ไปใช้งานโปรแกรมในรูปแบบเทเลเน็ตแต่จะใช้งานผ่านบรรทัดคำสั่ง ซึ่งไม่มีการเข้ารหัสระหว่างการติดต่อกับ Trinoos ตรงที่ใช้ ยูดีพี ในการติดต่อระหว่างแฮนด์เลอร์ กับ เอเจนต์หลังจากติดต่อให้ใส่พาสเวิร์ดปกติเป็น “N7%diApf” ในการค้นหาใหม่หรือ “sex” ใน ซอสโค้ดเดิม

3.5.3.4 คำสั่งของ Mstream

คำสั่งของ handler

โดยถ้าไม่มีการติดต่อภายใน 420 วินาที หรือ 7 นาที จะยุติการติดต่อโดยทันที

help	แสดงคำสั่ง
stream	stream attack!
servers	แสดงรายชื่อของทุกๆเซิร์ฟเวอร์
ping	ping ไปยังทุกๆเซิร์ฟเวอร์
who	แสดงไอพีของผู้ล็อกอิน มาใช้เครือข่าย mstream อนุญาตให้มีมากกว่าหนึ่งไอพีได้
servers	แสดงรายชื่อเอเจนต์

who แสดงชื่อคนใช้งานปัจจุบัน

ping ส่งแพ็กเก็ต “ping” ไปยังเอเจนต์ทุกตัว ซึ่งจะตอบกลับด้วย “pong”

stream <hostname> <seconds>

 เริ่มการ โจมตีการปิดการให้บริการแบบกระจายไปยังโฮสต์เดียว

mstream <ip1:ip2:ip3:...> <seconds>

 เริ่มการ โจมตีการปิดการให้บริการแบบกระจายไปยังโฮสต์หลายๆตัวในระยะ
เวลาที่กำหนด

quit ยุติการเชื่อมต่อกับเซิร์ฟเวอร์

คำสั่ง Agent

Ping ตอบกลับโดยบอกที่อยู่ไอพีด้วย “Pong”

Stream/IP/second เริ่มต้นการส่ง stream ไปยังที่อยู่ไอพี

Mstream/IP1[:IP2[:IPN]]/second

 ส่ง stream โจมตีไปยังโฮสต์ได้หลายตัวในระยะเวลาที่กำหนด

3.5.3.5 ลักษณะการโจมตี

การโจมตีเริ่มต้นจากการทำเครือข่าย stacheldraht ด้วยการหาเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย
ซึ่งมีจุดอ่อนที่สามารถเจาะเข้าไปควบคุมการทำงานได้จากระยะไกล และเมื่อได้เป็นรูทของระบบแล้วจะ
ทำให้เรามีสิทธิ์ที่จะนำเอเจนต์ไปติดตั้ง และคอมไพล์เพื่อใช้งาน โดยจะทำการเก็บไอพีแอดเดรสของ
เครื่องนั้น ไว้ในเครื่องเซิร์ฟเวอร์เมื่อควบคุมเครื่องนั้นๆ

ผู้โจมตีจะทำการควบคุมเครื่องเซิร์ฟเวอร์จากระยะไกล ให้ไปส่งงานโจมตีเครื่องเป้าหมาย หรือ
เครื่องที่ตกเป็นเหยื่อที่ต้องการด้วยคำสั่งเพียงคำสั่งเดียว

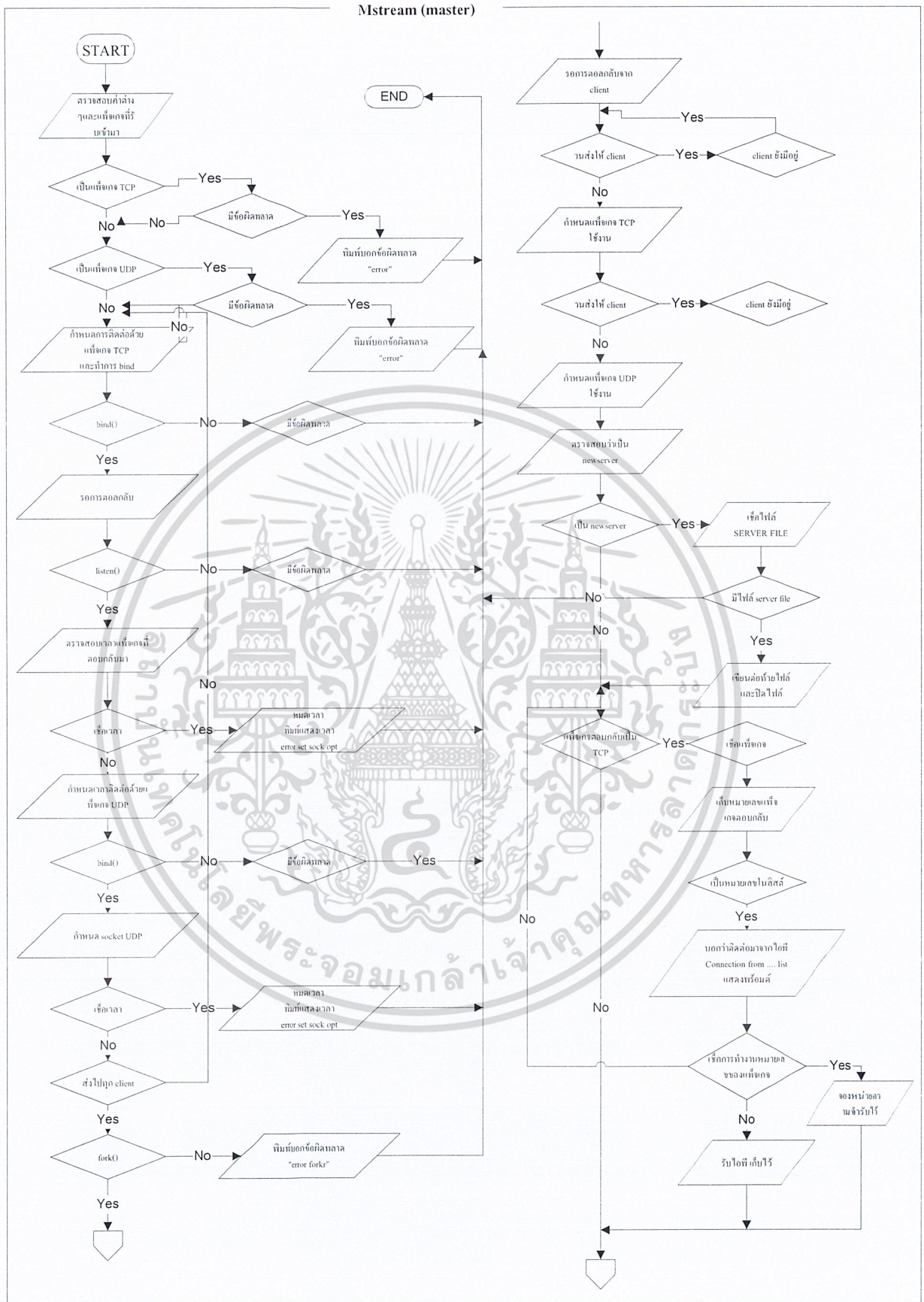
3.5.3.6 การป้องกัน

เน้นย้ำที่จะป้องกันไม่ให้เครื่องเราตกเป็นเหยื่อในการตั้งเครือข่ายของระบบการโจมตีนี้ โดย
ศึกษา และหาทางป้องกัน ทั้งจากตัวระบบปฏิบัติการ และโปรแกรมที่ทำหน้าที่ป้องกันการโจมตี

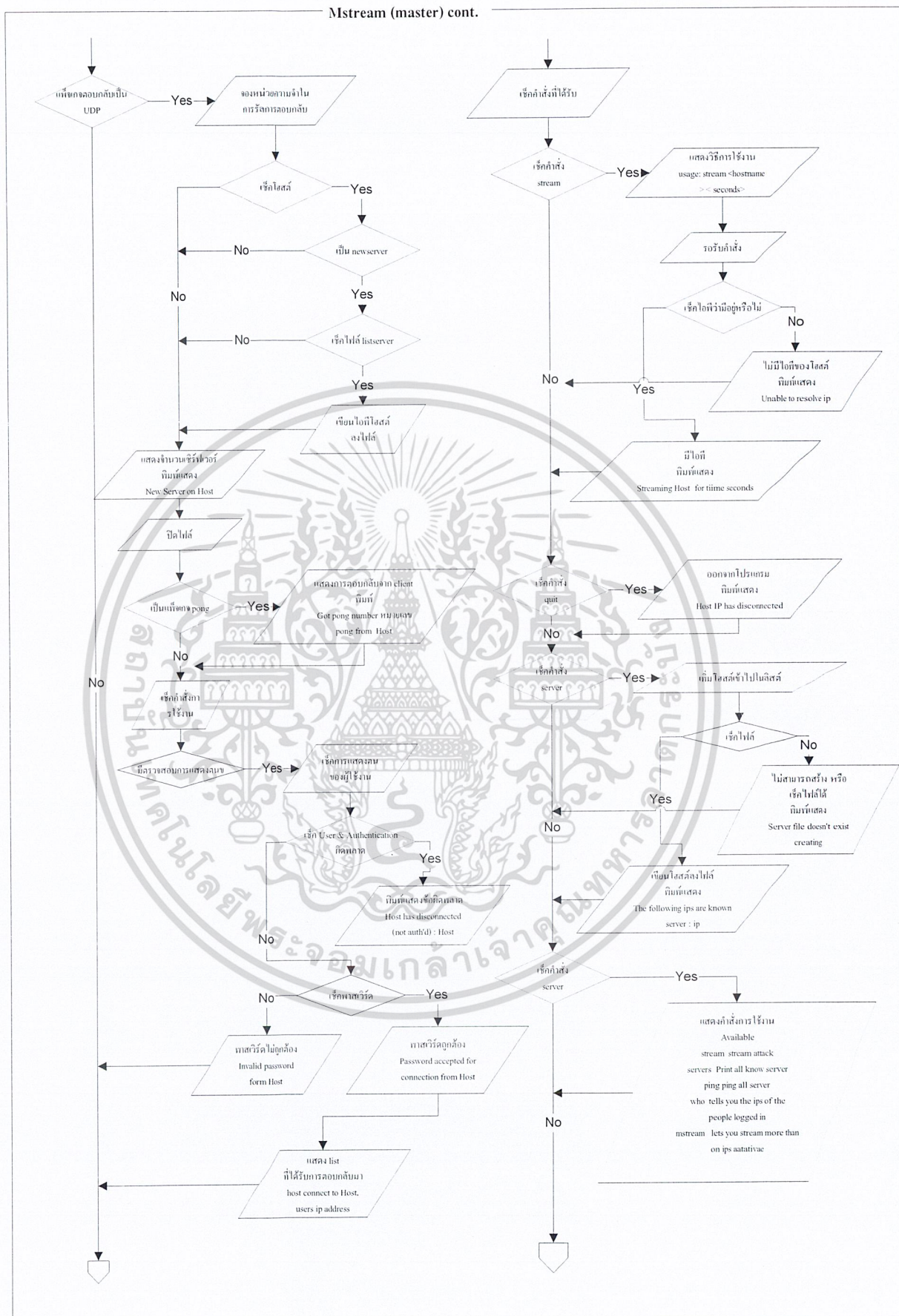
ขยายความสามารถในการรองรับบริการ และการ โจมตี (High Availabilities) ด้วยการเพิ่มประ
สิทธิภาพให้กับเครื่องในระบบ และมีการทำระบบสำรอง เมื่อเกิดระบบพังเสียหาย (Fail Over) ทั้งจากการ
โจมตี และการใช้งานปกติ

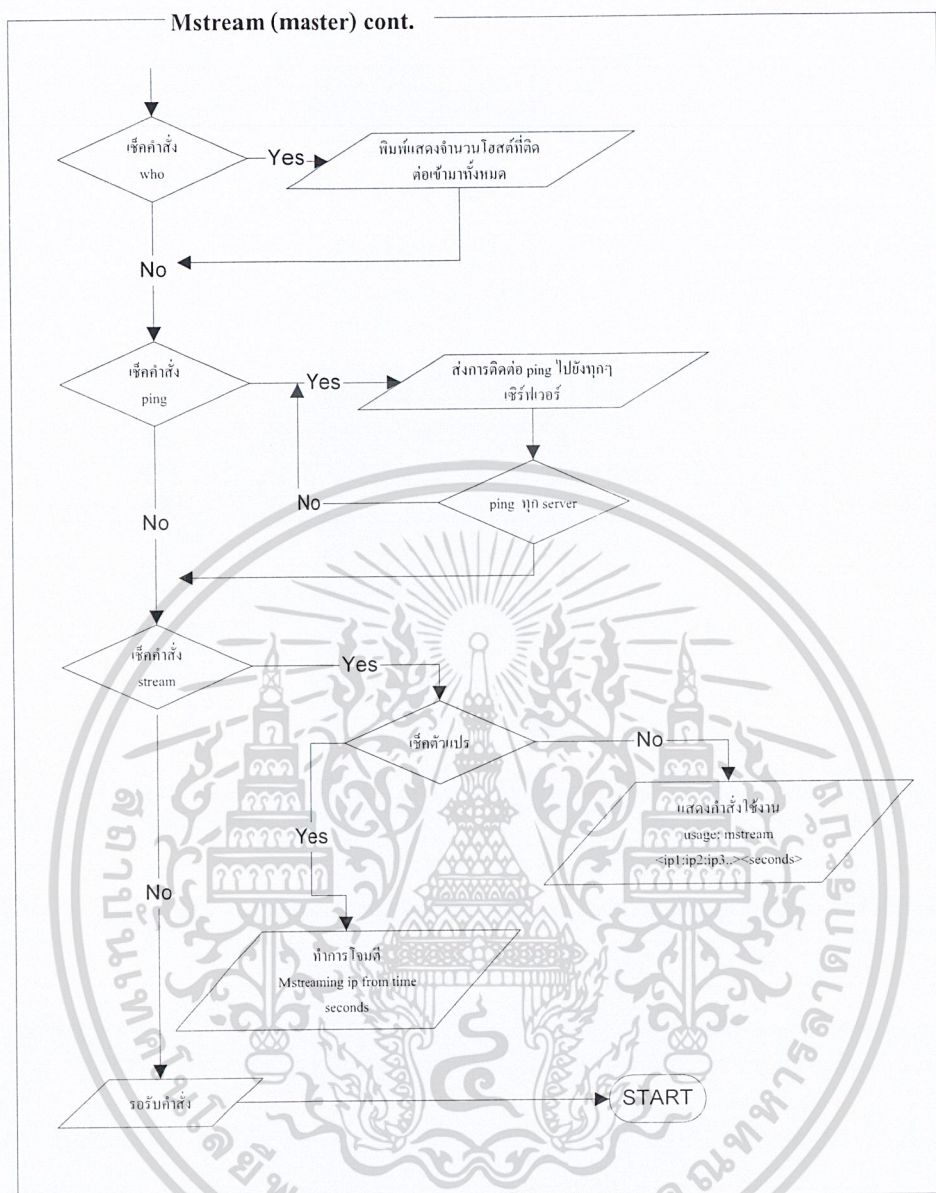
3.5.3.7 Mstream Algorithm

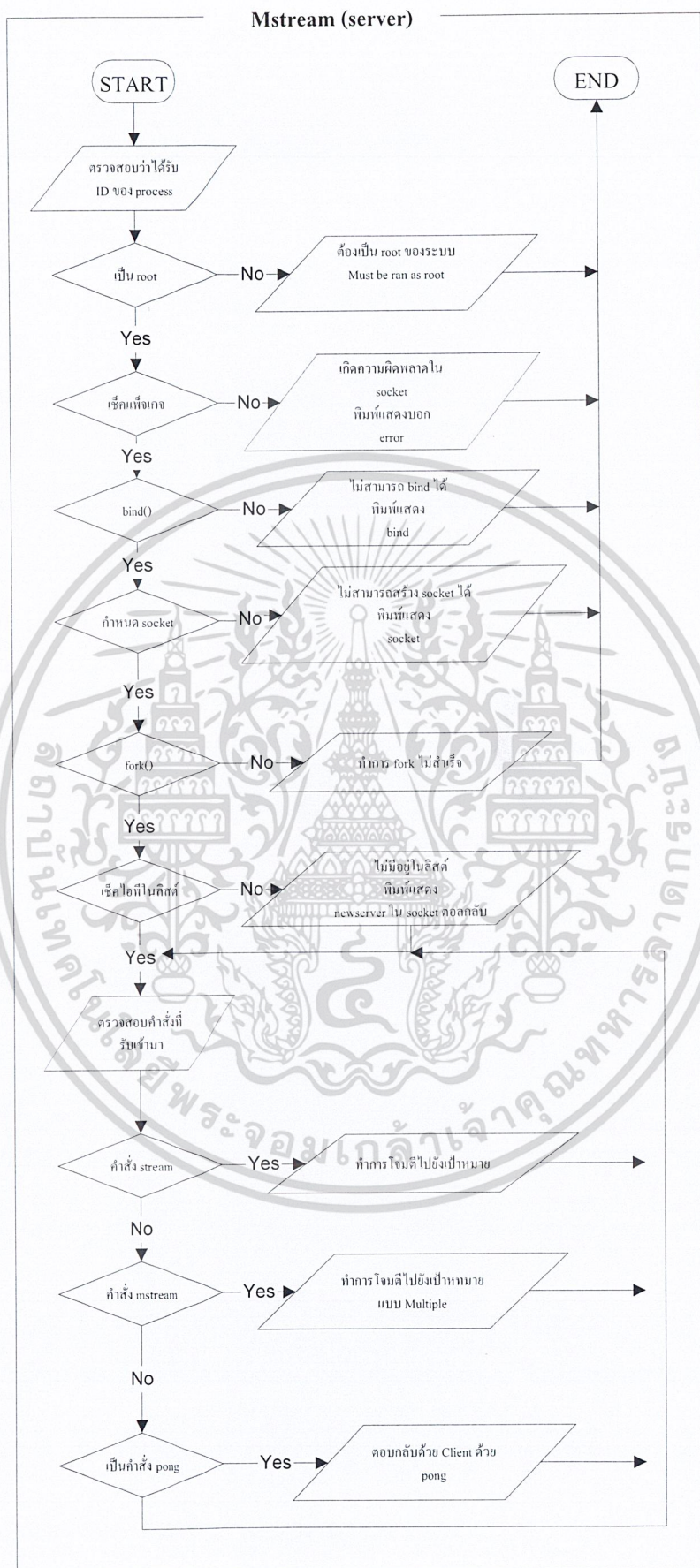
Mstream (master)



Mstream (master) cont.







3.5.4 Stacheldraht

3.5.4.1 แนะนำ

เป็นการพัฒนามาจาก Tribe Flood Network หรือ TFN โดย Stacheldraht ในภาษาเยอรมันเรียกว่า “barbed wire” ซึ่งรวมคุณลักษณะของ Trinoo ซึ่งเป็นพื้นฐานของ TFN เข้าไปด้วย และมีการเพิ่มการเข้ารหัสในการติดต่อสื่อสารหรือผู้โจมตีและ stacheldraht มาสเตอร์และมีการอัปเดตตัวเองอย่างต่อเนื่องอัตโนมัติ ซึ่งคล้ายกันกับ Trinoo โดย stacheldraht จะมีมาสเตอร์(แฮนด์เลอร์) และ เดมอน หรือ “bcast” (เอเจนต์) และระหว่างนี้ก็ได้มีการพัฒนาต่อจาก TFN ด้วย

โดยคุณลักษณะของ TFN จะมี ICMP Flood, SYN Flood, UDP Flood และการโจมตีแบบ “Smurf”

ข้อด้อยของ TFN คือ การติดต่อระหว่างผู้โจมตีและมาสเตอร์จะใช้การควบคุมเครือข่ายด้วยรูปแบบ clear-text ซึ่งอิงมาตรฐานการโจมตีแบบที่ซีพี และ stacheldraht จะมีการเพิ่มการเข้ารหัสด้วยเรียกว่า stacheldraht term โดย stacheldraht เอเจนต์เดิมพบในระบบปฏิบัติการ Solaris 2.x โดยจะมีการสังเกตได้ว่าในแคชของไฟล์จะมีการขโมยบัญชีรายชื่อไว้ด้วย ซึ่งในเบื้องต้นจะอ้างอิงบนระบบปฏิบัติการ Red Hat 6.0 แทน ซึ่งดูว่าลักษณะก็จะคล้ายกับ Trinoo และ TFN

โดยผู้โจมตีต้องใช้โปรแกรมคล้ายๆเทลเน็ต (“telnet/client.c”) ในการเชื่อมต่อไปยังแฮนด์เลอร์ (มาสเตอร์) ในโค้ดจะจำกัดไว้ 1,000 เอเจนต์ โดยเอเจนต์จะทำหน้าที่โจมตีผ่านแพ็กเก็ตจากการรับคำสั่งจากแฮนด์เลอร์ (“mserver”) หรือ “master server” ในโค้ด

3.5.4.2 เครือข่าย

client (s) → handler (s) → agent (s) → victim (s)

เครือข่าย stacheldraht จะคล้ายกับ trinoo และ shaft โดยใช้แฮนด์เลอร์ (“mserver.c”) , และกลุ่มของเอเจนต์ (“client.c”) ผู้โจมตีติดต่อกับแฮนด์เลอร์โดยไม่ได้มีการเข้ารหัสผ่านที่พิเศษเฉพาะแฮนด์เลอร์กับเอเจนต์จะใช้ยูดีพี

3.5.4.3 การติดต่อสื่อสาร

Client to handler(s) : 16660/tcp

Handler to/from agent(s) : 65000/tcp , ICMP_ECHOREPLY

การเข้าควบคุมระยะไกลของเครือข่าย stacheldraht จะใช้การเข้ารหัสแบบ Symmetric ในการสื่อสาร โดยไคลเอนต์จะส่งอาร์กิวเมนต์เดียวไปยังแฮนด์เลอร์ผ่านทางพอร์ต 16660/tcp และมีการใส่พาสเวิร์ด ซึ่งจะมีการบอกความช่วยเหลือในการเรียกสั่งในเอเจนต์ทำงานพร้อมผ่านแฮนด์เลอร์โดย

a! แสดงเอเจนต์ที่กำลังทำงาน

d! แสดงเอเจนต์ที่ไม่ทำงานแล้ว

.help แสดงข้อความช่วยเหลือ

3.5.4.4 การป้องกันพาสเวิร์ด

หลังจากติดต่อไปยังแฮนด์เลอร์โดยใช้โปรแกรมไคลเอนต์แล้ว จะมีการให้ใส่พาสเวิร์ดปกติจะมีค่าเป็น “sicken” โดยใช้มาตรฐานการเข้ารหัสแบบ Blowfish ในการติดต่อระหว่างเครือข่ายไปยังแฮนด์

เลอร์ และระหว่างแฮนด์เลอร์กับเอเจนต์ก็เช่นเดียวกัน ซึ่งจะคล้ายกับ TFN โดยมีการกำหนดค่าไว้ใน “config.h”

3.5.4.5 คำสั่ง

.distro user server	เป็นการสั่งให้ติดตั้งเอเจนต์และมีการก๊อปปี้ตัวเองโดยใช้คำสั่ง “rcp” หรือ บนระบบ “server” และใช้บัญชีรายชื่อ “user”
.help	แสดงรายการคำสั่ง
.killall	ยุติการทำงานของเอเจนต์ที่ทำงานอยู่
.madd ip1[:ip2[:ipN]]	เพิ่มที่อยู่ไอพีที่จะทำเป็นเหยื่อในการโจมตี
.mdie	ส่งการร้องขอให้หยุดการทำงานไปยังเอเจนต์ทุกตัว
.mdos	เริ่มต้นการโจมตีแบบปิดการให้บริการ (DoS)
.micmp ip1[:ip2[:ipN]]	เริ่มการโจมตีแบบ ICMP Flood ไปยังโฮสต์ที่ระบุ
.mlist	แสดงบัญชีรายชื่อโฮสต์ที่กำลังโจมตีการปิดการให้บริการ (DoS) ในขณะนี้
.mping	เป็นการ ping แบบกระจายเพื่อดูว่าเอเจนต์ใดบ้างที่ทำงานอยู่
.msadd	เพิ่มมาสเตอร์เซิร์ฟเวอร์ตัวใหม่ (แฮนด์เลอร์) ไปยังบัญชีรายชื่อของเซิร์ฟเวอร์
.msort	จัดเรียงชื่อเอเจนต์ที่ยังคงอยู่และตายแล้ว (bcast) โดยการส่ง ping และ แสดงจำนวนเอเจนต์ที่นับได้เป็นเปอร์เซ็นต์ของ dead/alive เอเจนต์
.mstop ip1[:ip2[:ipN]]	ยุติการโจมตีไปยังไอพีที่กำหนดหรือทั้งหมด
.mstop all	ยุติการโจมตีไปยังไอพีที่กำหนดหรือทั้งหมด
.msrem	เอาชื่อมาสเตอร์เซิร์ฟเวอร์ (แฮนด์เลอร์) ออกจากบัญชีรายชื่อของเซิร์ฟเวอร์
.msyn ip1[:ip2[:ipN]]	เริ่มต้นการโจมตีแบบ syn Flood ไปยังโฮสต์
.mtimer seconds	กำหนดระยะเวลาในการโจมตีเป็นวินาที
.mudp ip1[:ip2[:ipN]]	เริ่มต้นการโจมตีแบบยูดีพี Flood ไปยังโฮสต์
.setisize	กำหนดขนาดแพ็กเก็ต ICMP ในการ flood (สูงสุด : 1024 ค่าปกติเป็น 1024)
.setusize	กำหนดขนาดแพ็กเก็ตยูดีพี ในการ flood (สูงสุด : 1024 ค่าปกติเป็น 1024)
.showalive	แสดงเอเจนต์ที่ยังคงทำงานอยู่แบบ bcasts
.showdead	แสดงเอเจนต์ที่ไม่ทำงานแบบ bcasts
.sprange lowport-highport	กำหนดพอร์ตที่ใช้ในการโจมตีแบบ SYN Flood (ค่าปกติเป็น lowport : 0 , highport : 140)

3.5.4.6 ลักษณะการโจมตี

ใน Trinoo และ TFN จะใช้การติดตั้งโปรแกรม แฮนด์เลอร์/เอเจนต์ เหมือนการติดตั้งโปรแกรม ของระบบปฏิบัติการยูนิกซ์โดยทั่วไป แต่ stacheldraht จะใช้คำสั่ง “rcp” ของ Berkeley ผ่านทางพอร์ต 514/tcp จะใช้การขโมยบัญชีรายชื่อจากไชต์ที่อยู่บนแคช (cache) เมื่อเอเจนต์ทำงานจะพยายามติดต่อมาสเตอร์เซิร์ฟเวอร์

โดยดูจากไฟล์ที่กำหนดคุณลักษณะที่จะควบคุมมัน ซึ่งในไฟล์จะมีรายชื่อที่อยู่ของไอพี จะเข้ารหัสโดยใช้ Blowfish เป็น “randomsucks” ถ้าหาไม่เจอจะใช้ค่าที่เป็นค่าปกติแทนเอเจนต์หนึ่งตัวจะถูกควบคุมจากแฮนด์เลอร์เดียว เมื่อเริ่มส่งแพ็กเก็ต ICMP_ECHOREPLY ในฟิลด์ไอดี จะมีเป็น 666 และฟิลด์ค่าตัว เช่น “skillz” เมื่อมาสเตอร์ได้รับจะส่ง ICMP_ECHOREPLY กลับโดยในฟิลด์ไอดี เป็น 667 และฟิลด์ค่าตัว เป็น “ficken” ถ้าในฟิลด์มีการเซต Type of Service เป็น 7 ซึ่งปกติเป็น 0 เมื่อมาสเตอร์ได้รับ ICMP_ECHOREPLY ID เช่น 1000, ฟิลด์ค่าตัว เป็น “Spooftworks” จากสตริง “skillz” , “spooftworks” , “sicken\n” , “niggahbitch” และ “ficken” เป็นส่วนของข้อมูล ICMP ซึ่งไม่มีการเข้ารหัส และถ้าฟิลด์ไอดี เป็น 666 , 667 , 668 , 669 และ 1000 สามารถนับได้อีกหลายวิธี

ข้อดี

Stacheldraht ไม่มีการแสดงตนของแหล่งที่มาของแพ็กเก็ต ICMP และยังไม่มีการเข้ารหัสของข้อความในแพ็กเก็ต ICMP

ถ้าค่าต่างๆไม่มีการเปลี่ยนแปลงไปจากเดิมแล้วค่าแพ็กเก็ตที่ส่งไปให้เอเจนต์จะเป็น

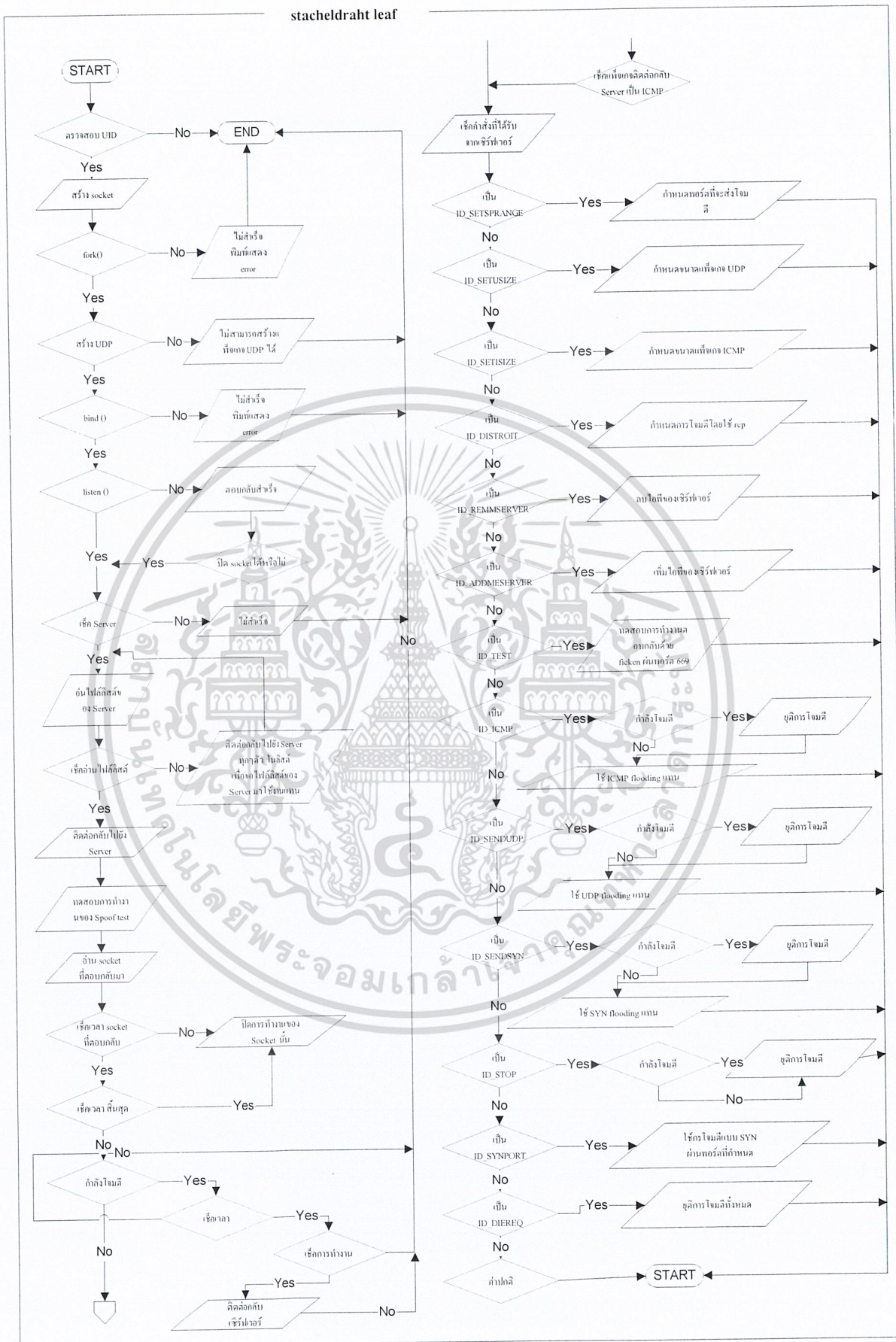
1. การส่งแพ็กเก็ต ICMP_ECHOREPLY ในฟิลด์ไอดี จะมีค่าเป็น 668 และดูแพ็กเก็ต ICMP_ECHOREPLY ตอบกลับด้วยค่า 669 และฟิลด์ค่าตัว เป็น “sicken\n”
2. ถ้าส่งแหล่งที่มาของที่อยู่ (source address) เป็น “3.3.3.3” (โดยค่าไอดี เป็น 666 และฟิลด์ค่าตัว เป็น “skillz”) และตอบกลับมาเป็นฟิลด์ไอดี ด้วยค่า 1000 และฟิลด์ค่าตัว เป็น “spooftworks”

3.5.4.7 การป้องกัน

พยายามป้องกัน โดยการค้นหาแนวทาง อาทิ เช่น การค้นหาทางเพื่อให้อีกเป็นเหยื่อ หรือเป็นเป้าหมายในการโจมตี และไม่ต้องเป็นเครื่องมือในการเป็นลูกข่ายในการโจมตี

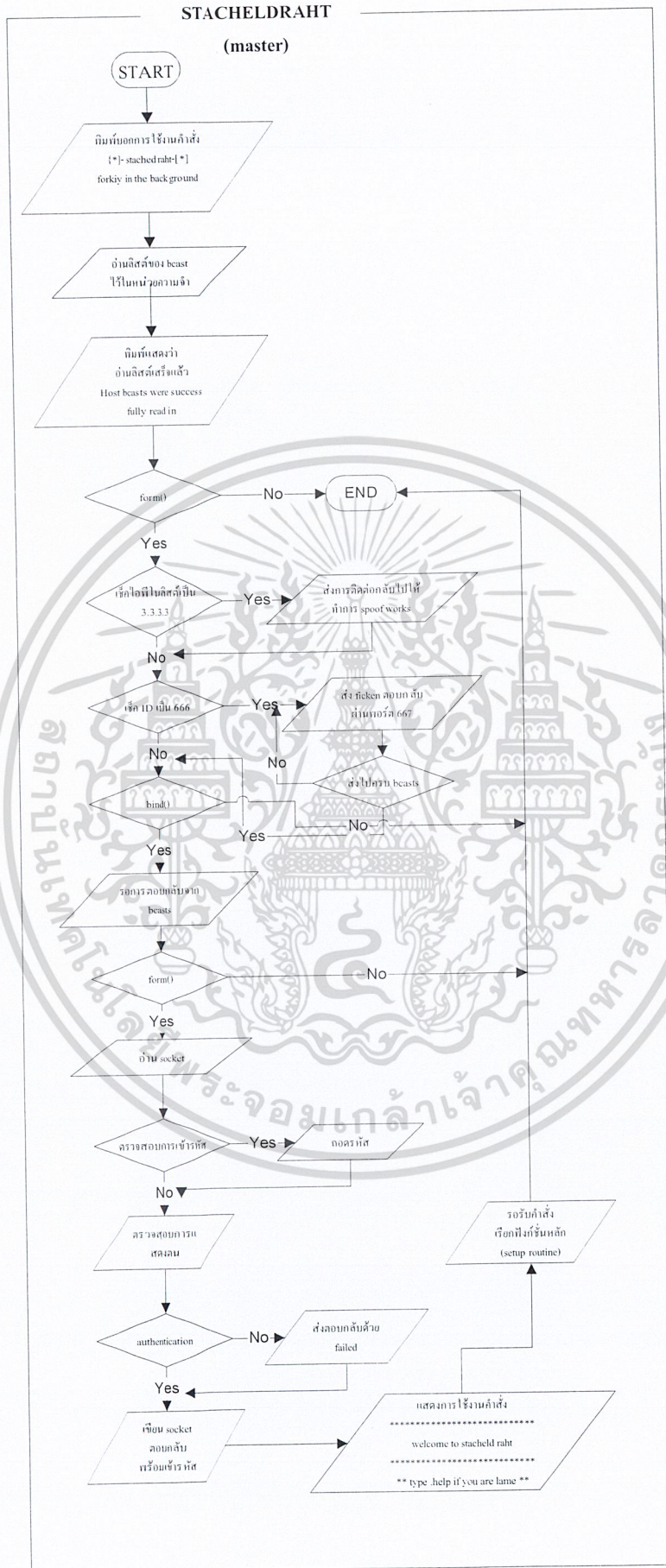
3.5.4.8 Stacheldraht Algorithm

stacheldraht leaf

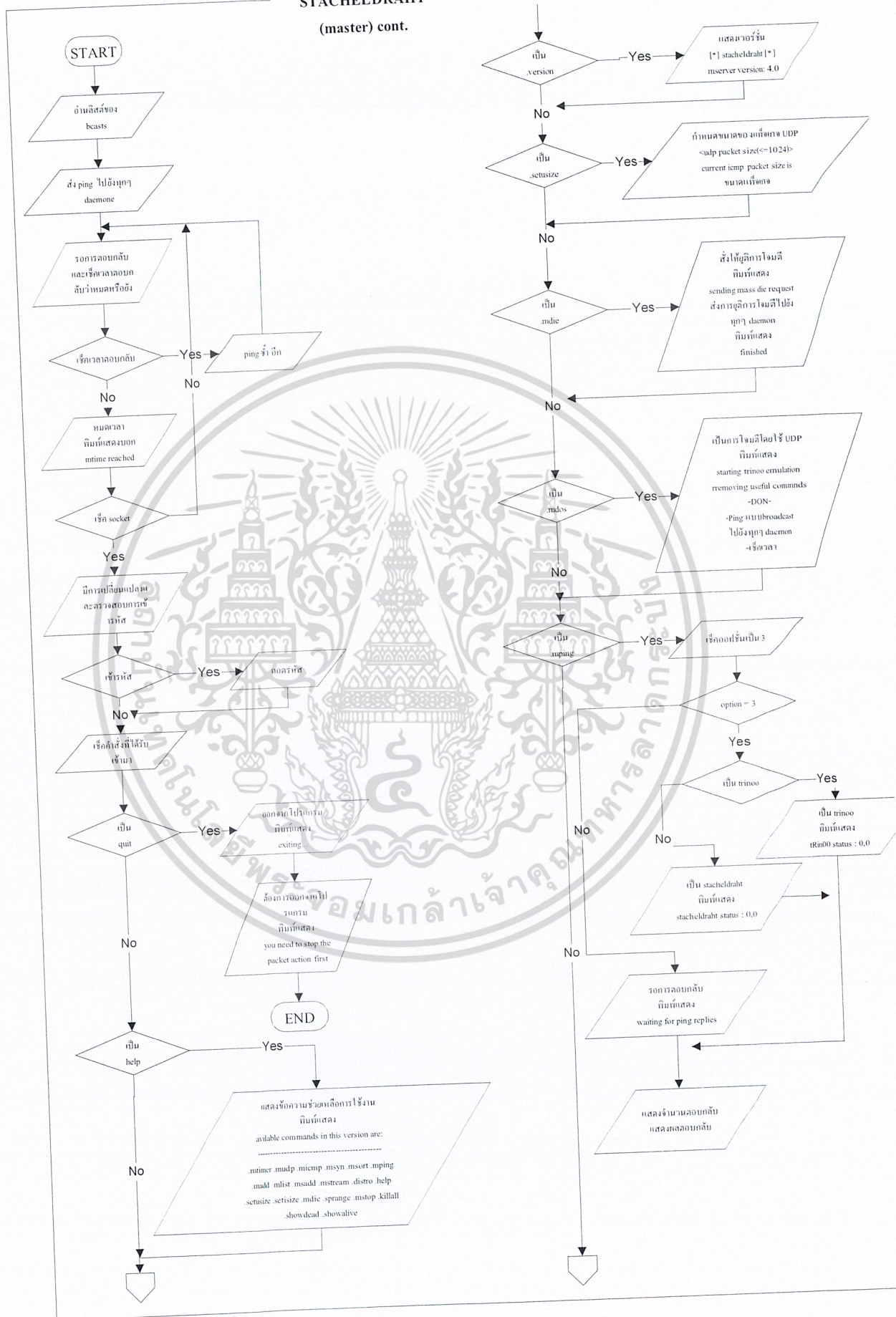


STACHELDRAHT

(master)

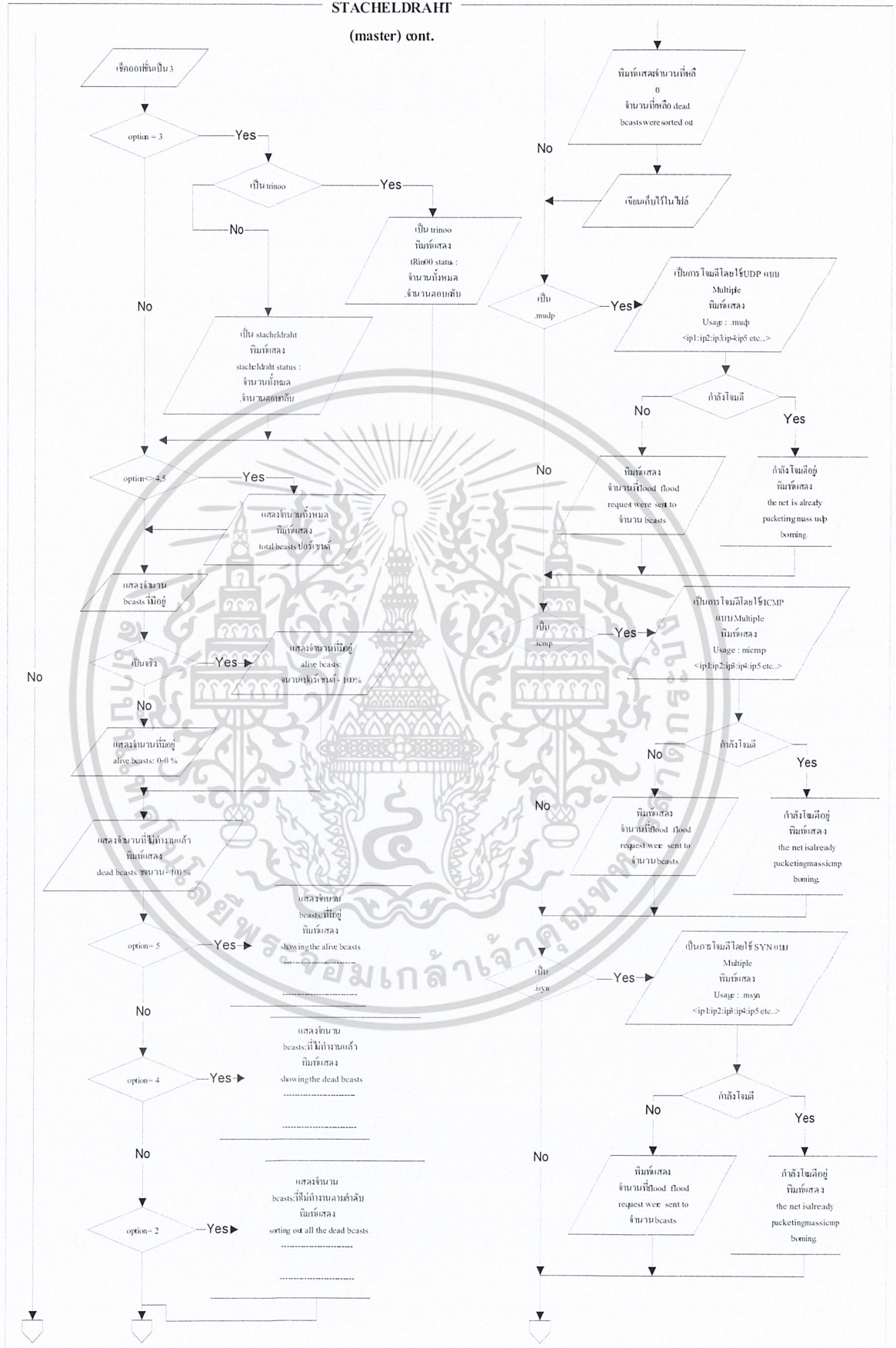


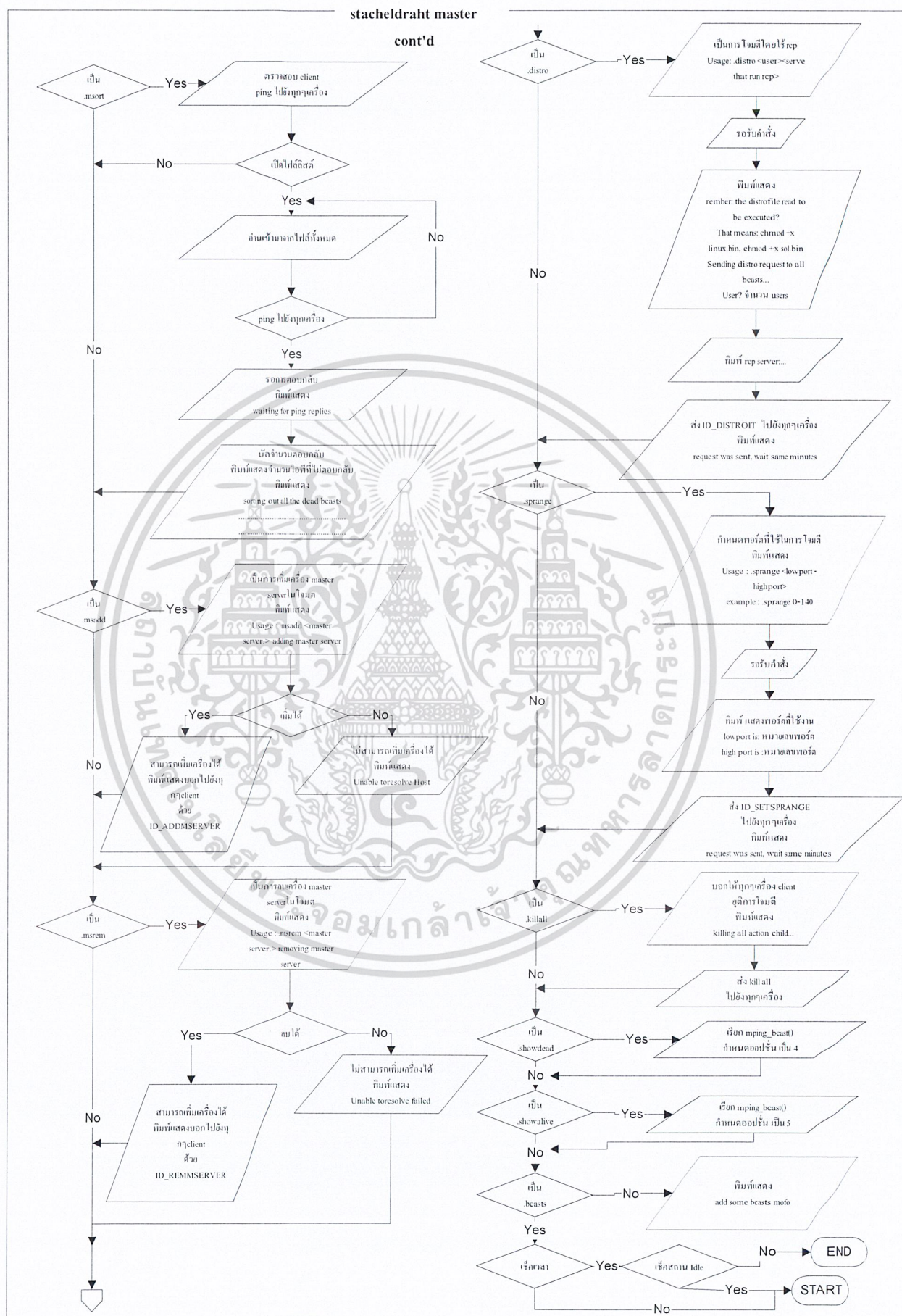
STACHELDRAHT (master) cont.



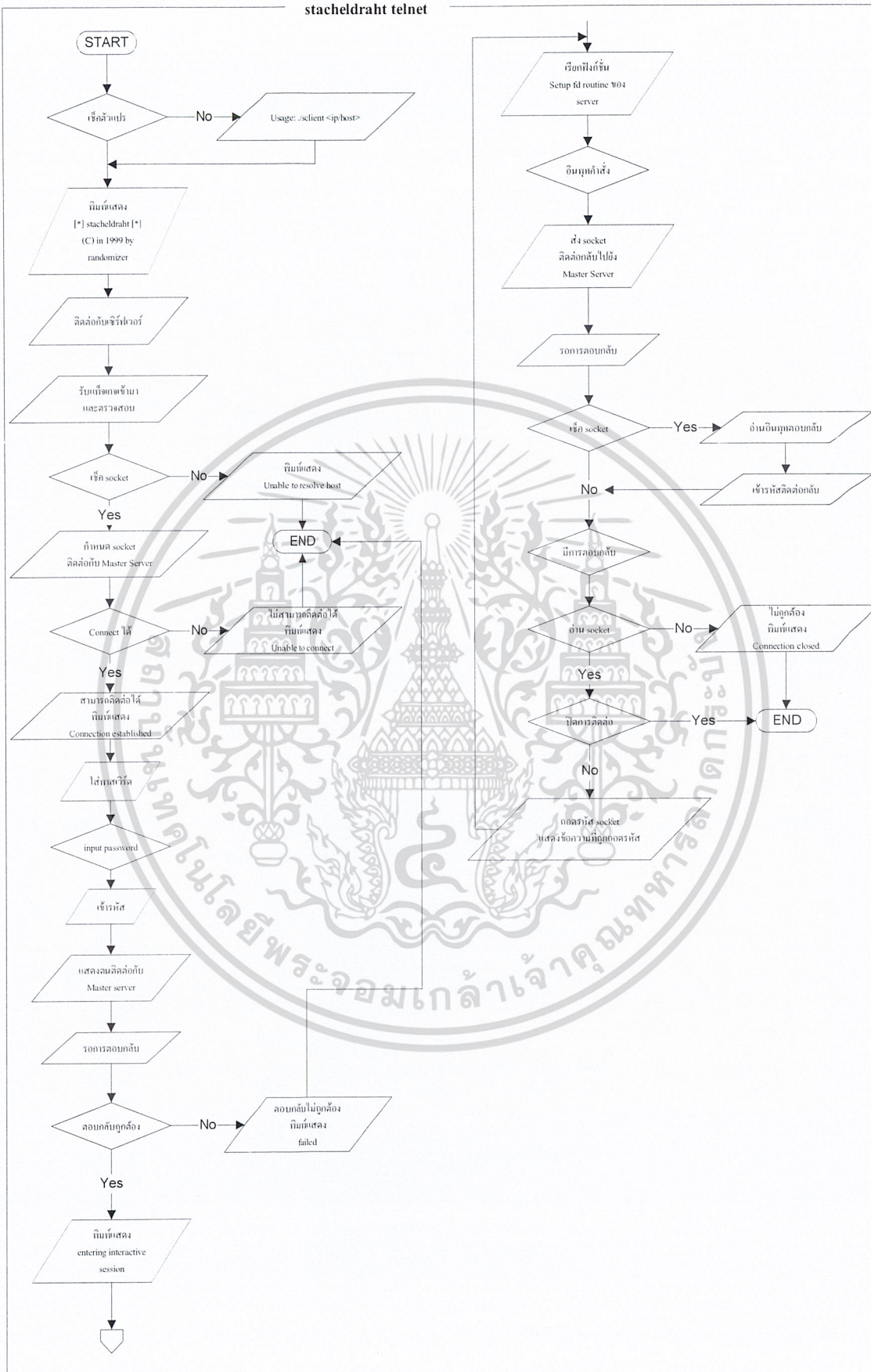
STACHELDRAHT

(master) cont.





stacheldraht telnet



บทที่ 4

เครือข่ายต้นแบบเพื่อต่อต้านการโจมตีด้วย DDoS

4.1 โมเดล 3 – Tier Architecture of DDoS Defending System

4.2 คอมโพเนนต์ของ N – Tier Architecture of DDoS Defending System

4.3 เราเตอร์ (Router)

4.4 ไฟร์วอลล์ (Firewall)

4.5 ระบบต่อต้านการโจมตี (Intrusion Detection System) และ NIDS

4.6 SSL Architecture

4.7 คลัสเตอร์ (Cluster)

4.8 โหลดบาลานซ์ (Load Balance)

4.1 โมเดล 3 – Tier Architecture of DDoS Defending System

เป็นโมเดลที่ออกแบบเพื่อรองรับการโจมตี ในหลายๆรูปแบบ และยังช่วยเพิ่มประสิทธิภาพการให้บริการ และสามารถใช้เป็นต้นแบบในการนำไปประยุกต์ใช้งานจริงได้ โดยอาศัยเทคโนโลยีแบบกระจาย (Distributed Technology) ในการออกแบบและประยุกต์ใช้กับระบบเครือข่ายคอมพิวเตอร์ที่ต้องการได้อย่างมีประสิทธิภาพ

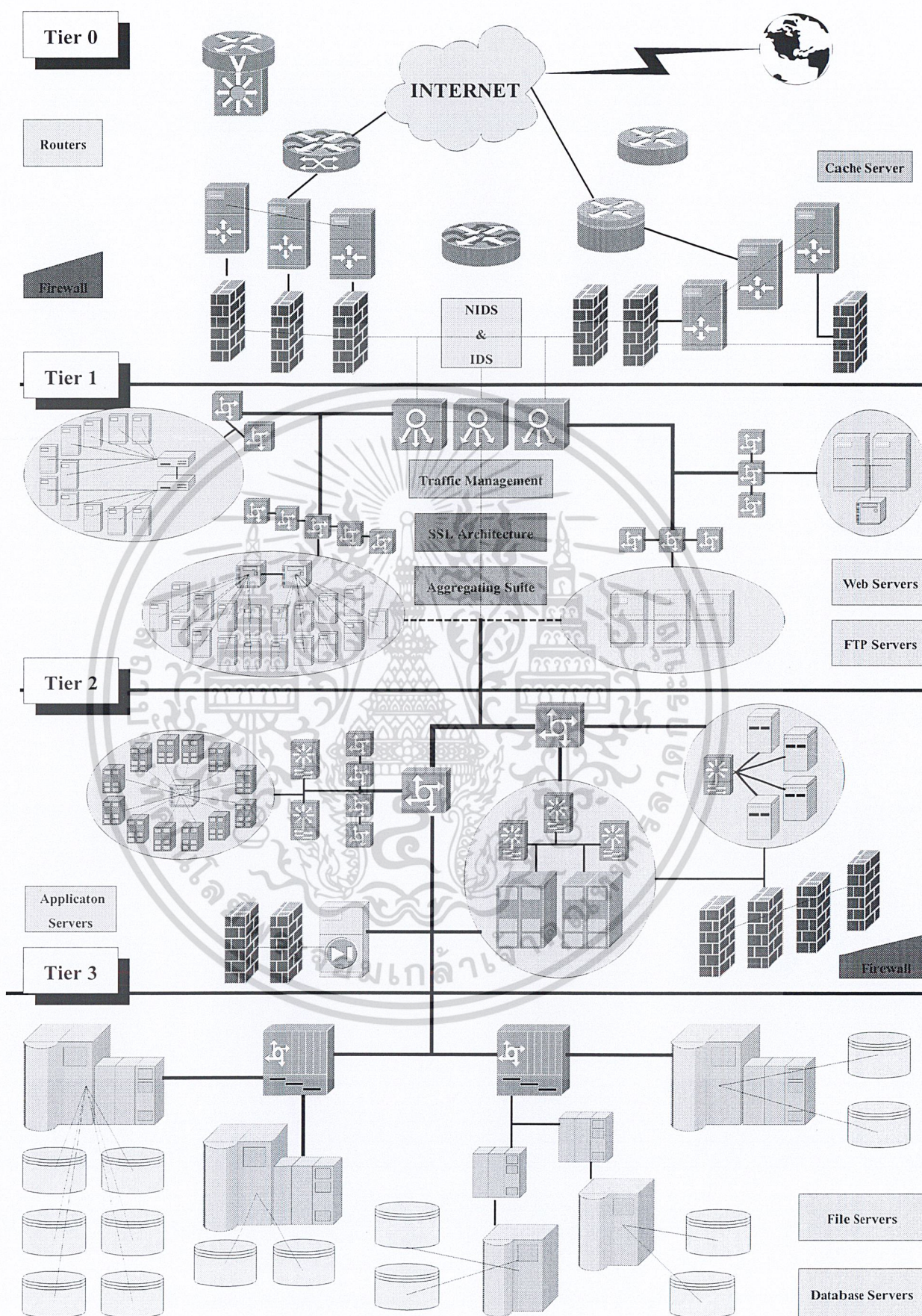
มีคอมโพเนนต์ที่สำคัญๆ ซึ่งได้แบ่งออกเป็น 3 Tier

Tier0 ประกอบด้วย เราเตอร์ (Router), ไฟร์วอลล์ (Firewall), IDS, Network IDS

Tier1 ประกอบด้วย Traffic Manager, SSL Architecture , Aggregating Suite, Web Servers, FTP Servers

Tier2 ประกอบด้วย Application Servers

Tier3 ประกอบด้วย File Servers, Database Servers



รูปที่ 4-1 แสดงเครือข่ายต้นแบบเพื่อต่อต้านการโจมตีด้วย DDoS

4.2 คอมพิวเตอร์ชั้นของ N – Tier Architecture of DDoS Defending System

Tier 0

- อินเทอร์เน็ต (Internet)
- เราเตอร์ (Router) ทำหน้าที่ในการค้นหาเส้นทางและป้องกันการโจมตี
- แคชเซิร์ฟเวอร์ (Cache Server) ทำหน้าที่ในการช่วยส่งการติดต่อ
- ไฟร์วอลล์ (Firewall) เป็นตัวป้องกันการโจมตีจากภายนอกเครือข่าย Dual DMZ
- IDS และ NIDS เป็นระบบต่อต้านการโจมตี

Tier 1

- Traffic Management ตัวจัดการและควบคุมการติดต่อสื่อสารในตอนเริ่มต้น
- SSL Architecture เป็นโครงสร้างโปรโตคอลที่สำคัญในการให้ความปลอดภัยในการติดต่อสื่อสารผ่านเว็บ

- Aggregation Suite เป็นตัวเร่งช่วยในการติดต่อสื่อสารและใช้บริการต่างๆของคอมพิวเตอร์
- เว็บเซิร์ฟเวอร์ (Web Servers) เป็นเซิร์ฟเวอร์ที่ให้บริการอินเทอร์เน็ตแก่ผู้ใช้งานทั่วไป
- เอฟทีพี (FTP Servers) เป็นเซิร์ฟเวอร์ที่จัดเก็บไฟล์ต่างๆ เพื่อการจัดส่งได้อย่างรวดเร็ว

Tier 2

- คลัสเตอร์ (Cluster) เป็นกลุ่มของคอมพิวเตอร์ที่ใช้งานประมวลผลข้อมูล
- โหลดบาลานซิง (Load Balancing) เป็นการจัดการที่รองรับ Failover ได้ในกรณีที่เสียหาย
- แอปพลิเคชันเซิร์ฟเวอร์ (Application Servers) เป็นเซิร์ฟเวอร์ที่เป็นโปรแกรมที่ใช้งาน
- ไฟร์วอลล์ (Firewall) ป้องกันการโจมตีจากภายใน

Tier 3

- ไฟล์เซิร์ฟเวอร์ (File Servers) เป็นเซิร์ฟเวอร์ที่จัดเก็บไฟล์ข้อมูลของระบบเครือข่ายคอมพิวเตอร์ที่สำคัญ
- ดาต้าเบสเซิร์ฟเวอร์ (Database Servers) เป็นเซิร์ฟเวอร์ที่ใช้ในการจัดการด้านฐานข้อมูลของระบบคอมพิวเตอร์

4.3 เราเตอร์ (Router)

4.3.1 แนะนำ

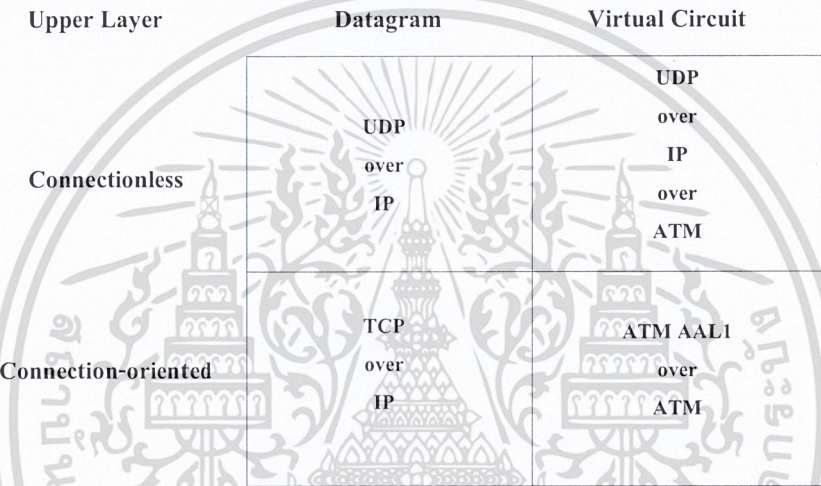
เป็นอุปกรณ์ฮาร์ดแวร์ที่สามารถทำงานได้เฉพาะอย่าง โดยมีความซับซ้อนมากใช้เป็นตัวเข้าถึงชั้นที่อยู่ของเน็ตเวิร์กและมีซอฟต์แวร์คอยจัดการหาเส้นทางระหว่างที่อยู่เหล่านั้นจึงเป็นการส่งผ่านโดยเฉพาะทำงานในชั้น ฟิสิคัล , ดาต้าลิงก์ และ เน็ตเวิร์ก ของ โมเดลโอเอสไอ โดยเราเตอร์จะค่อยๆส่งแพ็กเก็ตท่ามกลางการเชื่อมต่อเครือข่ายทั้งหลายเข้าด้วยกันจะทำการหาเส้นทางให้แพ็กเก็ตเกิดจากเครือข่ายหนึ่งไปยังเครือข่ายอื่นๆเปรียบเสมือนสถานีบนเครือข่าย สำหรับสมาชิกของเครือข่ายเท่านั้น ในตัวเราเตอร์จะเต็มไปด้วย (address) ไว้ และเชื่อมต่อกับเครือข่ายอื่นๆ ในเวลาเดียวกันหน้าที่ย่างง่ายที่สุดคือ รับแพ็กเก็ตเกิดจาก

เครือข่ายหนึ่งส่งไปยังอีกเครือข่าย อย่างไรก็ตามถ้ามีการรับแพ็กเก็ตที่ไม่ใช่สมาชิกเราเตอร์จะมีความสามารถกำหนดการเชื่อมต่อเครือข่ายที่ดีที่สุดให้กับแพ็กเก็ต

ทันทีที่มีการส่งแพ็กเก็ตออกไป เราเตอร์จะค้นหาเส้นทางในการส่งอย่างเหมาะสมไปยังเราเตอร์อีกตัว เราเตอร์จะตรวจสอบที่อยู่ปลายทาง (destination address) และค้นหาเส้นทางที่ดีที่สุดให้แพ็กเก็ตและส่งผ่านแพ็กเก็ตไปยังเครือข่ายปลายทางหรือไปยังอีกเครือข่าย

4.3.2 Routing Algorithm

ในเราเตอร์จะมี เรดต์ดิงอัลกอริทึม (routing algorithm) ซึ่งเป็นซอฟต์แวร์ที่ทำงานในชั้นเครือข่าย (Network Layer) รับผิดชอบตัดสินใจส่งออกแพ็กเก็ตที่เข้ามาไปยังเส้นทางที่เหมาะสม



ตารางที่ 4-1 แสดงชนิดของ subnet

โดยถ้าซับเน็ต (Subnet) ใช้ดาต้าแกรม (Datagram) ภายในจะตัดสินใจทุกครั้งทีแพ็กเก็ตข้อมูลเข้ามาด้วยว่ามีเส้นทางเปลี่ยนแปลงหาเส้นทางที่ดีที่สุดได้เปลี่ยนไป

ถ้าซับเน็ต (Subnet) ใช้ Virtual Circuits Routing จะทำแค่ดูว่ามีการเซตอัพ (Setup) Virtual Circuit เท่านั้น

คุณสมบัติ routing algorithm

- Correctness ความถูกต้องในการจัดส่ง
- Simplicity ความง่ายในการจัดการ
- Robustness ความคงทนของอัลกอริทึม (Algorithm)
- Stability ความมีเสถียรภาพ
- Fairness ความเท่าเทียมในการจัดรับ/ส่งแพ็กเก็ต

- Optimality ความขัดแย้งในการตัดสินใจน้อยที่สุดโดยไม่ขึ้นกับโทโพโลยี (Topology) และการใช้งานเครือข่าย (Traffic) จะมีการสร้างซิงค์ทรี (sink tree) เมื่อช่วยในการตัดสินใจ และป้องกันการเกิดลูปไม่จบสิ้น (Infinite Loop) โดยจำกัดจำนวนการค้นหาลำเส้นทาง (Route) ผ่านฮอป (hop)

4.3.3 ประเภทของ Routing Algorithm

Routing Algorithm แบ่งเป็น 2 ประเภทใหญ่ๆ คือ

- 4.3.3.1. Non Adaptive Algorithm ไม่สามารถปรับเปลี่ยนได้บางครั้งเรียกว่า “Static Routing”
- 4.3.3.2. Adaptive Algorithm สามารถปรับเปลี่ยนได้

4.3.3.1 Non adaptive algorithm

โดยมีอัลกอริทึมที่น่าสนใจดังนี้

4.3.3.1.1 Shortest Path Routing

มีการสร้างกราฟของชั้นเน็ตขึ้นมา แต่ละโหนดของ กราฟ , เราเตอร์ และแต่ละอาร์ค (arc) ของ กราฟแทนเส้นทางเชื่อมต่อการติดต่อสื่อสาร (บางครั้งเรียก ลิงค์ (link)) โดยจะเลือกหาเส้นทางที่เป็นคู่ของ เราเตอร์ในอัลกอริทึมจะหาเส้นทางที่สั้นที่สุดในระหว่างมันบนกราฟ

ค่า weight ของกราฟจะขึ้นอยู่กับระยะทาง (distance) , ช่องว่างการสื่อสาร (bandwidth) , ค่าเฉลี่ย การใช้งาน (average traffic) , ค่าใช้จ่ายในการติดต่อสื่อสาร (communication cost) , ค่าเฉลี่ยความยาวของ คิว (mean queue length) , ค่าความล่าช้า (measured delay) และองค์ประกอบอื่นๆ

4.3.3.1.2 Flooding

คือ แพ็กเก็ตจะถูกส่งออกทุกเส้นทางยกเว้นทางที่รับเข้ามา จึงทำให้เกิดความคับคั่งของการใช้ งานสายสัญญาณ (traffic) อย่างมากมายและไม่มีที่สิ้นสุด แต่จะใช้การนับฮอป (hop) ที่แพ็กเก็ตวิ่งผ่าน โดย ในเฮดเดอร์ของแพ็กเก็ตจะถูกลดลงทีละหนึ่งเมื่อผ่าน 1 ฮอป (hop) และแพ็กเก็ตจะไม่ถูกส่งต่อไปอีกค่า เคา์เตอร์ (counter) เป็น 0 อีกวิธีการคือ จะไม่ส่งแพ็กเก็ตเมื่อมีการ flood เป็นครั้งที่สองซ้ำโดยจะใส่ sequence number ของเราเตอร์ไว้ในแพ็กเก็ตที่รับจากโฮสต์ โดยเราเตอร์แต่ละตัวจะมีรายชื่อของ source router เพื่อบอก sequence number เริ่มต้น ถ้าแพ็กเก็ตที่เข้ามาอยู่ในลิสต์จะไม่ถูกส่งออกไปอีก และได้มี selective flooding algorithm คือเราเตอร์ไม่ส่งทุกๆแพ็กเก็ตที่เข้ามาไปยังทุกสายสัญญาณเฉพาะสายที่ถูก ส่งไปยังทิศทางที่ถูกต้องเท่านั้นที่ถูกส่งออกไป

4.3.3.1.3 Flow-Based Routing

ถ้ารู้ค่าเฉพาะค่าเฉลี่ยของโฟลว์ (flow) จะสามารถคำนวณค่าเฉลี่ยของแพ็กเก็ต ที่เข้าที่อยู่บนสาย สัญญาณได้จากทฤษฎี Queueing จากค่าเฉลี่ยค่าเข้าของทุกๆสายสัญญาณจะเอาไปหาค่า Flow Weight เพื่อให้ได้ค่าเฉลี่ยแพ็กเก็ตค่าเข้าของทั้งซับเน็ต (Subnet) โดยจะต้องรู้ค่าซับโทโพโลยี (Subtopology) , เมตริก ของเครือข่าย (Traffic Matrix) , เมตริกของค่าเฉพาะของสายสัญญาณ (Line capacity matrix) และอัลกอริ ทึมการค้นหาลำเส้นทาง (Routing Algorithm) ที่ใช้

4.3.3.2 Adaptive algorithm

4.3.3.2.1 Distance Vector Routing (Bellman-Ford)

โดยทำงานโดยแต่ละเราเตอร์ (Router) จะมีตารางเวกเตอร์ (Table Vector) ที่รู้ค่าระยะทาง (distance) แต่ละปลายทาง (destination) และเส้นทางที่ใช้ โดยตาราง (Table) จะถูกเปลี่ยนแปลงโดยการแลกเปลี่ยนข้อมูลกับเราเตอร์ (Router) ข้างๆ บางครั้งเรียกว่า “Bellman-Ford Algorithm” หรือ “Ford-Fulkerson Algorithm” ซึ่งเราเตอร์ (Router) จะมีอินเด็กซ์ของตารางการค้นหาเส้นทาง (Routing Table Index) และประกอบด้วยทางเข้า (Entry) ของแต่ละเราเตอร์ (Router) ในซับเน็ต (Subnet) โดยทางเข้า (Entry) ประกอบด้วยทางออกของสายสัญญาณไปยังปลายทาง (Destination) และค่าประมาณเวลา หรือ ระยะทางไปยังปลายทาง (Destination) โดยตารางเมตริก (Metric Table) จะใช้จำนวนของฮอป (hop) , เวลาล่าช้าเป็นวินาที , จำนวนแพ็กเก็ตในคิวตามเส้นทาง โดยจะสมมุติเราเตอร์รู้ระยะทาง (Distance) ไปยังเราเตอร์ข้างๆ

ค่าองค์ประกอบอื่นๆที่ต้องพิจารณา คือ

- ปัญหาการ Count_to_Infinity
- การ Split Horizon Hack

4.3.3.2.2 Link State Routing

สิ่งที่ต้องพิจารณา คือ

1. ค้นหา เราเตอร์ข้างๆและเรียนรู้ที่อยู่ของเครือข่าย (Network Address)
2. วัดความล่าช้าและค่าใช้จ่ายของเราเตอร์นั้นๆ
3. สร้างแพ็กเก็ตเพื่อบอกไปยังเราเตอร์ที่ต้องการเรียนรู้
4. ส่งแพ็กเก็ตไปยังเราเตอร์อื่นๆ
5. คำนวณเส้นทางที่สั้นที่สุดไปยังเราเตอร์เหล่านั้น

4.3.3.2.3 Hierarchical Routing

เมื่อเน็ตเวิร์กมีขนาดใหญ่ขึ้นเรื่อยๆจึงมีแนวคิดแบ่งเป็นระดับของเราเตอร์ โดยกำหนดขอบเขต (region) ให้กับเราเตอร์โดยเราเตอร์จะรู้จักรายละเอียดของทุกเราเตอร์ที่จะหาเส้นทางให้แพ็กเก็ตไปยังปลายทางในเฉพาะขอบเขตเดียวกันแต่ไม่รู้โครงสร้างภายในแต่ละขอบเขตนั้น แต่ละเวลาติดต่อกับเราเตอร์นอกขอบเขตจะผ่านเราเตอร์ที่จะส่งออกเท่านั้น

4.3.3.2.4 Broadcast Routing

คือ ต้องการส่งแพ็กเก็ตไปยังโฮสต์ที่มากกว่าหนึ่งหรือทั้งหมดมีปัญหาเดียวกับการ flooding algorithm คือ จะส่งแพ็กเก็ต ออกไปจำนวนมากและกินช่องว่างการสื่อสาร (Bandwidth) มากและมี multi destination routing คือ แต่ละแพ็กเก็ตจะมีรายชื่อของปลายทางอยู่ในคิว เมื่อเราเตอร์รับเข้ามาจะตรวจสอบปลายทางทั้งหมด และกำหนดเป็นกลุ่มของแพ็กเก็ตที่จะถูกส่งต่อโดยเราเตอร์ จะสร้างแพ็กเก็ตใหม่ที่มีเฉพาะเป้าหมายเดียวกันไปยังปลายทางของแต่ละตัวไป

วิธีการในการ Broadcast Algorithm

ใช้ spanning tree โดยแต่ละเราเตอร์จะส่งแพ็กเก็ตออกไปเพื่อค้นหาเราเตอร์จะเป็นราก (root) เมื่อกำลังหาเส้นทางของซับเน็ต (Subnet) บางครั้งเรียก “Reverse Path Forwarding”

4.3.3.2.5 Multicast Routing

เป็นการส่งแพ็กเก็ตไปยังเฉพาะกลุ่มเดียวกันเรียกว่า “Multicasting” และอัลกอริทึมการค้นหาเส้นทาง (Routing Algorithm) คือ “Multicast Routing” วิธีการเราเตอร์แต่ละตัวมาคำนวณหา Spanning Tree ครอบคลุมไปยังเราเตอร์ทุกๆ ตัวในซับเน็ต (Subnet) ส่ง Multicast packet ไปยังกลุ่มเราเตอร์ตัวแรกจะตรวจสอบ Spanning Tree เวลาจัดเรียงใหม่แล้วเราเตอร์ที่ไม่อยู่ในกลุ่มออกไป การจัดเรียงจะใช้อัลกอริทึมอื่นมาช่วย อาทิ เช่น Link State Routing

ข้อเสีย คือ ขยายแพ็กเก็ตได้ยาก บางครั้งใช้ “Core-Base Trees” ในการกระทำ Spanning Tree ของกลุ่มโดยจะหา root เป็นแกนกลางแทน

4.4 ไฟร์วอลล์ (Firewall)

4.4.1 แนะนำ

ทำหน้าที่ ปกป้องระบบหรือ เครือข่ายจากการโจมตีผ่านเครือข่าย และในเวลาเดียวกันจะต้องป้องกันการเข้าใช้งานจากเครือข่ายภายนอกผ่าน WAN หรือ อินเทอร์เน็ต โดย ไฟร์วอลล์ จะต้องมีความสามารถ

1. การใช้งานทั้งภายในและภายนอกและกรณีอื่นๆจะต้องผ่านไฟร์วอลล์
2. เฉพาะการใช้งานที่แสดงสิทธิ (Authorize) ที่กำหนดโดยนโยบายความปลอดภัยจึงจะอนุญาตให้เข้าใช้งานได้
3. ไฟร์วอลล์จะต้องมีความสามารถป้องกันการเจาะระบบรวมถึงป้องกันระบบปฏิบัติการ (Operation System) ด้วย

4.4.2 เทคนิคที่ใช้งาน

ในการควบคุมการเข้าใช้งานแล้วเพิ่มประสิทธิภาพให้กับนโยบายความปลอดภัยของเว็บไซต์ต่างๆ

Service Control

กำหนดชนิดของบริการอินเทอร์เน็ตให้สามารถใช้งานทั้งจากภายในและภายนอก โดยจะทำ filter traffic บนที่อยู่ไอพี (IP Address) ทางพอร์ตที่ซีพีอาจจะมียอฟแวร์หรือซีพีที่จะคอยรับและแปลงการขอบริการ แต่ละประเภทก่อนจัดส่ง หรืออาจจะเป็นซอฟต์แวร์โฮสต์เซิร์ฟเวอร์เช่น เว็บ (Web) หรือ เมล์ (Mails)

4.4.2.1 Direction Control

กำหนดทิศทางของการร้องขอบริการ อาจะกำหนดค่าเริ่มต้นและอนุญาตให้ทำงานผ่านไฟร์วอลล์

4.4.2.2 User Control

ควบคุมการเข้าใช้บริการตามสิทธิของผู้ใช้งาน (Users) กำหนดการใช้งานได้

4.4.2.3 Behavior Control

ควบคุมพฤติกรรมของผู้ใช้งานว่า มีการเข้าใช้งานบริการ โดยเฉพาะที่กำหนดเท่านั้น

4.4.3 ความสามารถของไฟร์วอลล์

1. ไฟร์วอลล์ กำหนดให้ยูติจิตี้ที่ทำให้เกิดจากผู้ใช้งานไม่แสดงตนจากการป้องกันเครือข่าย , ตรวจสอบเซ็นสิติการเข้าใช้บริการเข้า/ออกจากเครือข่าย และยังสามารถป้องกัน การทำ IP spoofing และการโจมตีเส้นทางได้ (Routine Attack)

2. ไฟร์วอลล์ จะต้องตรวจสอบเหตุการณ์ความปลอดภัยต่างๆ สังเกตความเปลี่ยนแปลง และแจ้งไปยังระบบ Firewall

3. จะต้องมีความสามารถป้องกันได้หลาย Platform เพื่อสะดวกตามความหลากหลายของอินเทอร์เน็ตที่ไม่มีความปลอดภัยได้

4. ต้องรองรับ IPSec (IP Security) ได้

4.4.4 ข้อจำกัดของไฟร์วอลล์

1. ไม่สามารถป้องกันและตอบโต้การโจมตีที่ไม่ผ่านไฟร์วอลล์ได้

2. ไม่สามารถป้องกันการโจมตีจากภายในได้

3. ไม่สามารถป้องกัน โปรแกรมไวรัส ที่ทำให้เกิดการติดเชื้อหรือไฟล์ของระบบ (File System) ได้

4.4.5 ชนิดของไฟร์วอลล์

แบ่งออกเป็น 4 ชนิดคือ

4.4.5.1 Packet Filter

4.4.5.2 Application Level Gateway

4.4.5.3 Circuit level Gateway

4.4.5.4 Stateful Inspection (Dynamic packet filtering)

4.4.5.5 Cache Server

4.4.5.1 Packet Filter Router

มีการนำไปใช้กับข้อกำหนดของกฎ (Rules) เกี่ยวกับการรับแพ็กเก็ตเข้ามาและจะทำการส่งต่อ (Forward) หรือยกเลิก (Discard) แพ็กเก็ตนั้น

จะใช้ทั้งสองทิศทางคือทั้งเข้าและออกจากเครือข่าย โดยกฎจะมีทั้งไอฟีและทรานสปอร์ต (TCP,UDP,Header) โดยแหล่งที่อยู่ (Source) และที่อยู่ปลายทาง (Destination IP Address) จะใช้ในฟิลด์ของโปรโตคอลไอฟี (IP Protocol) และหมายเลขพอร์ต ที่ซีพี, ยูติพี ซึ่งกำหนดใช้โดยโปรแกรม

นโยบาย 2 ข้อตามปกติ คือ

- Default = discard

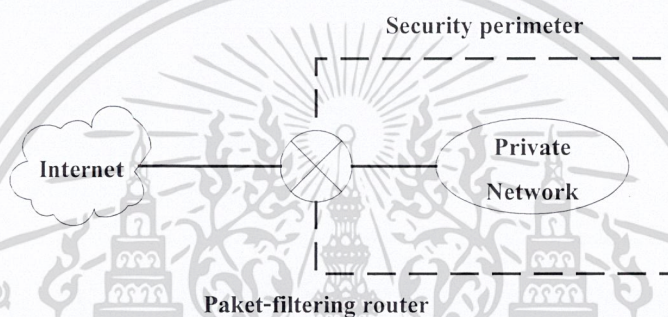
- Default = forward

ข้อดี คือ ง่ายต่อการกำหนดให้เราเตอร์

ข้อเสีย คือ ยากในการเช็ตอัปเดต (Rules) ในการกรองแพ็กเก็ต และทำให้เกิดความล่าช้าในการแสดงตน

การโจมตีที่มีบน Packet Filtering Router

- IP Address Spoofing
- Source Routing Attack
- Tiny Fragment Attack



รูปที่ 4-2 แสดง Packet-filtering router

4.4.5.2 Application Level Gateway

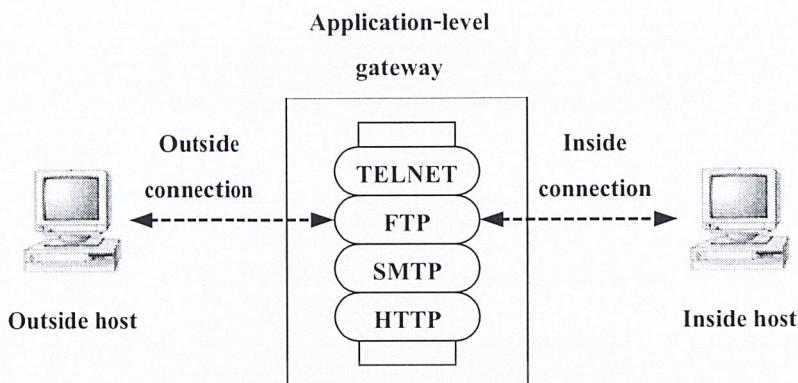
บางครั้งเรียก “Proxy server” กระทำการลดความคับคั่งการใช้งานในระดับโปรแกรมให้ช้าลง พยายามทำการป้องกันในระดับของโปรแกรม (Application) อาทิ เช่น TELNET , FTP , SMTP , HTTP และ เกตเวย์ จะทำตามผู้ใช้งานสำหรับชื่อของ โฮสต์ปลายทาง (Remote Host) ที่จะเข้าใช้งาน โดยการตอบกลับของผู้ใช้งานจะต้องมี ไอดีของผู้ใช้งาน (User ID) และ ข้อมูลการแสดงตน แล้วเกตเวย์จะติดต่อไปยัง โปรแกรมของโฮสต์ปลายทาง (Remote Host) และใช้เซ็กเมนต์ของทีซีพีในโปรแกรมข้อมูลระหว่าง 2 จุดได้

ถ้าเกตเวย์ไม่สามารถปรับเป็นพรีอ็อกซีโค้ด (Proxy Code) ให้ใช้กับโปรแกรมได้ จะไม่สามารถใช้บริการนั้น และจะไม่ผ่านไฟร์วอลล์ไปได้

โดยเกตเวย์สามารถกำหนดคุณลักษณะให้ใช้งานคุณลักษณะเฉพาะในโปรแกรมที่ โดยผู้ดูแลระบบเครือข่าย (Network Administrator) พิจารณาให้ใช้งานได้

พยายามที่จะทำให้ปลอดภัยมากกว่าการใช้แพ็กเก็ตไฟเตอร์เพื่ออนุญาต หรือซ่อนไว้ในระดับที่ซีพี และ ไอพี โดยจะตรวจสอบให้ถี่ถ้วนในระดับของโปรแกรมส่วนที่เพิ่มเติมขึ้นมาคือ ง่ายในการตรวจสอบและเปรียบเทียบการใช้งาน (Traffic) ที่เข้ามาทั้งหมดที่ระดับโปรแกรม (Application)

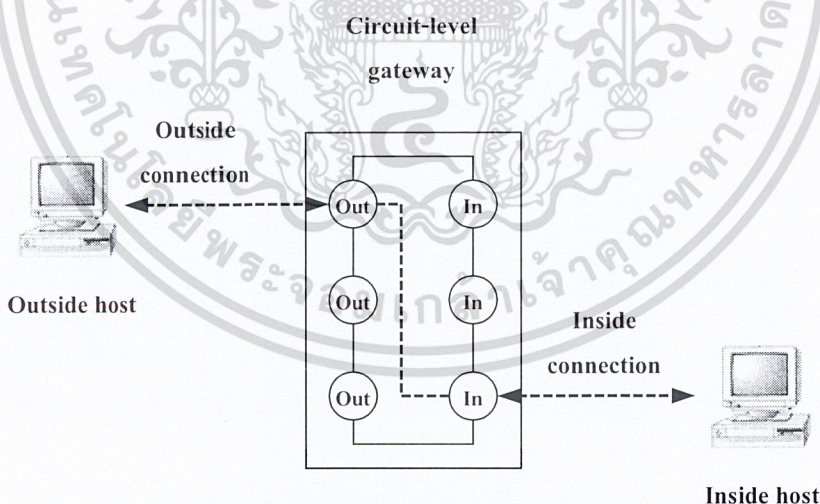
ข้อเสีย คือ เกตเวย์ จะมีกระบวนการตรวจสอบเพิ่มขึ้นสำหรับแต่ละการติดต่อ



รูปที่ 4-3 แสดง Application-level gateway

4.4.5.3 Circuit Level Gateway

เป็นการทำงานทั้งในระบบ Stand Alone และหน้าที่เฉพาะอย่างในระดับโปรแกรมใช้โดยไม่อนุญาตให้ใช้การติดต่อ ที่ซีพี แบบ End-to-End โดยเกตเวย์จะเชื่อมต่อการติดต่อที่ซีพี 2 อย่าง คือ ระหว่างตัวมันเองกับผู้ใช้ภายในโฮสต์ และ ตัวมันเองกับผู้ใช้ภายนอกโฮสต์ การใช้งานคือ ผู้ดูแลระบบ (System Administrator) ใช้ Application Level Gateway หรือ บริการพร็อกซี (Proxy) เชื่อมต่อภายใน และใช้ Circuit Level ในการเชื่อมต่อออกสู่ภายนอก



รูปที่ 4-4 แสดง Circuit-level gateway

4.4.5.4 Stateful Inspection (Dynamic packet filtering)

คำว่า สเตตฟูลอินสเปกชัน (Stateful Inspecton) หรือ ไดนามิกแพ็กเก็ตฟิลเตอร์ริง (Dynamic packet filtering) เป็นไฟร์วอลล์แบบแพ็กเก็ตฟิลเตอร์ริง จะตัดสินใจโดยดูข้อมูลในส่วนของเฮดเดอร์เท่านั้น

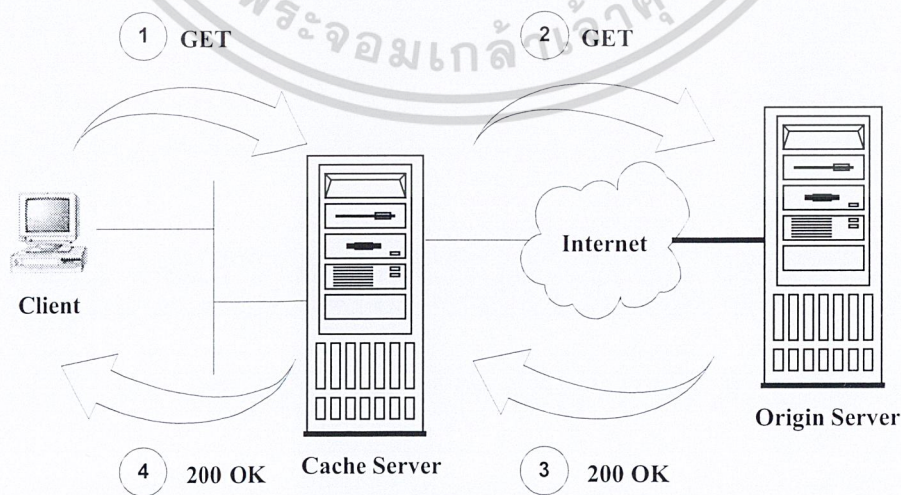
และจะพิจารณาเฉพาะแพ็กเก็ตนั้นๆ โดยไม่ได้คำนึงถึงแพ็กเก็ตก่อนหน้านี้ ส่วนสเตตฟูลอินสเปกชันจะเอามาแก้ไขจាកนี้ ทั้งในส่วนของคุณสมบัติที่ใช้ในการตัดสินใจ โดยจะดูถึงข้อมูลในส่วนของ Payload ด้วย และจะคำนึงถึงในส่วนของแพ็กเก็ตอื่นๆ ที่อยู่ก่อนหน้านี้ประกอบในการตัดสินใจ ทำให้ความสามารถในการตัดสินใจมากขึ้น เพราะรู้ข้อมูลมากขึ้น ซึ่ง สเตตฟูลอินสเปกชันก็เหมือนกับแพ็กเก็ตฟิเตอร์ริง ก็ถูกเพิ่มลงในเราเตอร์ ดังนั้นโอสต์ที่มีฟังก์ชันของสเตตฟูลอินสเปกชันทำงานอยู่นั้นจะต้องเป็นหรือสามารถทำงานเสมือนเป็นเราเตอร์ได้ด้วย

การทำงานของไฟร์วอลล์ชนิดสเตตฟูลอินสเปกชันจะมีการติดตาม State การทำงานของการเชื่อมต่อแบบทีซีพี ซึ่งเป็นผลให้สามารถสรุปแบบการทำงานได้ทั้งกระบวนการ ไม่ได้ดูเพียงข้อมูลในแต่ละแพ็กเก็ต โดยสามารถดูลักษณะการเชื่อมต่อ การโต้ตอบของแต่ละโพรโตคอลที่มีลักษณะที่แตกต่างกัน โดยสามารถแยกแยะโพรโตคอลที่ถูกต้องกับโพรโตคอลที่ไม่ถูกต้องออกจากกันได้ นอกจากนี้ไฟร์วอลล์ชนิดสเตตฟูลอินสเปกชันยังมีความปลอดภัยมากกว่า เพราะสามารถจะปิดพอร์ตที่มีหมายเลขมากกว่า 1,024 ได้ เนื่องจากในการเชื่อมต่อแบบทีซีพี อย่างเช่น เว็บนั่น แม้เมื่อเริ่มแรกจะติดต่อกันโดยผ่านพอร์ต 80 แต่หลังจากที่ติดต่อกันแล้ว จะมีการใช้หมายเลขพอร์ตในลักษณะสุ่ม โดยมีหมายเลขมากกว่า 1,024 ซึ่งทำให้ไฟร์วอลล์ชนิดแพ็กเก็ตฟิเตอร์ริงจำเป็นต้องเปิดพอร์ตที่มีหมายเลขมากกว่า 1,024 เอาไว้ตลอดเวลา แต่ไฟร์วอลล์แบบสเตตฟูลอินสเปกชันจะเปิดพอร์ตเฉพาะเวลาที่มีการเชื่อมต่อผ่านพอร์ตนั้นๆ เท่านั้น ทำให้ปลอดภัยมากกว่า

4.4.5.5 Cache Server

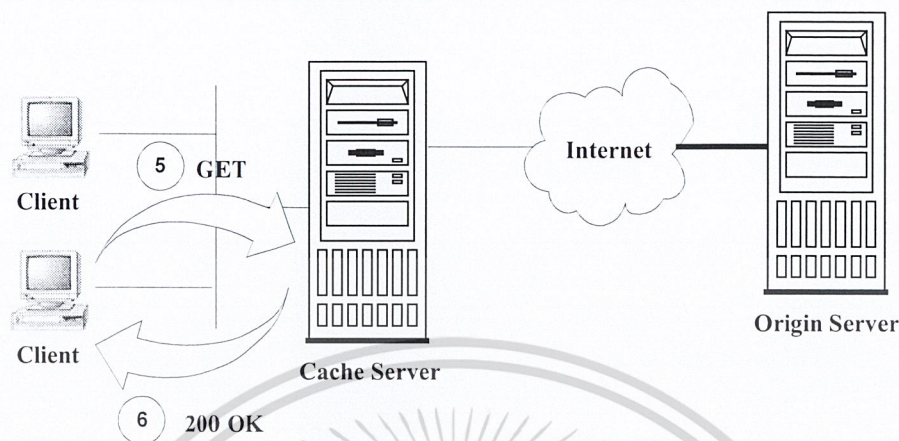
เป็นพร็อกซีเซิร์ฟเวอร์ชนิดพิเศษ ซึ่งใช้ในการช่วยเพิ่มประสิทธิภาพให้กับระบบเซิร์ฟเวอร์ที่ให้บริการงานด้านต่างๆ อาทิ เช่น เว็บเซิร์ฟเวอร์

ซึ่งจะมีการจดจำออบเจกต์ (object) ที่ถูกร้องขอโดยเครื่องไคลเอ็นต์ และถ้ามีการร้องขอออบเจกต์เดิมซ้ำ (ทั้งจากเครื่องไคลเอ็นต์เดิม และเครื่องไคลเอ็นต์อื่นๆ) ออบเจกต์นั้นจะถูกส่งกลับไปแทนที่จะร้องขอจากเซิร์ฟเวอร์หลัก ดังรูปข้างล่าง



รูปที่ 4-5 แสดงการทำงานของ cache server

จากรูป แสดงหน้าที่ปกติของพร็อกซีโดยคีย์หลัก (key) ของการจดจำออบเจกต์นั้น ด้วยการก๊อปปี้ไว้ใน ดิสก์ในเครื่อง หรือหน่วยความจำ



รูปที่ 4-6 แสดงเมื่อมีการร้องขอออบเจกต์เดิมจากเครื่องไคลเอนต์

จากรูป มีการร้องขอออบเจกต์เดิมจากเครื่องไคลเอนต์อื่นๆแคชเซิร์ฟเวอร์ไม่จำเป็นต้องไปร้องขอซ้ำจาก origin servers ซ้ำ โดยแคชเซิร์ฟเวอร์จะช่วยปรับปรุงประสิทธิภาพ เครื่องไคลเอนต์และเครื่องเซิร์ฟเวอร์ที่ให้บริการ ด้านเครื่องไคลเอนต์ จะใช้ระยะเวลาในการร้องขอน้อยลง ส่วนเซิร์ฟเวอร์จะลดภาระงานที่ต้องรับ โดยใช้ช่องว่างการสื่อสารลดลง

4.4.6 ชนิดของพร็อกซีเซิร์ฟเวอร์ (Proxy Servers)

4.4.6.1 Generic Firewall Proxies

4.4.6.2 Departmental Proxies

4.4.6.3 Personal Proxies

4.4.3.4 Specialized Proxies

4.4.3.4.1 พร็อกซี (Proxies) ระหว่างไคลเอนต์ (Client) และพร็อกซี (Proxies) อื่นๆ

4.4.3.4.2 พร็อกซี (Proxies) ทำหน้าที่ แปลงรูปแบบ

4.4.4.5 Reverse proxies

4.4.6.1 Generic Firewall Proxies

หมายถึง พร็อกซี (Proxies) ทั่วๆ ไปทำหน้าที่เป็น Web Traffic รวมทั้งโพรโตคอล HTTP , FTP และ Gopher มีโพรโตคอล เพื่อความปลอดภัยให้กับเว็บ มีการใช้ SSL อาทิ เช่น HTTPS และ SNEWS

- ควบคุมการเข้าใช้งาน (Access Control)

- การกรอง (Filtering)

- การล็อก (Logging)

- การทำแคช (Caching)

4.4.6.2 Departmental Proxy Server

คล้ายกับไฟร์วอลล์พร็อกซีที่เพียงแค่ดูแลเฉพาะแผนก (Department) หรือส่วนย่อยๆ ในบริษัทหรือองค์กรขนาดใหญ่

4.4.6.3 Personal Proxy Server

บางครั้งเรียก “Trimmed_down Proxy Server” แบ่งตามจุดในเฉพาะบุคคลเท่านั้น คุณลักษณะแตกต่างจากโปรแกรมอื่นๆ คือ

- การทำ Local Caching
- การเปลี่ยนแปลง Active Cache
- การเปลี่ยนแปลง Polling
- การทำ Notification

4.4.6.4 Specialized Proxy Server

เป็น พร็อกซี (Proxies) ใช้งานเฉพาะอย่าง

4.4.6.4.1 Specialized Proxy Server ระหว่างพร็อกซี (Proxies) กับไคลเอนต์(Client)

มีจุดประสงค์ใช้งานเฉพาะมากขึ้น ส่วนใหญ่จะติดตั้งก่อนเข้าพร็อกซีจริง เมื่อช่วยทำงานฟิเตอร์แพ็กเก็ตที่ต้องการ อาทิ เช่น Java applet หรือการค้นหาไวรัส (Virus screening)

4.4.6.4.2 ช่วยเพิ่มประสิทธิภาพในการติดต่อ (Accelerator)

ติดตั้งก่อนเข้าพร็อกซี (Proxies) จริง จุดประสงค์เพื่อช่วยในการเข้าใช้งานหรือให้บริการได้เร็วขึ้นเพราะ เว็บเซิร์ฟเวอร์ (Web Server) จะทำงานจริง แต่แคชซิง (Caching) จะช่วยในการเข้าใช้งานได้เร็วกว่าไอโอ (I/O) มาก ซึ่งจะช่วยให้เว็บเซิร์ฟเวอร์ (Web Server) ช้า แต่เว็บเซิร์ฟเวอร์ (Web Server) มักจะมีคุณลักษณะหน้าที่นี้อยู่บ้าง

4.4.6.5 Reverse Proxy Server

ทำหน้าที่ เหมือนกับพร็อกซี (Proxies) ทั่วไปแต่มีคุณลักษณะพิเศษ คือ

1. Replication ไปยังพื้นที่ต่างกันที่กระจายตัวกันอยู่ ในบริษัทที่มีการกระจายตัวอยู่ในที่ต่างๆ กัน รีเวิร์สพร็อกซี (Reverse Proxy) ที่เสมือนเป็นบริษัทเดียวกันที่เข้าใช้งานได้อย่างปกติ
2. Replication สำหรับโหลดบาลานซิง (Load Balancing) ในโหลดบาลานซิง (Balancing) ช่วยรับภาระโหลดของเว็บเซิร์ฟเวอร์ (Web Server)

ส่วนประกอบในการติดตั้งรีเวิร์สพร็อกซีเซิร์ฟเวอร์ (Reverse Proxy Server)

- Request URL Remapping
- Request Header Remapping
- Response Header Remapping
- Content Remapping
- Virtual Multihosting ทำเสมือนการใช้งานเซิร์ฟเวอร์ตัวเดียวจาก เซิร์ฟเวอร์จริงหลายๆตัว

ความปลอดภัยของ รีเวิร์สพร็อกซีเซิร์ฟเวอร์ (Reverse Proxy Server)

- การทำ Certificate-Based Client Authentication อยู่บนพื้นฐานของ SSL
- การทำ HTTP Authentication มีการทำ Masquerad ของเว็บเซิร์ฟเวอร์ (Web Server) จึงต้องมีการแสดงสิทธิ์ (Authentication) โดยมีรีเวิร์สพร็อกซี (Reverse Proxy) เว็บเซิร์ฟเวอร์ (Web Server) เป็นตัวจัดการการแสดงผล (Authentication) ไม่ใช่พร็อกซีเซิร์ฟเวอร์ (Proxies Server) เป็นตัวจัดการ

4.4.6.6 การออกแบบระบบไฟร์วอลล์ (Firewall Configuration)

ในการออกแบบไฟร์วอลล์นั้น มีรูปแบบที่หลากหลาย โดยการออกแบบควรคำนึงถึงลักษณะของระบบเครือข่าย แต่โดยทั่วไป สามารถสรุปเป็นรูปแบบหลักๆ ที่มักจะเห็นกันทั่วไป ได้ 3 แบบ

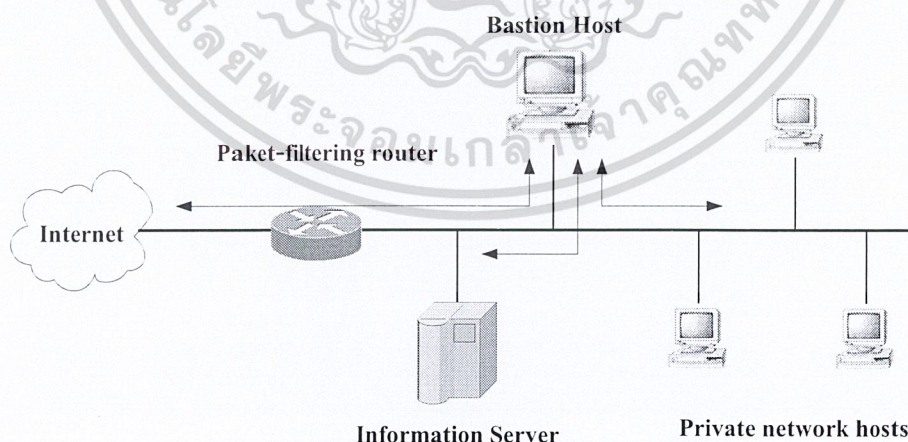
4.4.6.6.1 Screened host firewall (Single Homed Bastion Host)

4.4.6.6.2 Screened host firewall system (dual-homed bastion host)

4.4.6.6.3 Screened-subnet firewall system

4.4.6.6.1 Screened Host Firewall (Single Homed Bastion Host)

ระบบจะประกอบด้วย 2 ส่วน คือ ประกอบด้วยไฟร์วอลล์ชนิดแพ็กเก็ตฟิลเตอร์ริง (Packet Filtering) และประกอบด้วย Bastion Host โดย Bastion Host คือ เครื่องคอมพิวเตอร์พิเศษที่กำหนดให้เป็นจุดเชื่อมต่อกับภายนอก ดังนั้น จึงต้องเป็นเครื่องที่มีการดูแลเป็นอย่างดี และมีความแข็งแกร่งมากกว่าเครื่องอื่นๆ โดยอาจเป็นเครื่องที่มีระบบปฏิบัติการพิเศษก็ได้ โดย Bastion Host จะทำหน้าที่เป็น พร็อกซี หรือ เกตเวย์ ที่จะส่งต่อไปยังเครื่องอื่นๆ ในเครือข่าย ดังนั้นหากมีบริการใดๆ ในระบบ เครื่อง Bastion Host ก็จะต้องรู้จักโปรโตคอลนั้นด้วย นอกจากนั้นใน Bastion Host ยังต้องมีระบบ Authentication ที่เหมาะสม โดยเมื่อเข้าใช้บริการแต่ละบริการแล้ว อาจมีการ Authentication เพิ่มเติมอีกก็ได้

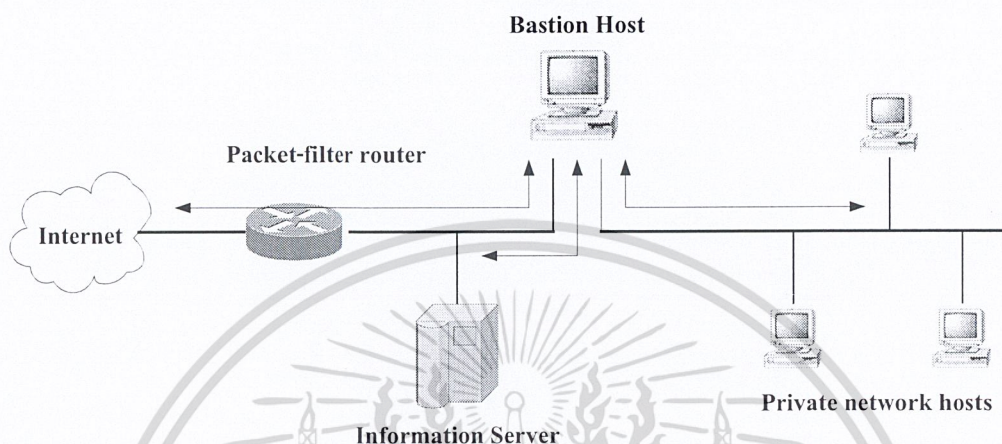


รูปที่ 4-7 แสดง Screened host firewall system (single-homed bastion host)

ดังนั้นในรูปข้างบน จะเห็นว่าในการเข้าใช้งาน จะต้องเริ่มจากไป Authentication กับ Bastion Host ก่อน โดยไฟร์วอลล์ จะเปิดให้เฉพาะการติดต่อกับ Bastion Host เท่านั้น การติดต่อกับเครื่อง

คอมพิวเตอร์อื่นๆ จะกระทำมิได้ ดังนั้นการติดต่อกับเครื่องอื่นๆ ต้องกระทำผ่าน Bastion Host เสมอ วิธีนี้ หากไฟร์วอลล์มีความปลอดภัยเพียงพอ และ Bastion Host มีความปลอดภัยเพียงพอแล้ว ก็หมายความว่า การติดต่อกับเครื่องให้บริการ จะต้องผ่าน พร็อกซี หรือ เกตเวย์ เสมอ ซึ่งทำให้ปลอดภัยมากขึ้น

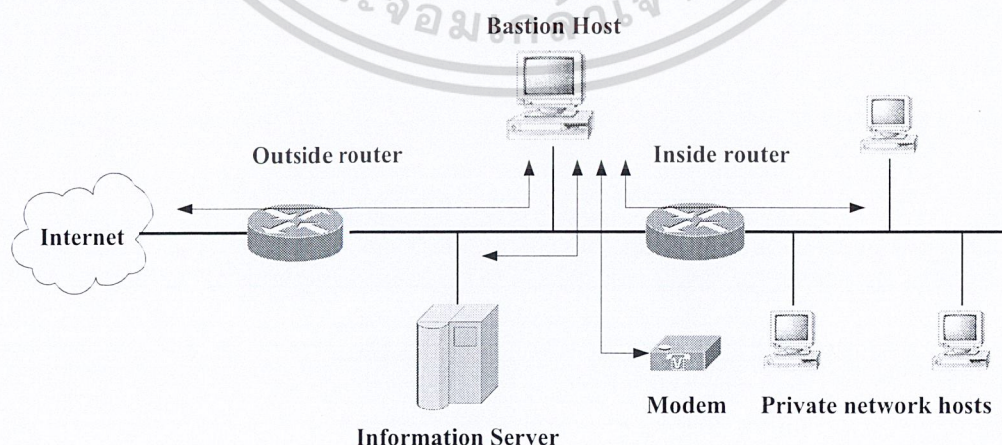
4.4.6.6.2 Screened Host Firewall (Dual Homed Bastion Host)



รูปที่ 4-8 แสดง Screened host firewall system (dual-homed bastion host)

ส่วนรูปนี้จะแตกต่างจากรูปก่อนที่ส่วนของ Bastion Host จะมีการแบ่งเครือข่ายภายในออกจากเครือข่ายที่ให้บริการ ซึ่งสามารถจะให้ความปลอดภัยมากกว่า ในกรณีที่ไฟร์วอลล์ถูกโจมตีจนทะลุผ่านมาได้ เพราะแม้ว่าไฟร์วอลล์จะไม่สามารถทำหน้าที่ได้ แต่การติดต่อกับเครื่องคอมพิวเตอร์ในเครือข่ายภายในนั้น ก็จะต้องกระทำผ่าน Bastion Host เสมือนกับมีด่านป้องกัน 2 ชั้น โดยจะโจมตีได้เฉพาะเครื่อง Information Server เท่านั้น

4.4.6.6.3 Screened-subnet Firewall

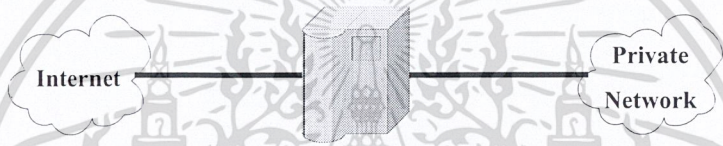


รูปที่ 4-9 แสดง Screened-subnet firewall system

จากรูปจะเห็นได้ว่า จะใช้ไฟร์วอลล์ 2 ตัว ในการเชื่อมต่อกับเครือข่าย โดยไฟร์วอลล์ตัวที่ 2 จะทำหน้าที่ป้องกันไม่ให้เครือข่ายภายในถูกโจมตี ในกรณีที่ไฟร์วอลล์ตัวนอก ถูกเจาะทะลุ นอกจากนั้น ยังช่วยป้องกัน ไม่ให้มีการโจมตีจากเครือข่ายภายในได้อีกด้วย เพราะแม้ว่าจะอยู่ในเครือข่ายภายใน แต่การเชื่อมต่อกับ Information Server ก็จะต้องกระทำผ่าน Bastion Host เท่านั้น

รูปแบบไฟร์วอลล์ทั้ง 3 รูปแบบที่ได้กล่าวมานั้น เป็นรูปแบบที่ถือได้ว่าเป็นมาตรฐาน และมีความปลอดภัยสูง อย่างไรก็ตาม เนื่องจากปัจจุบัน โปรแกรมไฟร์วอลล์ได้มีการพัฒนาไปมาก จนกระทั่งมีความสามารถต่างมากมายอย่างไฟร์วอลล์แบบ สเตตฟูลอินสเปกชันที่สามารถติดตามสเตต (State) ได้ ไฟร์วอลล์บางตัว มีความสามารถในการทำงานร่วมกับโปรแกรมป้องกันไวรัส โดยจะส่งไฟล์มาสแกนทุกครั้ง ไฟร์วอลล์บางตัวมีความสามารถในการทำงานร่วมกับโปรแกรมประเภท Intrusion Detection ทำให้รูปแบบข้างต้นไม่เป็นที่นิยมใช้

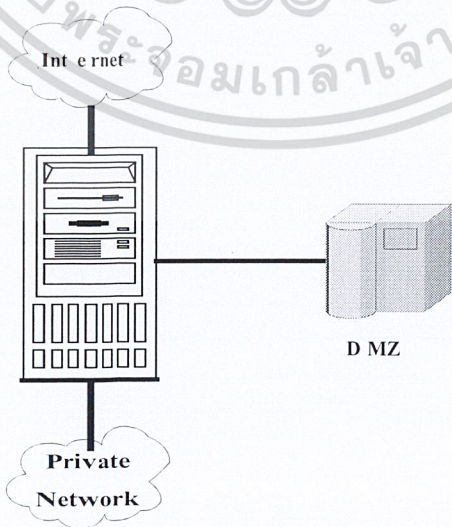
4.4.7 Basic border Firewall



รูปที่ 4-10 แสดง Basic Border Firewall (Single Layer)

รูปแบบไฟร์วอลล์แบบนี้ เป็นโฮสต์ตัวเดียวที่มีการเชื่อมต่อระหว่างเครือข่ายภายในขององค์กร และเครือข่ายภายนอก ในลักษณะแบบนี้ฟังก์ชันของไฟร์วอลล์ทั้งหมดจะถูกบรรจุอยู่ในโฮสต์ตัวเดียว ทำให้เมื่อมีการโจมตีไฟร์วอลล์ จะทำหน้าที่เป็นหน้าด่านเพียงหน้าด่านเดียวที่รับการโจมตี ดังนั้นหากทะลุไฟร์วอลล์มาได้ ก็จะสามารถเข้าถึงเครือข่ายภายในทั้งหมด

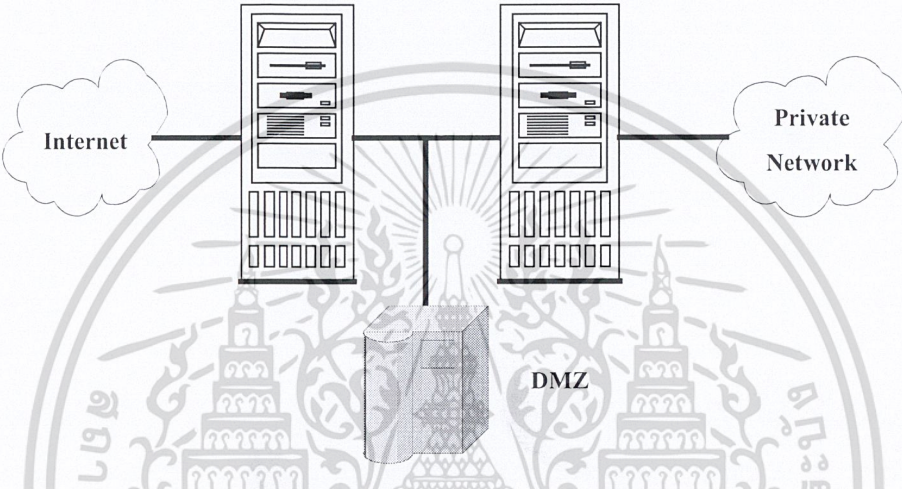
4.4.8 เครือข่าย DMZ



รูปที่ 4-11 แสดง Basic Border Firewall with DMZ Network

ในเครือข่าย DMZ (De-Militarize Zone) โฮสต์ที่ไม่น่าเชื่อถือ หรือโฮสต์ที่มีความปลอดภัยถูกนำไปไว้ภายในไฟร์วอลล์แต่ไม่ได้อยู่ในเครือข่ายภายใน โดยการทำเช่นนี้ไฟร์วอลล์จะทำการเชื่อมต่อกับสามเครือข่าย และโดยการทำเช่นนี้เป็นการเพิ่มประสิทธิภาพของ ความปลอดภัย ความน่าเชื่อถือ และความสามารถในการนำมาใช้งานของโฮสต์ที่ไม่น่าเชื่อถือ แต่ไม่ได้เพิ่มระดับของความปลอดภัยในการแอ็กเซสจากโฮสต์ที่อยู่ในภายในเครือข่าย โฮสต์ที่ไม่น่าเชื่อถือมีเพื่อจุดประสงค์ต่างกัน (ตัวอย่าง เช่น FTP Server และ Web Server) จะถูกนำมาไว้ในเครือข่าย DMZ เป็นการสร้างเครือข่ายบริการสาธารณะ

4.4.9 ดูอัลไฟร์วอลล์ (Dual Firewall)



รูปที่ 4-12 แสดง ดูอัลไฟร์วอลล์ (Dual Firewall)

เครือข่ายภายในขององค์กรจะกำหนดให้แยกกับเครือข่ายที่ไม่น่าเชื่อถือ โดยการเพิ่มไฟร์วอลล์โฮสต์ตัวที่สองลงไป โดยการเชื่อมต่อเครือข่ายที่ไม่น่าเชื่อถือกับไฟร์วอลล์ตัวแรก จะเรียกไฟร์วอลล์ตัวนี้ว่า ไฟร์วอลล์ภายนอก (Outer firewall) ส่วนการเชื่อมต่อเครือข่ายภายในองค์กรกับไฟร์วอลล์อีกตัวเรียกไฟร์วอลล์ตัวนี้ว่า ไฟร์วอลล์ภายใน (Inner Firewall) และเครือข่าย DMZ อยู่ระหว่างไฟร์วอลล์ทั้งสองทางเดินของข้อมูลระหว่างเครือข่ายภายใน และอินเทอร์เน็ตจะต้องผ่านมายังไฟร์วอลล์ทั้งสอง และเครือข่าย DMZ

4.5 ระบบต่อต้านการโจมตี (Intrusion Detection System) และ Network IDS

4.5.1 แนะนำ

ระบบต่อต้านการโจมตี บ่อยครั้งที่เรามีการพูดถึง Misuse Detection และ Intrusion Detection ข้อแตกต่างระหว่าง 2 อย่างคือ ถ้าเราพิจารณาวิเคราะห์ส่วนของเครือข่ายโดยสังเกตถึงกิจกรรมของผู้ใช้งาน น่าจะใช้ Misuse Detection แต่ถ้าสนใจการโจมตีจากภายนอกหรือผู้บุกรุกใช้ Intrusion Detection จะเหมาะสมกว่า

ข้อแตกต่างอีกอย่างที่น่าสนใจคือ ถ้าผู้ใช้มีสิทธิในเครือข่ายจะเป็น Misuse Detection แต่ถ้าไม่มีสิทธิเราใช้ Intrusion Detection

4.5.2 วิธีการตัดสินใจของลักษณะการบุกรุก

4.5.2.1 ใช้สถิติในการวิเคราะห์หาความผิดปกติ (Statistical Anomaly Detection)

จะทำการเก็บเป็นสถิติจากพฤติกรรมใช้งานปกติ โดยมีตัวแปรที่ควรสนใจ คือ ยูสเซอร์ , กรุป , เวอร์กสเตชัน , เซิร์ฟเวอร์ , ไฟล์ (File) , การ์ดเครือข่าย (Network Adapter) และอื่นๆ พื้นฐานคือ ถ้ามีการใช้งานผิดปกติไปจากเดิมนั่นเอง อาทิ เช่น เวลาที่เข้าใช้งาน โปรแกรมที่ใช้ เป็นต้น

ข้อดี คือ

- เข้าใจได้ง่ายอาศัยสถิติเข้าช่วย
- จำนวนของตัวแปรในรูปแบบที่ใช้ไม่มากนักทำให้ใช้หน่วยความจำในการจัดเก็บน้อย
- สถิติที่ได้ขึ้นอยู่กับเวลาโดยมีการหาค่าเฉลี่ย คำนวณหนัก และตัวแปรภายใน
- พฤติกรรมโดยรวมอย่างง่าย คือ failed login ทำให้ผู้ใช้งานเข้าใจง่าย

ข้อเสีย คือ

- ยังรวมไปถึงข้อมูลที่ไม่ต้องทำเป็นสถิติด้วย
- รวมค่าจากหลายตัวแปรอาจจะทำให้ได้สถิติไม่ถูกต้อง
- ไม่มีมาตรฐานที่แน่นอนรองรับ
- พฤติกรรมของผู้ใช้งานแต่ละคนไม่เหมือนกัน
- แอ็กเตอร์ผู้รู้สถิติจะเข้าใจและพยายามหลีกเลี่ยงการตรวจสอบและเปลี่ยนไปใช้วิธีการอื่นแทน
- ผู้โจมตีอาจจะให้หลาย account ในการแสดงพฤติกรรมที่แตกต่างหลีกเลี่ยงการตรวจจับได้
- ค่าเฉลี่ยเวลาที่ใช้ในการแสดงพฤติกรรมเพื่อโจมตีแตกต่างกัน

4.5.2.2 การตรวจสอบจากรูปแบบ (Pattern Matching Detection)

เป็นการกำหนดรูปแบบของการโจมตีไว้ โดยดูจากเหตุการณ์ที่ใช้ โดยเฉพาะรูปแบบที่ใช้คือ เหตุการณ์ , ลำดับของเหตุการณ์ , ภาพรวมของเหตุการณ์ หรือใช้ Regular Expression ซึ่งรวม AND , OR แต่ละอย่างเพื่ออนุญาตให้ใช้งานโดยใช้ Finite-state Machine , ระบบ Rule-based หรือ ทรัพยากรในการตัดสินใจ (Decision Tree) , ระบบผู้เชี่ยวชาญ (Expert System) , เครือข่ายนิวรอล (Neural Network) , ระบบฟัซซี่ (Fuzzy Classification System) หรือ โมเดลความน่าจะเป็น โดยใช้เหตุผล (Probabilistic Reasoning Model)

ข้อดี คือ จำนวนและชนิดของเหตุการณ์ที่พบจำเป็นต้องมีรูปแบบเหมือนกันและพยายามจะเพิ่มการคำนวณให้ไปถึงเลขทศนิยมด้วย

ข้อเสีย คือ

- การขยายตัวหรือเพิ่มประสิทธิภาพเป็นหน้าที่ของขนาดและ โครงสร้างของฐานข้อมูล รูปแบบ หรือ กฎที่ใช้

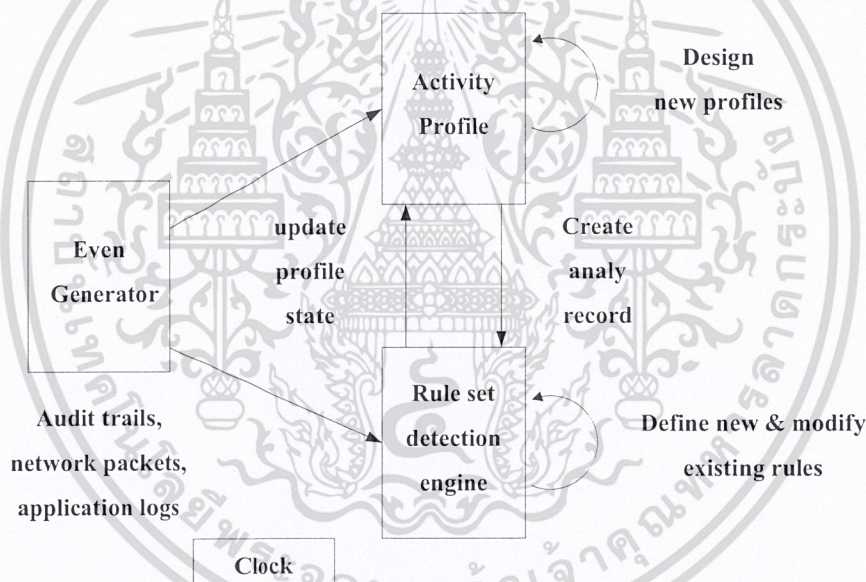
- เพิ่มขนาดได้ยาก เพราะไม่มีรูปแบบทั่วไป, ความเฉพาะของภาษา, เพิ่มรูปแบบทำให้ยากมากขึ้น

- การเพิ่มรูปแบบให้ฐานข้อมูล , ไม่มีการเรียนรู้ได้เอง

4.5.3 Realtime หรือ Internet Based

Vulnerability Scanner ใช้ในการหาจุดอ่อนของระบบปกติจะทำงานภายในพื้นฐานจุดประสงค์ ใช้ในการตรวจสอบเครือข่ายและระบบเพื่อหาจุดอ่อน

แนวคิด คือ จะมีการแสดงพฤติกรรมที่เปลี่ยนแปลงของผู้ใช้ซึ่งคิด Realtime IDS มาเพื่อตรวจสอบดูพฤติกรรมเฉพาะอย่าง สิ่งที่ทำหายคือ เหตุการณ์ที่ต้องการจะตรวจสอบนั่นเอง



รูปที่ 4-13 แสดง Generic Intrusion Detection Model

ประกอบด้วย 3 คอมโพเนนต์ คือ

4.5.3.1 Event Generator

4.5.3.2 Activity Profile

4.5.3.3 Rule Set

4.5.3.1 Event Generator

ส่วนจัดการข้อมูลเกี่ยวกับกิจกรรมของระบบ อาทิเช่น audit trails จาก network traffic หรือจากระบบย่อย และโปรแกรมเป็นไฟร์วอลล์ หรือ authentication servers

4.5.3.2 Activity Profile

ดูแลสถานะของระบบ หรือ เครือข่ายที่ตรวจสอบถ้ามีเหตุการณ์ที่เกิดขึ้นอยู่จาก data source จะมีการ uodate ใน Activity Profile หรือ กำหนดสร้างตัวแปรขึ้นอยู่กับ Rule Set ที่ใช้

การฟีดแบ็ก (feedback) มีความจำเป็นในโมเดล เหตุการณ์ที่เกิด rule-based มีการเรียนรู้ (learn) และเพิ่มกฎ (rule) ใหม่ ถ้า Rule Set ตรวจพบการเปลี่ยนแปลงใน Activity Profile อย่างเช่น เปลี่ยนแปลงชนิด , ความถี่ในการใช้งานหรือรายละเอียดของเหตุการณ์ที่มาจาก Event Generator นั้นเอง

4.5.3.3 Rule Set

เป็นตัวกำหนดว่าจะให้แสดงการตอบโต้อย่างไรเมื่อมีการโจมตีเกิดขึ้น บ่อยครั้งใช้ Rule-based expert systems มาใช้งาน ซึ่งเกี่ยวกับเหตุการณ์ (events) และลักษณะของข้อมูลที่ใช้ กฎ (rules) , รูปแบบ (patterns) และสถิติจากการใช้พฤติกรรมโจมตี

4.5.4 ประเภทของการโจมตี (Class of Attacks)

จุดเริ่มต้น (Point of Origin)	ผู้ใช้งานภายใน (Internal User)	ผู้ใช้งานภายนอก (External User)
การปิดการให้บริการ (Denial of Services)	การรบกวน (Annoying)	การรบกวน (Annoying)
การเพิ่มระดับความเป็นส่วนตัว (Increased Privilege)	ปัญหาระดับปานกลาง (Moderately Serious)	ปัญหาระดับความเสี่ยงสูง (Serious Risk)
ระดับผู้ดูแลระบบ (Superuser Privilege)	ปัญหาระดับความเสี่ยงสูงมาก (Very Serious)	ปัญหาความเสียหายต่อระบบ (Disaster)

ตารางที่ 4-2 แสดงประเภทของการโจมตี

แบ่งการโจมตีออกเป็น 2 ประเภท

4.5.4.1 ผู้ใช้งานภายใน (Internal Users)

4.5.4.2 ผู้ใช้งานภายนอก (External Users)

ซึ่งระดับของความปลอดภัยขึ้นอยู่กับกลุ่มของผู้ใช้งาน มีสิทธิ์ในแต่ละระดับไม่เท่ากัน จากตารางจะเห็นได้ว่าเป็นระดับแสดงให้เห็นถึงรูปแบบโปรแกรมที่ใช้งานอีกด้วยจะผูกติดกับความน่าสนใจของปัญหาที่ควรใส่ใจในการโจมตีไว้ด้วย

4.5.4.1 ผู้ใช้งานภายใน (Internal Users)

4.5.4.1.1 การโจมตีการปิดการให้บริการภายในเครือข่าย (Internal Denial of Service Attack)

เป็นการปฏิเสธการให้บริการจากภายในเครือข่ายปัญหานี้เกิดจาก

- การใช้ทรัพยากรของระบบมาก
- เขียนโปรแกรมประเภทใช้หน่วยความจำในบัฟเฟอร์เกินกว่าจะรองรับได้
- เพิ่มภาระคิวงานให้กับพรินเตอร์
- สร้างการติดต่อกับอุปกรณ์ I/O แบบขนาน เช่น ดิสก์

4.5.4.1.2 การเพิ่มระดับการใช้งานภายใน (Internal Privilege Escalation)

- โปรแกรมไม่ได้ใช้ขนาดจำกัดของบัฟเฟอร์ทำให้เกิด Buffer overflow attack
- โปรแกรมไม่ได้ตรวจสอบตัวแปรอินพุตค่าต่างๆ
- โปรแกรมที่ใช้ทรัพยากรของระบบมาก
- โปรแกรมมีกลยุทธที่ใช้งานมาก

4.5.4.1.3 การเพิ่มระดับเป็นผู้ดูแลระบบภายใน (Internal Superuser Privileges)

- ในระดับรูต (root) ของระบบมีสิทธิ์ที่ใช้งานโปรแกรมต่างจากผู้ใช้งานทั่วไปมาก อาทิเช่น buffer overflow attack , data driven attack , spoofed resourced , spoofed network packet

4.5.4.2 ผู้ใช้งานภายนอก (External Users)

มักเป็นบุคคลที่มีสิทธิ์ในระบบซึ่งความเป็นจริงปัญหาส่วนใหญ่มักเกิดจากบุคคลภายนอกเข้ามาโจมตีระบบเพื่อก่อให้เกิดความเสียหายกับระบบ

4.5.4.2.1 การทำความเสียหายการปิดการให้บริการจากภายนอก (External Denial Of Service Threats)

การโจมตีการปิดบริการจากภายนอกเป็นการยากที่จะป้องกันและแก้ไขแม้ว่าจะมี ไฟร์วอลล์ หรือ สกรีนนิ่งเราเตอร์ (screening router) เป็นตัวช่วยลดความเสียหายที่เกิดขึ้นได้ แต่ก็ได้ไม่มากนัก

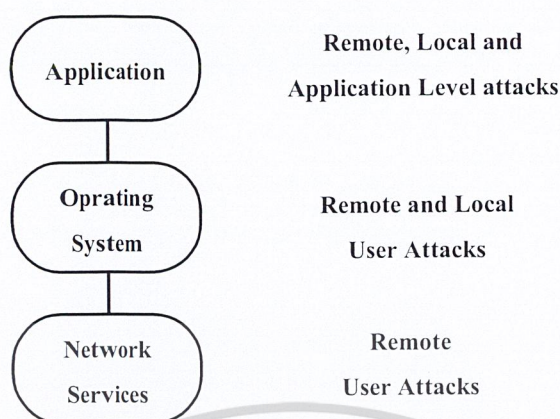
4.5.4.2.2 การเพิ่มระดับการใช้งานของผู้ใช้งานภายนอก (External Privilege Escalations)

มี 2 ทางที่ใช้ในการเพิ่มระดับของผู้ใช้งานจากระยะไกล (ภายนอก)

- โปรแกรมไม่ยอมให้ล็อกอินเข้าระบบแต่ยอมรับการเชื่อมต่อเครือข่าย
- ผู้ใช้งานระยะไกล (ภายนอก) พยายามล็อกอินเข้าระบบ โดยใช้โปรแกรมดักฟังการเชื่อมต่อจาก

ภายนอก

4.5.5 ระดับของแหล่งข้อมูลข่าวสาร (Layer of Information Sources)



รูปที่ 4-14 แสดงระดับของแหล่งข้อมูลข่าวสาร

4.5.5.1 การบริการงานเครือข่าย (Network Services)

ประกอบด้วยชั้นย่อยๆภายในมันเอง แต่ในที่นี้จะมองเป็นชั้นเดียวของ ทีซีพี/ไอพี หรือ SMB traffic

4.5.5.2 ระบบปฏิบัติการ (Operating System)

ประกอบด้วยชั้นของเครือข่าย (Network Stack) ซึ่งไม่รับผิดชอบเกี่ยวกับข้อมูลข่าวสาร โดยตรง ประกอบด้วย คำสั่ง (commands) , ไลบรารี (libraries) , วิธีการแบ็กอัพ (backup routine) , วิธีการล็อกอิน (login routine)

4.5.5.3 โปรแกรม (Application)

เป็นส่วนของโปรแกรมที่ติดต่อ , ทำงานกับโอเอสทั้งหมด

4.5.6 IDS ในเชิงพาณิชย์

จะแบ่งการสนใจไว้ที่ชั้นเครือข่าย (Network Layer) และ ระบบดูแล (System Monitor) ที่สนับสนุนการทำงานในชั้น แอปพลิเคชัน มีความแตกต่างของแต่ละผลิตภัณฑ์และวิธีการที่ใช้ เช่น

- IP Address spoofing , SYN flood , Teardrop Attack จะถูกตรวจสอบที่ใช้ซอฟต์แวร์เน็ตเวิร์ค
- Buffer Overflow Attack ถูกใช้โดยผู้ใช้งานใช้การล็อกอินถูกตรวจสอบโดยในชั้นระบบปฏิบัติการ (Operating System : OS) เพราะเหตุการณ์ (events) จะไม่ถูกสร้างในชั้นเครือข่าย Network IDS จะวิเคราะห์เหตุเรื่อง Buffer Overflow Attack

- ส่วนการทำปฏิบัติการเก็บข้อมูลในโปรแกรมจะถูกตรวจสอบโดยชั้น แอปพลิเคชัน ปัญหาที่แตกต่างกันทำให้ IDS มีความสามารถที่จะแก้ไขในแต่ละชั้นได้แตกต่างกัน เช่น ในชั้นของเครือข่ายจะมีเฮดเดอร์ (header) ของแพ็กเก็ต คือ

- IP Address

- Port Numbers
- Sequence Numbers
- Protocol Type
- Names of Network Entities เช่น server name ใน SMB
- Checksums
- ออปชัน เช่น source routing สำหรับ ไอพี

ในการตรวจสอบดูแลเครือข่ายจะดูการโจมตีเป็น Address Spoofing , Session Hijacking ในชั้นนี้ จำนวนของเหตุการณ์ที่ไม่ง่ายในการตรวจสอบคือ Failed Login , การลบไฟล์ (File Deletion) , การแสดง ตนล้มเหลว (Fail Authentication)

4.5.7 การโจมตีอย่างง่าย หรือ ซับซ้อน (Simple or Complex Attacks)

- ผู้ใช้งานไม่มีสิทธิ์ที่จะใช้โปรแกรมเปลี่ยนระดับการใช้งาน (Privilege Level)
- ผู้ใช้งานไม่มีสิทธิ์ที่จะอ่านไฟล์ที่ไม่มีสิทธิ์
- โปรแกรมลบไฟล์จำนวนมาก
- บางคนสร้างไฟล์โดยมีอักขระพิเศษ
- ผู้ใช้งานเรียกใช้งานโปรแกรมใน path ไม่ถูกต้อง
- การ spoof แพ็กเก็ต ไอพีแอดเดรส
- ล็อกอินล้มเหลว (fail login)
- ใช้พอร์ตสแกนนิ่ง (Port Scanning) จากไอพีแอดเดรสเดียวกัน
- ใช้ Resource Flooding เป็นตัวส่งแพ็กเก็ตจำนวนมาก

4.5.7.1 ความยากของการตรวจสอบผู้บุกรุก

- ผู้ใช้งานเรียกโปรแกรม privilege
- ผู้ใช้พยายาม login เป็น superuser (su) ล้มเหลว

4.5.7.2 Vulnerability Scanners

ขั้นต่อไป คือ การค้นหาช่องโหว่และจุดอ่อนของระบบโดยการสแกนระบบแบ่งออกเป็น 2 ประเภท

- Local Scan ตรวจสอบภายในเครือข่าย
- Remote Scan ตรวจสอบภายนอกเครือข่าย

โดยพื้นฐานการสแกนจะตรวจสอบสถานะของระบบเป็นช่วงๆ

4.5.7.3 ในระดับของ system ที่ต้องตรวจสอบ

- พยายามเขียน , ลิงค์ , ลบ ไดเรกทอรีที่ไม่มีสิทธิ์
- พยายามแก้ไข resource เช่น เวลาของระบบ
- พยายามแก้ไขค่า Configuration ของระบบ
- พยายามยุติ audit ของระบบย่อย

- พยายามรันโปรแกรม Rogue เช่น Zap , Crack , SATAN , COPS และอื่นๆ

- พยายามเปิดการทำงานของการ์ดเครือข่าย (Network Adapter) ในโหมด Promiscuous (สำหรับดักฟัง)

- พยายามรันโปรแกรม Exploratory เช่น who , rwho , finger , ps , find เป็นต้น
- พยายามเรียกใช้โปรแกรมที่ผู้ใช้งานไม่มีสิทธิ์ เช่น mount , export fs , mknfs เป็นต้น

4.5.7.4 การดักฟังของผู้โจมตี (Sniffing For Intruders)

ใช้ network IDS ในการดักฟังหรือดักจับแพ็กเก็ตเกิดในเครือข่ายโดยแก้ค่าคุณลักษณะของการ์ดเครือข่าย (Network Adapter) ให้ทำงานในโหมด Promiscuous คือ จะดักจับแพ็กเก็ตในซับเน็ต (Subnet) แล้วส่งให้ไดรเวอร์ควบคุมอุปกรณ์ (Device Driver) แล้วให้ IDS วิเคราะห์

4.5.8 NetRanger พัฒนาโดย WheelGroup และ Cisco

ซึ่งทำงานกับเราเตอร์เพื่อดูข้อมูลของ network traffic โดยมีการเข้ารหัสและความมั่นคง (Integrity) ระหว่างกันควรติดตั้งไว้หลังเราเตอร์เพื่อตรวจสอบแพ็กเก็ตที่เข้าระบบ จะมีคุณลักษณะอื่นๆอีกคือ

- สามารถที่จะดูแลได้หลายเครือข่ายโดยมีการควบคุมจากศูนย์กลาง
- มีการส่งรายงานไปยังศูนย์กลาง
- มีความปลอดภัยในการติดต่อสื่อสารระหว่างกัน
- สามารถกำหนด Signature ที่จะตรวจสอบเป็นกลุ่มของการโจมตีได้
- มีรายงานที่อ่านและเข้าใจได้ง่าย
- มีการ Alert แบบ Real-times และมีการตอบสนองได้
- มีการรวบรวมเฟรมเวิร์คของการจัดการระบบ เช่น Tivoli TME

4.5.9 ข้อดีของ Network IDS

ข้อดีที่เห็นได้ชัด คือ มีการเข้าไปใช้งานได้ง่าย ต่างจาก IDS ที่ต้องทำงานในระดับของ System ของทุกระบบที่ต้องการ ส่วน NIDS ต้องการแค่จุดเดียวในการมอนิเตอร์ (Monitor) การเปลี่ยนแปลงในซับเน็ต (Subnet)

- ค่าใช้จ่ายถูกกว่าที่จะต้องติดตั้งในทุกๆไคลเอ็นต์(Client) เหมือน IDS
- โครงสร้างของ Network IDS สนับสนุนการกำหนดคุณลักษณะในปัจจุบัน
- ข้อมูลที่อ่านเข้ามาจะถูกตรวจสอบแบบอิสระ
- การทำงานของ Network IDS ไม่ไปรบกวนการทำงานของระบบ

4.5.10 ข้อจำกัดของการดักจับแพ็กเก็ตเกิดในเครือข่าย

- ไม่สามารถตรวจสอบทุกๆแพ็กเก็ตได้ (Network Sniffer Do Not See All Packets)

- ไม่สามารถตรวจสอบแพ็กเก็ตที่มีการซ่อนการเข้ารหัสเอาไว้ได้ (Network Sniffers Are Blinded by Encryption) เช่น ในกรณีของ SSL ที่มีการเข้ารหัสโดยเว็บเซิร์ฟเวอร์ (Web Server) ดังนั้นเพื่อแพ็กเก็ตไหลผ่านไฟร์วอลล์โดยเข้ารหัสจะไม่สามารถถอดรหัสได้โดยจนกว่าจะถึงเว็บเซิร์ฟเวอร์ถอดรหัสเอง

- ไม่สามารถตรวจสอบในระดับ System ได้ (Missed System Level Attacks) เช่น

4.5.10.1 ไม่ทราบผลกระทบที่จะเกิดขึ้น (unknown side effects)

- สร้างไฟล์ที่สามารถเขียนได้โดยทั่วไป โดยโปรแกรมระดับการใช้งาน (Privilege) ที่มีผลกระทบกระบวน

การทำงานของการ์ดเครือข่าย (Network Adapter)

- ลดความปลอดภัยที่มีอยู่ในแหล่งทรัพยากร (Resource) เช่น สร้างไฟล์ /etc/passwd ที่เขียนได้โดยทั่วไป

- เปลี่ยน ระดับการใช้งานของผู้ใช้งาน (Privilege User Level)

- โปรแกรมประเภทสร้าง Back Door

4.5.10.2 กรณีที่การโจมตีไม่ได้ใช้ในการรับ/ส่ง แพ็กเก็ตในเครือข่าย

- การแฮ็ก (Hack) ผ่านจอเทอร์มินอล (Terminal : TTYs) โดยตรง และมีการใช้โปรแกรมเดาพาสเวิร์ดแบบ Brute Force โดยไม่มี Network IDS ตรวจสอบได้

- กรณี Network IDS ไม่ใช่โหนดปลายทาง (The Network IDS Is Not The Destination Node) เช่น NIDS ไม่รู้ว่าโหนดปลายทางจะรับแพ็กเก็ตที่เข้ามา หรือในกรณีที่ระบบไม่รับแพ็กเก็ตที่มี Check Sum

1. **Insertion Attack** คือ มีการรับแพ็กเก็ตไม่สมบูรณ์หรือติดเชื่อ (Inject) แต่โหนดปลายทางปฏิเสธหรือไม่รับ

2. **Evasion Attack** คือ โหนดปลายทางรับแพ็กเก็ตที่ไม่ถูกสนใจโดย IDS ตัวอย่าง กรณีที่มีการส่งหมายเลขลำดับ (Sequence Number) ของแพ็กเก็ตเข้ามาทำให้เกิดการซ้อนทับกันของข้อมูล IDS ปฏิเสธแต่โหนดปลายทางนำมาต่อกัน

3. การปฏิเสธการให้บริการ (**Denial of Service**) โดยเวลาดักจับ (Sniffer) แพ็กเก็ต NIDS จะจัดการคิวและจองหน่วยความจำ แต่โดยความเป็นจริงจะมีข้อบังคับขอบเขตไว้ด้วยข้อจำกัดของ Physical Memory และ Virtual Memory และถ้า NIDS ทำงานในโหมด Passive และการ Sniff เครือข่ายจะไม่สามารถบล็อกแพ็กเก็ตได้ ซึ่งไฟร์วอลล์จะทำได้ หรือกรณีที่ไฟร์วอลล์เสียหาย (Failed) จะต้องไม่อนุญาตให้มีแพ็กเก็ตผ่านเข้ามาได้ เพราะจะไปละเมิดนโยบายการรักษาความปลอดภัยของระบบ แต่ถ้าเป็น NIDS failed เกิดจากการใช้งานทรัพยากรมากเกินไปนโยบายจะถูกละเมิดเพราะ IDS ไม่สามารถบล็อกแพ็กเก็ตได้โดยจะทำงานแค่ Alert เมื่อมีปัญหาเกิดขึ้น

Product ที่แนะนำ (4 พฤษภาคม 1998)

1. IBM's out sourced solution using Net Ranger
2. ISS Real Secure
3. Network Flight Recoder (NFR)
4. Abirnet Session Wall

4.5.11 ความท้าทายของปัญหาของ IDS

คือ ต้องสามารถป้องกัน

- Ping of Death
- SATAN scanning
- ISS SAFE Snite scanning
- Port scanning
- ftp cwd~root
- phf
- SYN Flood

4.5.12 ปัญหาและการแก้ไข

4.5.12.1 เตรียมการป้องกันก่อนโจมตี (Preparation before you're hit)

4.5.12.2 ตรวจสอบ (Detection) หรือ ทำให้ค้นสภาพ (discovery) เมื่อเหตุการณ์เกิดขึ้น

4.5.12.3 ตอบสนองต่อเหตุการณ์ (Response to the incident)

4.5.12.1 เตรียมการป้องกันก่อนโจมตี (Preparation before you're hit)

- ตั้งทีมงานรับผิดชอบ โดยเฉพาะเตรียมพร้อมให้ผู้รับผิดชอบงานแต่ละคน
- ตั้งและแจกจ่ายนโยบายด้านความปลอดภัย
- แบ่งกัข้อมูลด้วยเทปหรือสื่อแบ็กอัพ
- ศึกษาเอกสารเกี่ยวกับไฟร์วอลล์ (Firewall) , เราเตอร์ (Router) , โมเด็ม (Modem) และ อุปกรณ์

ที่จะทำให้เกิดความเสียหายในเครือข่าย

- พยายามศึกษาและแนะนำเทคนิควิธีการโจมตีใหม่ๆ
- ซ้อมหรือทดลองโดยจำลองการโจมตี
- ทำเอกสารและนำไปทดลองกระบวนการตอบสนอง
- พัฒนาและวางแผนการติดต่อสื่อสาร
- บอกให้ผู้ใช้งานเข้าใจปัญหาที่เกิดขึ้น
- มีการเก็บล็อก (log) ในกิจกรรมที่สำคัญ
- เก็บและจดบันทึกการตั้งค่าระบบและการเปลี่ยนแปลงรวมทั้งกำหนดข้อบังคับ

4.5.12.2 ตรวจสอบ (Detection) หรือ ทำให้ค้นสภาพ (discovery) เมื่อเหตุการณ์เกิดขึ้น

ผู้ดูแลระบบ (Administrator) จะต้องรู้จักรบบหลายระบบที่มักเก็บ Audit Log แต่ก็ไม่สามารถรู้ไปหมดทุกอย่างเกี่ยวกับข้อมูล จุดอ่อนแรกที่เป็นจุดน่าสนใจตัวโปรแกรมมีไฟล์ล็อก (Log) แต่ IDS มีการรายงานออกมาแบบ Real-time Alert จะต้องมีการเฝ้า Account เท่านั้นที่จะสามารถตรวจสอบความผิดปกติหรือการโจมตีได้

4.5.12.3 ตอบสนองต่อเหตุการณ์ (Response to the incident)

ข้อแนะนำ 2 ข้อจาก Garfinkel และ Spafford (1996)

ข้อแรก remain calm and don't panic

จากการวิจัยจิตวิทยาแสดงให้เห็นว่ามนุษย์ไม่สามารถทำงานได้ดีภายใต้แรงกดดันหรือบีบคั้น จนกว่างานที่ได้จะได้รับการยอมรับและบรรลุตามวัตถุประสงค์

ส่วนอีกข้อหนึ่งเป็นการทำงานตามเอกสารที่ได้รับรู้ไว้ ซึ่งแตกต่างจาก Chapman และ Zuicky (1995) แนะนำไว้

Step 1 : Evaluate the situation and decide what response is required

ปฏิบัติเหตุการณ์และตัดสินใจว่าต้องการการตอบสนองได้อย่างไรบ้าง

Step 2 : Disconnect or shut down resource if necessary

ถ้าจำเป็นให้ตัดการเชื่อมต่อหรือปิดระบบ

Step 3 : Analyze and respond to the incident

วิเคราะห์และตอบสนองต่อเหตุการณ์

Step 4 : Alert other people according to your response policy

เตือนบุคคลอื่นว่าคุณจะต้องตอบสนองต่อปัญหาตามนโยบายที่ตั้งไว้

Step 5 : Save the system state

มีการแบ็กอัพข้อมูลของระบบเพื่อช่วยการ recovery ระบบใหม่

Step 6 : Restore hacked systems

ต้องตรวจสอบระบบว่ามีการเปลี่ยนแปลงหรือแก้ไขจากการ hack ให้ restore ระบบจากอุปกรณ์แบ็กอัพ

Step 7 : Document what happened

รายงานบอกถึงปัญหาที่เกิดขึ้นอาจจะเพิ่มการตรวจสอบให้มากยิ่งขึ้น

4.5.13 ชนิดของระบบต่อต้านการโจมตี (Intrusion Detection System : IDS)

1. Scanners
2. System Level IDSs
3. Network Sniffers

4.6 SSL Architecture

4.6.1 แนะนำ

จัดออกแบบให้กับ TCP เพื่อเพิ่มความปลอดภัยให้กับบริการความน่าเชื่อถือของ end-to-end SSL ประกอบด้วยโปรโตคอล (Protocol) 2 ชั้น

SSL Handshake Protocol	SSL Change Cipher Spec Protocol	SSL Alert Protocol	HTTP
SSL Record Protocol			
TCP			
IP			

ตารางที่ 4-3 แสดง SSL Protocol

โดย SSL Record Protocol เป็นบริการความปลอดภัยพื้นฐานให้กับชั้นโปรโตคอลที่สูงกว่า ประกอบด้วย Handshake โปรโตคอล , Change Cipher Spec โปรโตคอล และ Alert โปรโตคอล

4.6.2 แนวคิดหลัก 2 ข้อ

4.6.2.1 SSL session : ใช้เกี่ยวกับไคลเอ็นต์ (Client) และเซิร์ฟเวอร์ (Server) โดยเซสชัน (Session) ถูกสร้างโดยโปรโตคอลแฮนด์เชกและยังกำหนดค่าพารามิเตอร์ในการรักษาความปลอดภัยโดยการเข้ารหัส

4.6.2.2 SSL connection : ทำการติดต่อกับโปรโตคอลทรานสปอร์ตในโมเดลโอเอสไอซึ่งให้เหมาะสมกับการใช้บริการ

4.6.3 ในโปรโตคอล Handshake มีพารามิเตอร์ดังนี้

4.6.3.1 Session State กำหนดพารามิเตอร์ คือ

- Session Identifier
- Peer Certificate ใช้ X509.3 certificate
- Compression Method ใช้อัลกอริทึมในการบีบอัดเข้ารหัสไฟล์
- Cipher Spec บอกว่าจะใช้อัลกอริทึมอะไรในการเข้ารหัสข้อมูล และ Hash Algorithm ซึ่งใช้คำนวณใน MAC ซึ่งจะบอกค่าในการเข้ารหัสเป็นขนาดของ Hash
- Master Secret : ขนาด 48 บิตในการแชร์ระหว่างไคลเอ็นต์ (Client) กับเซิร์ฟเวอร์ (Server)
- Is Resumable : เป็นแฟล็กที่บอกถึงค่าเริ่มต้นการติดต่อสื่อสาร

4.6.3.2 Connection State มีพารามิเตอร์

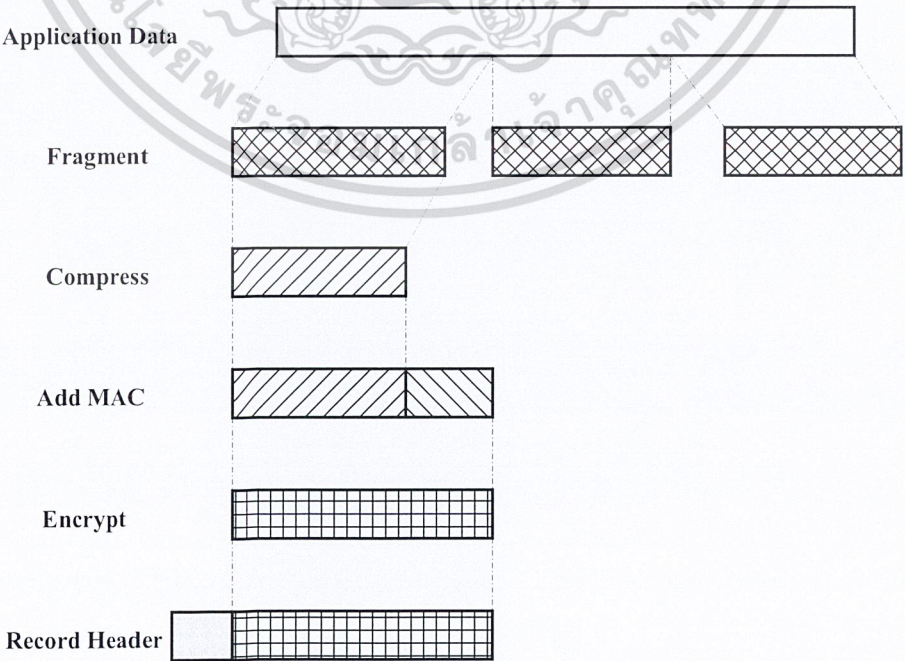
- Server และ Client Random เป็นลำดับ (Sequence) ของไฟล์ที่ถูกใช้โดยไคลเอนต์ (Client) กับเซิร์ฟเวอร์ (Server)
- Server write MAC secret เป็น secret key ที่ใช้ใน MAC ของข้อมูลถูกส่งโดยเซิร์ฟเวอร์ (Server)
- Client write MAC secret
- Server write key เป็น key เข้ารหัสธรรมดาที่เซิร์ฟเวอร์ (Server) เข้ารหัสและถอดรหัสโดยไคลเอนต์ (Client)
- Client write key เป็น key เข้ารหัสธรรมดาที่ไคลเอนต์ (Client) เข้ารหัสและถอดรหัสโดยเซิร์ฟเวอร์ (Server)
- Initialization Vectors (IV) ในโหมด CBC เมื่อ Block cipher ใช้ IV (Initialization vector) จะคูณแยกกันแต่ละตัว
- Sequence Numbers ลำดับของตัวเลขปกติมีค่าไม่เกิน $2^{64} - 1$

4.6.4 SSL Record Protocol

ให้บริการ 2 อย่างใน SSL connection

4.6.4.1 Confidentiality โพรโตคอล Handshake กำหนดใช้ Shared Secret Key ที่ใช้ในการเข้ารหัส Conventional ของ SSL Payload

4.6.4.2 Message Integrity โพรโตคอล Handshake กำหนด Shared Secret Key อยู่ในรูปแบบของ Message Authentication Code (MAC)



รูปที่ 4-15 แสดง SSL Record Protocol Operation

4.6.5 อธิบายการทำงาน

- 1. **Fragmentation** แบ่งขนาดข้อมูลเป็นบล็อกๆละ 2^{10} ไบต์ (16 , 384 ไบต์)
- 2. **Compression** การบีบอัดจะต้องไม่สูญหาย และขนาดเพิ่มไม่เกิน 1024 ไบต์
- 3. **Message Authentication Code** ใช้ Shared Secret Key

การคำนวณโดยสูตร

```
hash(MAC_write_secret || pad_2 ||  
      hash(MAC_write_secret || pad_1 || seq_num || SSLCompressed.type ||  
            SSLCompressed.length || SSLCompressed.fragment))
```

เมื่อ

	= Concatenation
MAC_write_secret	= shared secret key
hash	= algorithm ในการเข้ารหัส (MD5 , SHA-1)
pad_1	= เป็นไบต์ 0x36 (0011 0110) วง 48 รอบ (384 บิต) สำหรับ MD5 วง 40 รอบ (320 บิต) สำหรับ SHA-1
pad_2	= เป็นไบต์ 0x5C (0101 1100) วง 48 รอบสำหรับ MD5 วง 40 รอบสำหรับ SHA-1
seq_num	= ลำดับของหมายเลข
SSLCompressed.type	= โพรโตคอลในชั้นที่สูงกว่าใช้กระบวนการแฟร็กเมนต์ (Fragment)
SSLCompressed.length	= ขนาดความยาวของแฟร็กเมนต์บีบอัด
SSLCompressed.fragment	= แฟร็กเมนต์บีบอัด (ถ้าไม่มีการบีบอัดจะแบ่ง Plaintext แทน)

4. การเข้ารหัส (Encrypted)

มีการเข้ารหัสแบบ Symmetric ไม่เกิน 1,020 ไบต์ เมื่อรวมกันแล้วไม่เกิน $2^{14} + 2048$

โดยใช้อัลกอริทึมเข้ารหัสอื่น

Block Cipher		Stream Cipher	
Algorithm	Key Size	Algorithm	Key Size
IDEA	128	RC4 - 40	40
RC2 - 40	40	RC2 - 128	128
DES - 40	40		
DES	56		
3DES	168		
Fortezza	80		

ตารางที่ 4-4 เปรียบเทียบการเข้ารหัส

5. Append SSL record header

จะมีการเพิ่ม header ต่างๆดังนี้

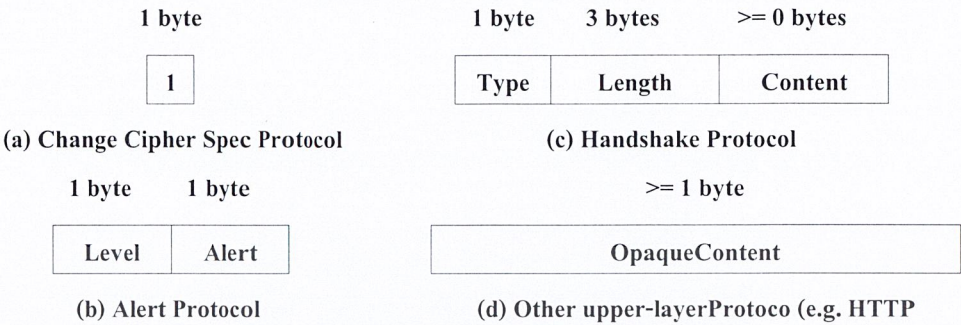
- Content Type (8 บิต) ใช้โดยโพรโตคอลที่สูงกว่าที่ปิดแฟรกเมนต์
- Major Version (8 บิต) บอกเวอร์ชันหลัก SSL เป็น SSLv3 ค่าเป็น 3
- Minor Version (8 บิต) บอกเวอร์ชันรอง SSL เป็น SSLv3 ค่าเป็น 0
- Compressed Length (16 บิต) ความยาวของ plaintext ที่ถูกแบ่งมีค่าเป็น ไบต์(หรือแฟรกเมนต์บีบอัดถ้าใช้การบีบอัดด้วย) มีค่าสูงสุด $2^{14} + 2048$ ไบต์

Encrypted	Content Type	Major Version	Minor Version	Compressed Length
	Plaintext (optionally compressed)			
	MAC (0, 16 or 20 bytes)			

ตารางที่ 4-5 แสดง SSL Record Format

4.6.6 Change Cipher Spec Protocol

เป็นขนาดเพียง 1 ไบต์ มีค่าเป็น 1 บอกว่าเป็นส่วนที่เพิ่มเติมจาก State ที่มีอยู่เพื่อปรับปรุง Cipher Suit ในการติดต่อสื่อสาร



รูปที่ 4-16 แสดง SSL Record Protocol Payload

4.6.7 Alert Protocol

ถูกใช้งานใน SSL_related เตือนไปยังปลายทางและโปรแกรมอื่นที่ใช้ SSL โดยข้อความ Alert มีถูกบีบอัดและเข้ารหัส โดยข้อความประกอบด้วย 2 byte ไบต์แรกเป็นค่า Warning หรือ Fatal ส่วนไบต์ที่สองใช้โดยเมสเสจเอง

โดยถ้าค่าเป็น Fatal จะยุติการเชื่อมต่อ ส่วนการเชื่อมต่ออื่นๆจะยังคงมีต่อไปแต่จะไม่สามารถมีการเชื่อมต่อใหม่มาได้ ไบต์ที่สองจะเป็นค่าเฉพาะ

- unexpected_message

: ไม่สามารถรับข้อความได้
- bad_record_mac

: ค่า MAC ที่ได้รับไม่ถูกต้อง
- decompression_failure

: มีฟังก์ชัน decompression รับอินพุตไม่ถูกต้อง
- handshake_failure

: ผู้ส่งไม่สามารถรับค่าพารามิเตอร์ตอบกลับได้
- illegal_parameter

: ในฟิลด์ของข้อความ handshake ไม่อยู่ในค่าหรือไม่สัค่าบนฟิลด์
- close_notify

: ผู้รับตอบกลับไปยังผู้ส่งว่าไม่สามารถรับข้อความเพิ่มเติมในขณะนี้ได้
- no_certificate

: ค่า certificate ที่ได้รับไม่ถูกต้อง
- bad_certificate

: ค่า certificate ที่ได้รับเสียหาย
- unsupported_certificate

: ชนิดของ certificate ที่รับมามีชนิดไม่สนับสนุนการทำงาน
- certificate_revoked

: certificate ถูกยกเลิก
- certificate_expired

: certificate หมดอายุ
- certificate_unknown

: certificate ไม่รู้จัก

4.6.8 Handshake Protocol

โพรโทคอลที่จะอนุญาตให้เซิร์ฟเวอร์ (Server) กับไคลเอนต์(Client) ต้องมีการแสดงตนและมีการเข้ารหัส , MAC algorithm และ มีคีย์ (key) ที่ใช้ป้องกันข้อมูลใน SSL record อย่างเหมาะสม ใช้ก่อนการส่งข้อมูลระหว่างโปรแกรม

โดยมีรูปแบบดังรูป มี 3 필ด์คือ

- Type (1 ไบต์) แสดงประเภทของข้อความในตาราง
- Length (3 ไบต์) ความยาวของข้อมูล หน่วยเป็นไบต์
- Content (>= 1 ไบต์) เกี่ยวข้องกับข้อความซึ่งสัมพันธ์กับในตาราง

Message Type	Parameters
hello_request	Null
client_hello	Version, random, session id, cipher suite, compression method
server_hello	Version, random, session id, cipher suite, compression method
certificate	Chain of X.509v3 certificates
server_key_exchange	Parameters, signature
certificate_request	Type , authorities
server_done	Null
certificate_verify	Signature
client_key_exchange	Parameters, signature
finished	Hash Value

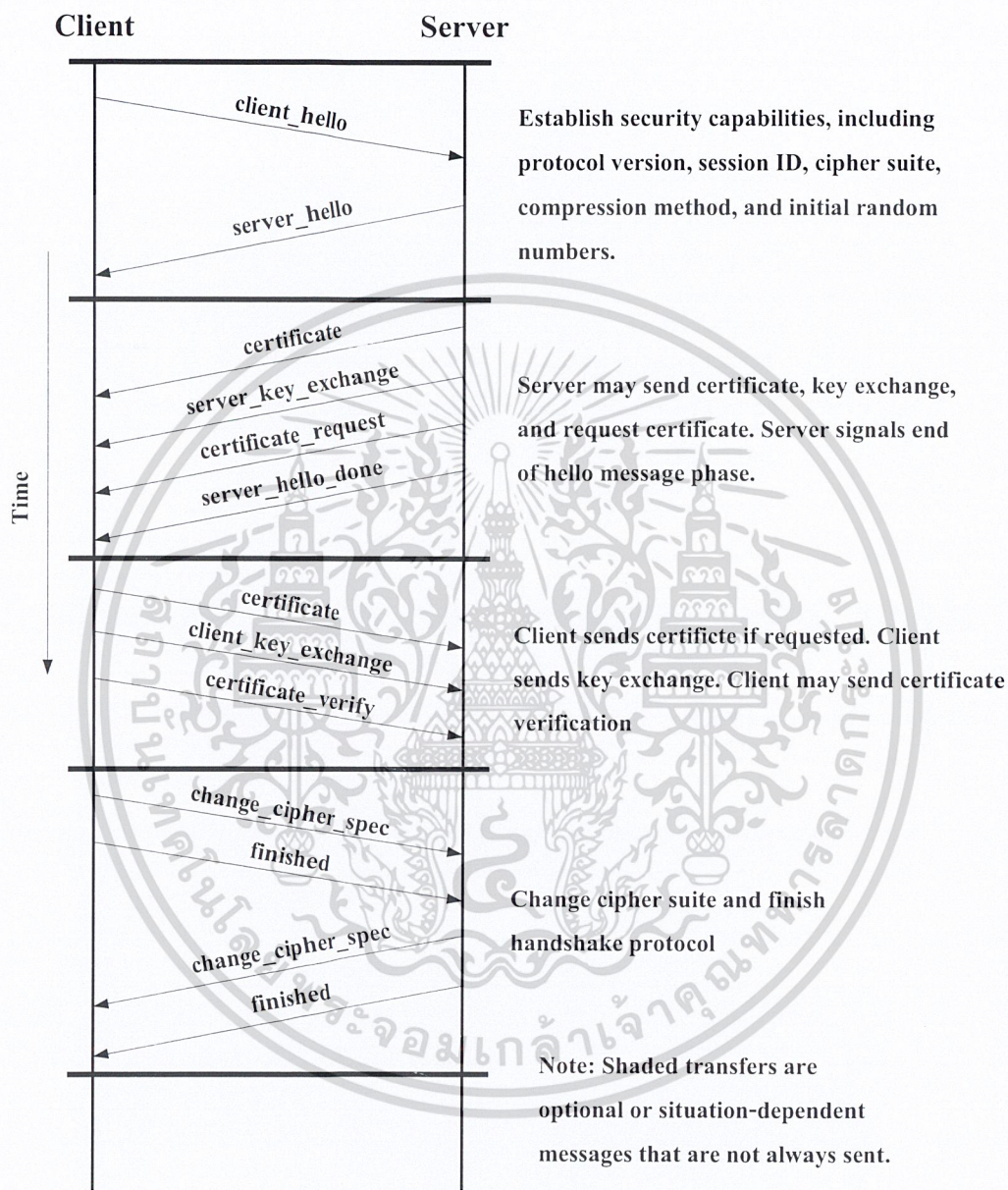
ตารางที่ 4-6 แสดงประเภทของข้อความ

Phase 1 : Establish Security Capabilities

เป็นการเริ่มต้นการติดต่อแบบปลอดภัยโดยไคลเอนต์จะส่ง client_hello_message โดยพารามิเตอร์คือ

- version เวอร์ชันสูงสุดของ SSL ถูกเข้าใช้โดยไคลเอนต์ได้
- Random ไคลเอนต์จะสร้างโครงสร้างใหม่โดยมี Timestamp ขนาด 32 บิตและอีก 28 บิตสำหรับเลขแบบสุ่มโดยปลอดภัยเพื่อป้องกัน Replay Attack

- session ID เปลี่ยนแปลงขนาดได้ตามแต่ละเซสชันเป็นศูนย์แสดงว่า ไคลเอ็นต์ต้องการเปลี่ยนแปลงพารามิเตอร์ที่เชื่อมต่ออยู่ หรือสร้างการเชื่อมต่อใหม่เป็นศูนย์แสดงว่าไคลเอ็นต์ต้องการเริ่มต้น เชื่อมต่อใหม่บน เซสชันใหม่



รูปที่ 4-17 แสดง Handshake Protocol Action

- Cipher Snite เป็นการรวมอัลกอริทึมในการเข้ารหัสที่ไคลเอ็นต์เข้าใจได้ โดยแต่ละ cipher snite จะกำหนดทั้งอัลกอริทึมในการแลกเปลี่ยนคีย์และ Cipher Spec
- Compression Method เป็นการวางวิธีการบีบอัดที่ไคลเอ็นต์เข้าใจ

เมื่อส่ง client_hello ไปแล้วไคลเอนต์จะรอ server_hello ตอบกลับ ซึ่งจะมีพารามิเตอร์เหมือนของ ไคลเอนต์โดยส่วนแรกของ Cipher Suite มีพารามิเตอร์บอกวิธีการแลกเปลี่ยนคีย์ มีวิธีดังนี้

- RSA ใช้ secret key เข้ารหัสโดยผู้รับ RSA public key ในการถอดรหัส
- Fixed Diffie-Hellman มีการแลกเปลี่ยนคีย์โดย เซิร์ฟเวอร์ ที่ certificate ต้องมี Diffie-Hellman ที่ sign โดย certificate authority (CA) โดย public key certificate จะประกอบด้วยพารามิเตอร์ Diffie-Hellman public key เมื่อไคลเอนต์ ใน certificate ต้องแสดงตนหรือมีการแลกเปลี่ยนคีย์จะได้ secret key กงที่ใช้ในการติดต่อ 2 จุดจะใช้ public key ขนาดคงที่ด้วยเช่นกัน
- Ephemeral Diffie-Hellman ใช้การสร้าง ephemeral secret key (ชั่วคราว , one_time) ใช้ Diffie-Hellman public key ในการแลกเปลี่ยน sign โดยใช้ key private RSA , DSS ของผู้ส่ง ผู้รับใช้ public key โดย Certificate ถูกใช้แสดงตนของ public key โดยใช้ความปลอดภัยสูงสุดเพราะใช้อุปกรณ์พิเศษ 3 ชั้น ผลที่ได้ชั่วคราวคือ Authenticate key

- Anonymous Diffie-Hellman ใช้ algorithm Diffie-Hellman แต่ไม่มีการ authentication
- Fortezza ใช้ใน smart card

วิธีการแลกเปลี่ยนคีย์คือ Cipher Spec จะประกอบด้วย

- CipherAlgorithm : ได้แก่ RC4 , RC2 , DES , 3DES , DES40 , IDEA , Fortezza
- MACAlgorithm : คือ MD5 หรือ SHA-1
- Cipher Type : Stream หรือ Block
- IsExportable : True หรือ False
- HashSize : 0 , 16 ไบต์ (MD5) หรือ 20 ไบต์ (SHA-1)
- Key Material : ลำดับของไบต์ที่รวมในข้อมูลใช้สร้างคีย์
- IV Size : ขนาดของค่าเริ่มต้น (Initialization Value) ของการเข้ารหัส CBC Cipher Block

Chairing

Phase 2 : Server Authentication and Key Exchange

เซิร์ฟเวอร์จะส่ง certificate ถ้ามีการแสดงตน (Authenticate) โดยข้อความจะมี X.509 certificate ด้วย โดย certificate message ต้องการวิธีแลกเปลี่ยนคีย์ agree_on ยกเว้นเป็น anonymous Diffie-Hellman แล้วตามด้วย server_key_exchange message ถ้ามีความต้องการใช้ด้วย แต่ไม่ต้องใช้ใน 2 กรณี คือ 1) เซิร์ฟเวอร์ ส่ง certificate ด้วยพารามิเตอร์ของ Diffie-Hellman ขนาดคงที่ไว้ หรือ 2) ใช้ RSA ในการแลกเปลี่ยนจำเป็นต้องใช้ตามด้วย

- Anonymous Diffie-Hellman มีค่า prime number และ primitive root รวมกับ public Diffie-Hellman key
- Ephemeral Diffie-Hellman ประกอบด้วย 3 พารามิเตอร์ใช้โดย anonymous Diffie-Hellman รวมกับพารามิเตอร์ signature ด้วย

- RSA key exchange ซึ่ง เซิร์ฟเวอร์ ใช้ RSA แต่ใช้เฉพาะ sign กับคีย์ RSA
- Fortezza โดยจะมีการใช้ hash ของข้อความและเข้ารหัสโดยผู้ส่ง public key โดย hash มีค่าเป็น $\text{hash}(\text{ClientHello.random} \parallel \text{ServerHello.random} \parallel \text{ServerParams})$ ครอบคลุมทั้งพารามิเตอร์ของ Diffie-Hellman และ RSA แต่ยั้งรวมถึง hello เริ่มต้นด้วยยังต้องป้องกัน replay attack และ misrepresentation กรณีของ DSS signature hash จะใช้ SHA-1 Algorithm ดังนั้น RSA signature ใช้ใน MD5 และ SHA-1 ตามด้วยเข้ารหัส 36 ไบต์ด้วย public key ของเซิร์ฟเวอร์

ต่อไปคือ ขอ certificate ของไคลเอ็นต์ โดยส่ง certificate_request message มี 2 พารามิเตอร์คือ certificate_type และ certificate_authorities และ server_done message บอกว่าจบ hell ของ เซิร์ฟเวอร์ และรอตอบกลับจากไคลเอ็นต์

Phase 3 : Client Authentication and Key Exchange

ตรวจสอบว่าถ้า เซิร์ฟเวอร์ ต้อง certificate ไคลเอ็นต์แล้ว certificate message แต่ถ้าไม่ส่ง no_certificate แทน ต่อไปยัง client_key_exchange message ขึ้นอยู่กับชนิดของคีย์แลกเปลี่ยนที่ใช้

- RSA สร้าง 48 ไบต์บน pre_master secret และเข้ารหัสด้วย public key จาก server certificate หรือ RSA ขั้วคราวหาก server_key_exchange message เอาไปใช้ master secret

- Ephemeral or Anonymous Diffie-Hellman ส่ง pulse Diffie-Hellman ของ ไคลเอ็นต์
- Fortezza กรณีเป็น Fortezza และจบ phases นี้ด้วยส่ง certificate_verify message จะถูก sign โดย hash code กำหนดเป็น

CertificateVerify.signature.md5_hash

$\text{MD5}(\text{master_secret} \parallel \text{pad_2} \parallel \text{MD5}(\text{handshake_message} \parallel \text{master_secret} \parallel \text{pad_1}));$

Certificate.signature.sha_hash

$\text{SHA}(\text{master_secret} \parallel \text{pad_2} \parallel \text{SHA}(\text{handshake_message} \parallel \text{master_secret} \parallel \text{pad_1}));$

โดยที่ pad_1 และ pad_2 ถูกกำหนดโดย MAC

Phase 4 : Finish

ไคลเอ็นต์จะส่ง change_cipher_spec message และใช้ CipherSpec ด้วย ตามด้วย finished message โดยมี hash เป็น $\text{MD5}(\text{master_secret} \parallel \text{pad_2} \parallel \text{MD5}(\text{handshake_message} \parallel \text{Sender} \parallel \text{master_secret} \parallel \text{pad_1}))$

$\text{SHA}(\text{master_secret} \parallel \text{pad_2} \parallel \text{SHA}(\text{handshake_message} \parallel \text{Sender} \parallel \text{master_secret} \parallel \text{pad_1}))$

และ เซิร์ฟเวอร์ตอบกลับด้วย change_cipher_spec message ตามด้วย finished message เสร็จสิ้น handshake ใช้ ไคลเอ็นต์ และ เซิร์ฟเวอร์ แลกเปลี่ยนข้อมูลในชั้น แอปพลิเคชัน แล้ว

4.7 คลัสเตอร์ (Cluster)

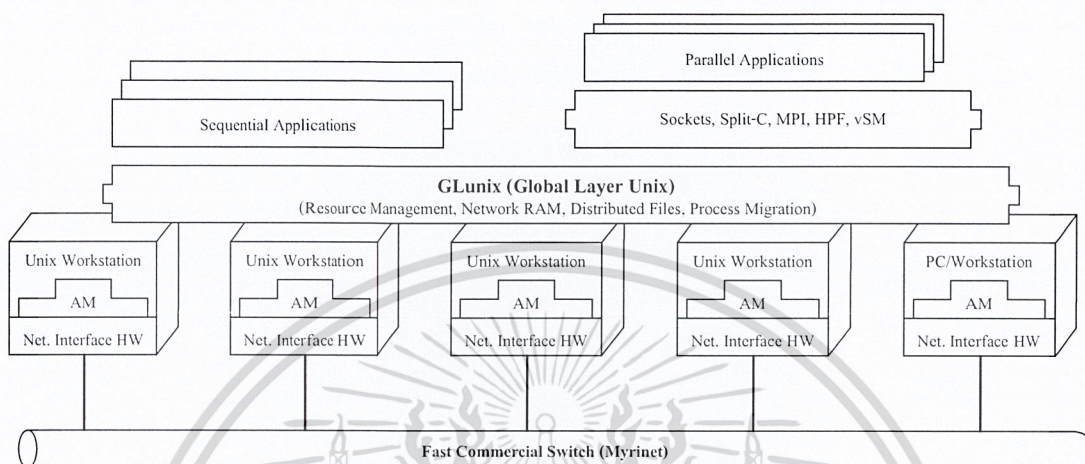
4.7.1 เครื่องคอมพิวเตอร์ประสิทธิภาพสูง (High Performance Computing)

- Massively Parallel Processor (MPP)
- Symmetric Multiprocessors (SMP)
- Cache-Coherent Nonuniform Memory Access (CC_NUMA)
- Distributed Systems
- Clusters

คุณลักษณะ	MPP	SMP CC_NUMA	Cluster	Distributed
จำนวนโหนด	0(100) – 0(1000)	0(10) – 0(100)	0(100) หรือน้อยกว่า	0(10) – 0(100)
ความซับซ้อนของโหนด	ธรรมดา หรือ ปานกลาง	ปานกลาง หรือ ซับซ้อนสูง	ประมาณปานกลาง	ขนาดใหญ่
การติดต่อสื่อสารระหว่างโหนด	Message passing/ แพร่ตัวแปรสำหรับ Distributed Shared Memory	Centralized และ Distributed Shared Memory (DSM)	Message Passing	Shared files , RPC , Message Passing และ IPC
การทำ scheduling	เป็นคิวเดียวบนโฮสต์	เกือบทั้งหมดใช้แบบคิวเดียว	หลายคิวแต่สัมพันธ์กัน	คิวจะอิสระจากกัน
โหนดของ OS และชนิดที่ใช้	N micro-kernels monolithic หรือ ในชั้นของ OS	SMP เดียวและ NUMA มีได้หลายตัว	รูปแบบ NOS homogeneous หรือ micro-kernel	รูปแบบ NOS homogeneous
ช่องว่าง address	แบบ Multiple – Single สำหรับ DSM	Single	Multiple หรือ Single	Multiple
ความปลอดภัยในการเชื่อมต่อ	ไม่จำเป็น	ไม่จำเป็น	จำเป็นบางกรณี	ต้อง
Ownership	องค์กรเดียว	องค์กรเดียว	องค์กรเดียว หรือ มากกว่า	หลายองค์กร

ตารางที่ 4-7 เปรียบเทียบคุณลักษณะของคอมพิวเตอร์แบบขนาน

คลัสเตอร์ (Cluster) คือ ชนิดของระบบขนานหรือกระจายการประมวลผลของระบบ ประกอบด้วย คอมพิวเตอร์ stand-alone แบบกลุ่มๆ ที่ทำงานด้วยกันทั้งแบบใช้งานเดี่ยวและรวมกันของทรัพยากร แต่ละโหนดอาจเป็นระบบเดี่ยวหรือระบบรวมของการประมวลผลทั้งหน่วยความจำ , I/O และระบบปฏิบัติการ



รูปที่ 4-18 แสดง โครงสร้างคอมพิวเตอร์แบบคลัสเตอร์ (Cluster)

การทำ Cluster จะได้คุณลักษณะเพิ่มเติม คือ

- ประสิทธิภาพสูง (High Performance)
- ขยายหรือเพิ่มขนาด ได้ง่าย (Exandability and Scalability)
- ประมวลผลเร็วขึ้น (High Throughput)
- เปลี่ยนแปลงได้ง่าย (High Availability)

4.7.2 องค์ประกอบสำหรับการทำคลัสเตอร์ (Cluster)

- ตัวประมวลผล (Processor)
 - กระบวนการในโครงสร้างการทำงาน (RISC , CISC , VLIW และ Vector)
 - แบบชิพเดี่ยว (Single_chip)
- หน่วยความจำและแคช (Memory และ Cache)
- ลิสต์และส่วนติดต่อ I/O
- บัสของระบบ
- การเชื่อมต่อของคลัสเตอร์มาตรฐานที่ใช้ อาทิ TCP/IP และระดับล่าง เช่น Active Message
- ระบบปฏิบัติการ (Operating System)

Linux (Beowulf Project) , Solaris (Solaris MC) , Window NT , Window 2000 Server , Advance Server

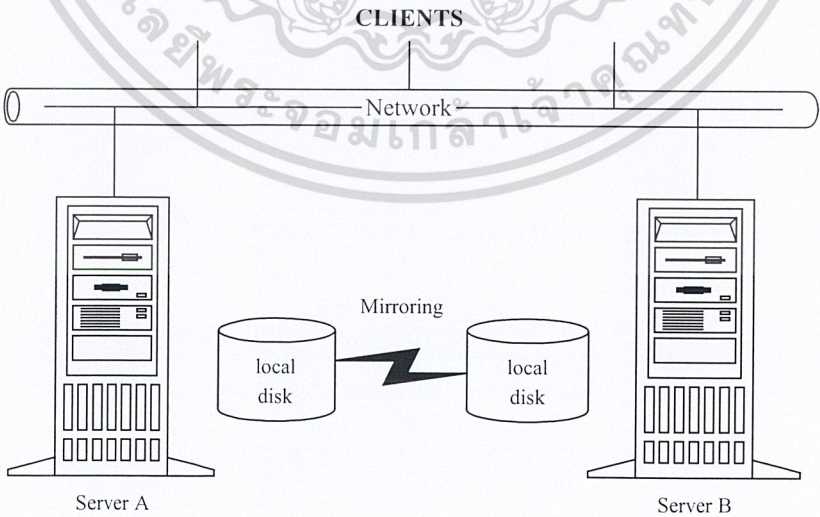
Project	Platform	Communication	OS	Other
Beowulf	PCs	Multiple Ethernet ด้วย TCP/IP	Linux และ Grendel	MPI/PVM Sockets และ HPF
Bereley NOW	Solaris-based PCs และ workstations	Myrinet and Active Message	Solaris + GLUunix + xFS	AM, PVM, MPI, HPF, Split-C
HPVM	PCs	Myrinet with Fast Message	NT หรือ Linux เชื่อมต่อกับตัวจัด การทรัพยากร + LSF	Java-fronted, FM, Sockets, Global Arrays, SHMEM และ MPI
Solaris	Solaris-based PCs และ workstations	Solaris-supported	Solaris + Globalization layer	C++ และ CORBA

ตารางที่ 4-8 เปรียบเทียบระบบคลัสเตอร์

4.7.3 โครงสร้างของคลัสเตอร์

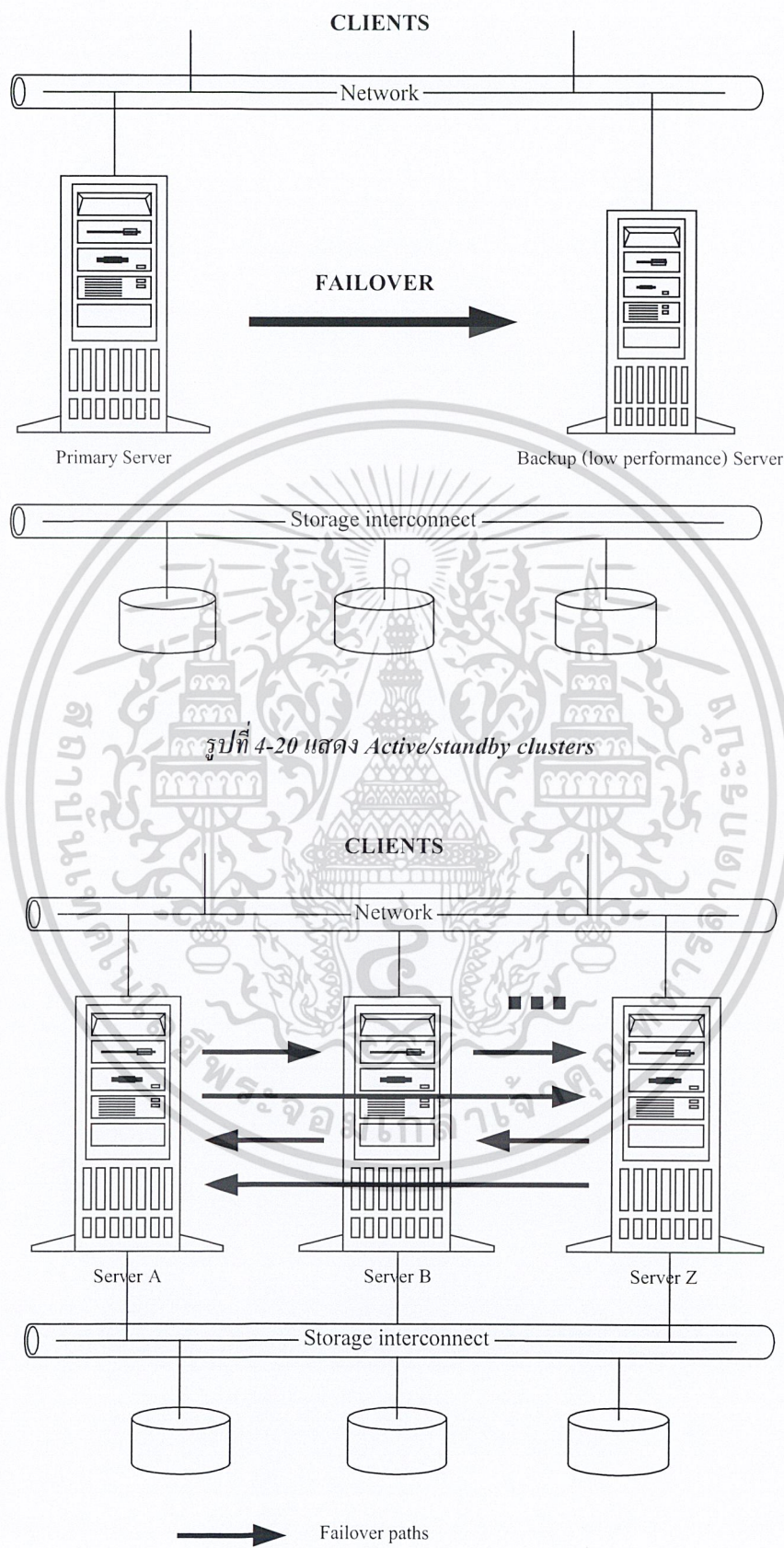
กำหนดได้หลายแบบ อาทิเช่น การเชื่อมต่อระหว่างระบบที่ใช้งาน storage , ไคลเอนต์ และ การตรวจดู (Monitorization) , ระดับของการแชร์ทรัพยากร , จำนวนสูงสุดของโหนด , การแก้ไขเมื่อเกิดความเสียหาย (failover) , ความสามารถในการใช้งานทรัพยากรของคลัสเตอร์อื่นๆ

4.7.3.1 Share-Nothing และ Shared-Storage



รูปที่ 4-19 แสดง Share-nothing clusters

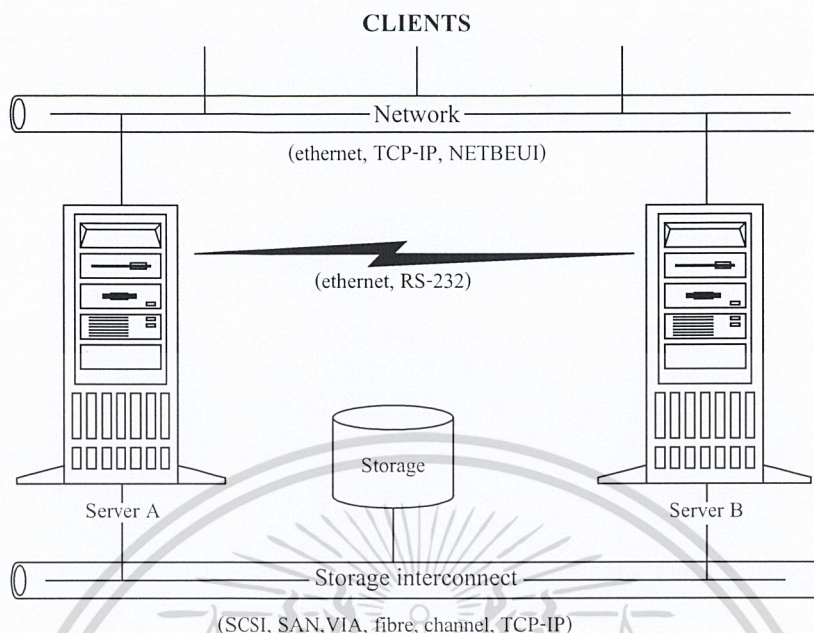
4.7.3.2 Active/Standby และ Active/Active



รูปที่ 4-20 แสดง Active/standby clusters

รูปที่ 4-21 แสดง N-Way clusters

4.7.3.3 การเชื่อมต่อ (Interconnects)



รูปที่ 4-22 แสดง Cluster interconnects

4.8 โหลดบาลานซ์ (Load Balance)

4.8.1 แนะนำ

โหลดบาลานซ์ คือ การใช้งานอุปกรณ์ภายนอกเพื่อใช้ประมวลผลโดยการแบ่งเป็นโหนดเพื่อจะกระจายงาน (workload) หรือ การใช้งานเครือข่าย (network traffic) ข้าม cluster สามารถกำหนดการกระจายงานเฉพาะ โหนดแบบคงที่ (static) และแบบปรับเปลี่ยนพลวัต (dynamic) ได้ขึ้นอยู่กับสถานะของเครือข่าย (network status)

โหนดแต่ละโหนดจะเชื่อมต่อกันเองภายในคลัสเตอร์ แต่ที่สำคัญ คือ จะต้องเชื่อมต่อกับอุปกรณ์บาลานซ์ที่เป็นตัวกระจายงาน (load distribution)

โหนดที่ทำหน้าที่ประมวลผลจะใช้สถานะของข้อมูลเกี่ยวกับภาระงานของโพรเซสเซอร์ (processor load) , ภาระงานของโปรแกรม (application system load) , จำนวนของผู้ใช้งาน (active users) , ขนาดบัฟเฟอร์ของโพรโทคอลเครือข่าย (network protocol buffers) , หน่วยความจำของระบบ (system memory) และอื่นๆ ในที่นี้จะใช้การมอนิเตอร์เพื่อตรวจสอบสถานะของโหนดประมวลผล (processing node) ที่อยู่ในโดเมนและการส่งงานที่เข้ามาต่อไป อุปกรณ์บาลานซ์สามารถทำงานเป็นแบบขนาน (parallel) หรือได้โครงสร้างทรี (tree hierarchy)

ในอุปกรณ์บาลานซ์จะใช้อัลกอริทึมตัวเดียวหรือมากกว่า หรือมีหลายวิธีการในการทำงานแบบคงที่ (static) และพลวัต (dynamic) ซึ่งมีการร้องขอการเชื่อมต่อเข้ามาด้วยวิธีการเหล่านี้ ทำให้โปรแกรมเฉพาะหรือโปรแกรมทั่วไปทำให้ระบบดีขึ้นได้และยังขึ้นอยู่กับโพรโทคอลเครือข่าย (network protocol) และการใช้งาน (traffic) มี 2 ช่องทางคือ 1) มองไปที่เครือข่ายและ 2) มองไปที่โปรแกรม

มองจากเครือข่าย (network point of view)

จะดูไปที่ข้อมูลที่เข้ามาเพื่อจะจัดแบ่ง (cluster) และกระจายงาน (distribute traffic) ขึ้นอยู่กับ โพรโตคอลเครือข่ายและข้อมูลการใช้งาน

มองจากโปรแกรม (application point of view)

จะใช้อัลกอริทึมกระจาย (distribution algorithm) ขึ้นอยู่กับ โปรแกรมเฉพาะนั้นๆ เน้นไปที่ โพรโตคอล ทีซีพี/ไอพี ซึ่งใช้งานบนพื้นฐานของแพ็กเก็ตแต่ ทีซีพี/ไอพี ในปัจจุบันรับประกันการใช้ (QoS : Quality of Service) ที่รับประกันการส่งข้อมูลในเครือข่ายที่เชื่อถือได้ แม้ว่าจะมีโพรโตคอล QoS และฮาร์ดแวร์เครือข่ายที่ใช้อยู่บนอินเทอร์เน็ตและเราเตอร์ของอินทราเน็ต (Intranet routers) และ เซิร์ฟเวอร์ก็ไม่ได้มีการนำมาใช้งานแต่การใช้งานโพลดาแลนซึ่งจะตั้งอยู่บนพื้นฐานของการรับประกันบริการ (QoS)

4.8.2 วิธีการ (Methods)

วิธีการนับเป็นหัวใจที่จะนำมาใช้งานในระบบ โดยจะใช้กรรมวิธีทางคณิตศาสตร์สัมพันธ์กับ สถิติของการทำงานของเครือข่ายและสถานะของโหนดเพื่อให้รองรับโหนดได้อย่างเหมาะสม โดยมี 4 โพรโตคอลมาเกี่ยวข้องคือ

1. Internet Protocol (IP)
2. Internet Control Message Protocol (ICMP)
3. Transmission Control Protocol (TCP)
4. User Datagram Protocol (UDP)

4.8.3 ปัจจัยหรือตัวแปรที่เกี่ยวข้อง

- 4.8.3.1 Network Address Translation
- 4.8.3.2 Domain Names
- 4.8.3.3 Wire-speed Processing
- 4.8.3.4 Node Operating System Limitations
- 4.8.3.5 Balancing Device Limitations
- 4.8.3.6 Session and nonsession-based Traffic
- 4.8.3.7 Application Dependencies

4.8.3.1 Network Address Translation

เป็นกระบวนการแปลงแอดเดรสภายในเครือข่ายและเครือข่ายส่วนตัวและข้อมูลหาเส้นทาง (routing) คู่แอดเดรส ภายนอกหรือ public address และหาเส้นทาง

ปัจจุบันจะใช้อินเทอร์เน็ตโพรโตคอล (Internet Protocol) เวอร์ชัน 4 (IPv4) ซึ่งในหลายบริษัทมีการนำไปใช้งานเพื่อเพิ่มงวดการใช้งานด้านความปลอดภัยให้กับระบบซึ่งมักเป็นไฟร์วอลล์ที่ทำหน้าที่นี้

โดยการทำงานจะใช้ตารางโดยแบ่งเป็นตารางภายในและตารางภายนอกในการตรวจสอบว่าจะติดต่อไปยังเครื่องใดหรือจะติดต่อออกไปภายนอกเครือข่าย

ด้านความปลอดภัยมักจะใช้ Internet IPSec เป็นมาตรฐานในการเข้ารหัสในชั้นทรานสปอร์ตระหว่างโหนด แต่ไม่มีการใช้งานใน Virtual Private Network (VPNs) ซึ่งอนุญาตให้เครื่องโหนดที่ติดต่อผ่านเครือข่ายหลักจะมีการเข้ารหัสโดยเครือข่ายหลักก่อนเมื่อต้องการส่งออกสู่เครือข่ายภายนอก

4.8.3.2 Domain Names

ในวิธีการบาลานซ์หลายวิธีใช้ Internet Domain names และ host names มาเกี่ยวข้อง โดยชื่อโฮสต์ใน ไอพี มี tag ที่สามารถจดจำและเข้าได้โดยมนุษย์ซึ่งสามารถนำไปใช้งานเป็น คีย์ หรือเป็นอุปกรณ์บาลานซ์ได้

Fully Qualified Domain Name (FQDN) ใช้งานทั้งชื่อโฮสต์ (host name) และชื่อโดเมน (Domain name) เป็นชื่อเดียวกันในระบบของอินเทอร์เน็ต โดยสามารถเป็นได้หลายชั้น

Domain Name System (DNS) เป็นวิธีการมาตรฐาน โดยในปัจจุบันสามารถ map ทั้ง multiple network address บางครั้งเรียก “Virtual domain” และที่อยู่ไอพี (IP address) จะสามารถมีได้หลายโดเมน (multiple domain names) บางครั้งเรียกว่า “Virtual IP address” และเซิร์ฟเวอร์ DNS จึงกลายเป็นตัวช่วยสำคัญในการทำระบบบาลานซ์เพื่อกระจายงาน

4.8.3.3 Wire-speed Processing

กล่าวถึงความสามารถของกระบวนการประมวลผลของการทำงานระบบเครือข่ายและการเปลี่ยนทิศทางเมื่อมีการใช้งานเต็มความเร็วของแพ็คเกจที่เข้ามาเพื่อป้องกันการเกิดคอขวด (bottlenecks) ของอุปกรณ์เครือข่าย

และยังขึ้นอยู่กับ Network Interface Card (NIC) ซึ่งเป็นอุปกรณ์ฮาร์ดแวร์ในการติดต่อผ่านเครือข่ายที่สำคัญในการรับ/ส่งข้อมูลและระบบปฏิบัติการอาจจะมีข้อจำกัดในส่วนนี้ด้วย ซึ่งผลที่ได้จะทำให้การตอบสนองช้าหรือไม่สามารถรองรับการเชื่อมต่อใหม่ในโหนดเฉพาะที่อยู่ในคลัสเตอร์ได้เลย

4.8.3.4 Node Operating System Limitations

บางระบบปฏิบัติการอาจมีข้อจำกัดในการประมวลผลแพ็คเกจที่รองรับการเชื่อมต่อไว้และยังรวมถึงชนิดของเครือข่ายด้วย ในเครือข่ายความเร็วสูง (High-speed network) สามารถรองรับการเชื่อมต่อได้จำนวนมากซึ่งทั้ง NIC และโอเอสมีข้อจำกัดร่วมกันทำให้บางครั้งไม่จำเป็นต้องทำงานให้เร็วเกินจำเป็นไว้

4.8.3.5 Balancing Device Limitations

ในอุปกรณ์บาลานซ์มีข้อจำกัดที่หน่วยความจำที่หน่วยความจำและความเร็วในการประมวลผล โดยจะเก็บตารางของข้อมูลที่เชื่อมต่อเข้ามาและสถานะของโหนดที่จำกัดขนาดของคลัสเตอร์ และอัตรา

เร็วของการประมวลผลข้อมูล (traffic processing rate) วิธีการบาลานซ์จะทำงานได้ดี เมื่อคลัสเตอร์มีขนาดเล็กอาจจะไม่สามารถขยายขนาดออกไปเมื่อมีโหนดเพิ่มขึ้น

4.8.3.6 Session and nonsession-based Traffic

ในการติดต่อแบบ ทีซีพี ปกติจะกำหนดแพ็กเก็ตไอพีเป็นแบบเรียงลำดับหมายเลข อุปกรณ์บาลานซ์บ่อยครั้งจะใช้ตารางการเชื่อมต่อของ ทีซีพี เพื่อเก็บการใช้งานการไหลไปยังโหนด

การใช้งาน Session-based โดยปกติจะมีในทุกๆอัลกอริทึมของ บาลานซ์ เมื่อเริ่มการติดต่อ ทีซีพี จากแหล่งต้นทางจะส่งแพ็กเก็ต โดยในส่วนหัว (header) จะบอกการเชื่อมต่อ TCP Synchronize (TCP_SYN) เพื่อบอกปลายทางว่าเริ่มต้นการเชื่อมต่อและปลายทางจะส่งหมายเลขเริ่มต้นกลับไป และเมื่อจบการติดต่อ (session) ต้นทางหรือปลายทางจะส่ง TCP Session Finished (TCP_FIN) บอกยุติการเชื่อมต่อ

อุปกรณ์บาลานซ์จะใช้แพ็กเก็ต ไอพี โดยจะดู TCP_SYN และ TCP_FIN เพื่อควบคุมการใช้งานระหว่างต้นทางปลายทางของโหนดเฉพาะใน คลัสเตอร์

ส่วนการใช้งาน nonsession-based เช่น ยูดีพีดาต้าแกรมโดยแพ็กเก็ต ยูดีพีไม่ต้องใช้การเชื่อมต่อ และบอกการเชื่อมต่อใดๆอย่างอิสระ คลัสเตอร์จะใช้การหว่านแพ็กเก็ตยูดีพีในโหนดจึงทำให้โปรแกรมไม่สามารถใช้งานได้ ในบางระบบใช้กำหนดระยะเวลาและต้นทางของยูดีพีดาต้าแกรมซึ่งเมื่อแพ็กเก็ตยูดีพีใหม่เข้ามาจะส่งไปยังโหนดได้ถูกต้อง แต่ก็รับประกันว่าจะแก้ไขยูดีพีโหนดบาลานซ์ได้

4.8.3.7 Application Dependencies

บางโปรแกรมต้องการติดต่อกับเฉพาะ โหนดซึ่งต้องการเชื่อมต่อโหนดเดิมในครั้งต่อไป เกิดขึ้นได้ในกรณี shared-nothing คลัสเตอร์เมื่อแต่ละ โหนดเป็นอิสระจากโหนดอื่น

การแก้ไขโปรแกรมได้โดยทำให้เป็น โปรแกรม cluster-aware ซึ่งบ่อยครั้งเป็นไปได้โดยอุปกรณ์บาลานซ์จะต้องระมัดระวังความต้องการในระดับชั้นของโปรแกรมซึ่งเป็นตัวเลือกใช้ในอัลกอริทึมบาลานซ์บางครั้งเรียก “sticky session” โดยส่วนใหญ่ในระบบเครือข่ายโหนดบาลานซ์จะใช้ ทีซีพี/ไอพีโดยโปรแกรมเครือข่ายจะเรียกว่า “Layer 4 optimized system” หรือ “Layer 4 switched network” อ้างอิงจากโมเดลโอเอสไอชั้นทรานสปอร์ต (Transport)

4.8.4 วิธีการ Balancing อย่างง่าย

จะใช้เพียงฟังก์ชันเดียวในการเลือกโหนดของคลัสเตอร์

4.8.4.1 Weighting

4.8.4.2 Randomization

4.8.4.3 Round-Robin

4.8.4.4 Hashing

4.8.4.5 Least Connections

4.8.4.6 Minimum Misses

4.8.4.7 Fastest Response

4.8.4.1 Weighting

เป็นการให้ค่า weight ของแต่ละโหนด โดยแต่ละโหนดจะมีค่าไม่เท่ากันดังนั้นเมื่อมีการเชื่อมต่อเข้ามาโหนดที่มีค่า weight สูงกว่าจะได้รับการร้องขอไป

4.8.4.2 Randomization

จะทำการตั้งค่าเริ่มต้นโดยใช้ pseudo random algorithm ให้แต่ละโหนดแต่ละจะใช้ weighting function โดยค่า weight ที่สูงสุด – ต่ำสุด จะกำหนดตามลำดับ

4.8.4.3 Round-Robin

บางครั้งเรียก “Rotary” หรือ “Cyclic” จะวนเรียงตามลิสต์ของโหนดโดยอุปกรณ์บาลานซ์จะส่งการร้องขอที่เข้ามาให้ไปให้ยังโหนดถัดไปเป็นวงรอบเมื่อเปรียบเทียบกับวิธี randomization จะดีกว่า และวิธีการนี้ยังถูกใช้ใน DNS name resolution ด้วย วิธีการนี้จะให้การได้ดีในกรณีที่ทุกๆ โหนดมีความสามารถและประสิทธิภาพเท่าเทียมกัน

4.8.4.4 Hashing

คล้ายกันกับการใช้ weighting แต่เพิ่มแพ็กเก็ตที่เข้ามาจากต้นทางเดิมจะถูกส่งไปยังเซิร์ฟเวอร์เดิมและอนุญาตให้ transaction ที่สำคัญกว่าก่อนอีกด้วย วิธีการนี้ยังช่วยในกรณีที่แพ็กเก็ตถูกใช้ในโปรแกรมเดียวกัน

4.8.4.5 Least Connections

วิธีการนี้อุปกรณ์บาลานซ์จะเก็บสถานะการเชื่อมต่อของแต่ละโหนดในคลัสเตอร์ไว้และจะเลือกโหนดที่ไม่ได้ถูกใช้งานก่อนเมื่อมีการเชื่อมต่อใหม่เข้ามา

ปัญหาของวิธีนี้คือ บางโปรแกรมมีการกินทรัพยากรในระบบมากกว่าตัวอื่น แก้ไขโดยกำหนดจำนวนการเชื่อมต่อให้จำกัดไว้ในแต่ละโหนด

4.8.4.6 Minimum Misses

จะมีการเก็บการร้องขอที่เข้ามาไว้ทั้งหมดที่ให้กับโหนดและให้กับการเชื่อมต่อที่น้อยที่สุดก่อน แตกต่างจาก Least Connections ตรงที่จะมีการเก็บและจัดเรียงจำนวนการเชื่อมต่อของปัจจุบันและในอดีต

4.8.4.7 Fastest Response

วิธีการอุปกรณ์บาลานซ์จะเก็บเวลาตอบสนองของเครือข่ายระหว่างโหนดไว้และเมื่อมีการเชื่อมต่อใหม่เข้ามาจะใช้โหนดที่ตอบสนองเร็วที่สุดไป ซึ่งต้องการ active monitor ในแต่ละโหนดโดยใช้เทคนิคการส่งแพ็กเก็ตไอซีเอ็มพีด้วยคำสั่ง Ping หรือวิธีการอื่น หรือใช้แพ็กเก็ตยูดีพีแทน

4.8.5 วิธีการ Advanced Balancing

ใช้วิธีการบาลานซ์อย่างง่ายมารวมกัน โดยมีตัวแปรที่ต้องพิจารณาเพิ่มเติมคือ

4.8.5.1 Network Traffic-based Balancing

4.8.5.2 Node Traffic-based Balancing

4.8.5.3 Node Load-based Balancing

4.8.5.4 Load-balancing Domain Name Resolution

4.8.5.5 Topology-based Redirection

4.8.5.6 Policy-based Redirection

4.8.5.7 Bandwidth Allocation Policies

4.8.5.8 Administrative Policies

4.8.5.9 Security Policies

4.8.5.10 Application-specific Redirection

Advanced Balancing Method	Optimization vectors
Network Traffic-based Balancing	network traffic optimization, network Route optimization, response latency minimization
Node Traffic-based Balancing	fair load distribution, response latency minimization
Node load-based Balancing	fair load distribution, response latency minimization
Load-balancing DNS Resolution	network traffic optimization, fair load distribution
Topology-based Redirection	response latency minimization, network route optimization, network traffic optimization, application-specific performance
Policy-based Redirection	administrative management optimization, network management optimization
Application-specific Redirection	network management optimization, fair load distribution, application-specific performance

ตารางที่ 4-9 แสดงตัวแปร Optimization Vectors ของวิธีการ Advanced Balancing

4.8.5.1 Network Traffic-based Balancing

ในระบบนี้ต้องการ active monitor เพื่อดูการใช้งานเครือข่ายที่เข้ามาจากหลายๆแหล่ง และกระจายงานไปตามโหนดต่างๆโดยจะอาศัยการทำนายปริมาณการใช้งานที่เข้ามาจากแหล่งต่างๆจากอดีตที่ผ่านมา วิธีการที่ง่ายที่สุดคือ อุปกรณ์บาลานซ์จะนับจำนวนการใช้งานที่เข้ามาเก็บไว้ในตาราง lookup จากค่าที่ได้เมื่อมีแพ็กเก็ตเข้ามาใหม่จากแหล่งเดิมอาจจะส่งไปยังโหนดเฉพาะการกระจายไปยังโหนดจะใช้ weighting function อย่างง่าย โดยให้งานโหนดเฉพาะ อัลกอริทึม distribution จะไม่ถูกใช้เฉพาะแต่ละโหนดโดยวิธีอย่างง่าย

ดังนั้นเทคนิคนี้จึงถูกใช้งานอย่างอิสระในคลัสเตอร์ซึ่งจะทำงานได้ดีที่สุดเมื่อเฉพาะโหนดให้ฟังก์ชันเหมือนกันหรือใกล้เคียงกัน

4.8.5.2 Node Traffic-based Balancing

เป็นส่วนกลับกันของ Network Traffic-based Balancing คือเมื่อมีการเข้ามาใช้งานจะถูกกระจายตามระดับของการใช้งานและค่าบัพเฟอร์เครือข่ายของแต่ละโหนดแทน

อุปกรณ์บาลานซ์จะติดต่อกับซอฟต์แวร์ เอเจนต์บนโหนดที่คอยเก็บสถานะของบัพเฟอร์เครือข่าย โดยจะกระจายงานที่เข้ามาไปยังโหนดที่มีบัพเฟอร์เครือข่ายมากที่สุด โดยมีการคาดการณ์ล่วงหน้า จากการวัดขนาดบัพเฟอร์ของปัจจุบันและอดีตเพื่อประมาณการล่วงหน้า วิธีการที่ใช้คือ Least Connections ซึ่งมีการใช้งานทั้ง Minimum Misses และ Fastest Response

วิธีนี้จะขึ้นอยู่กับ NIC โดยที่เซิร์ฟเวอร์ NIC จะมีหน้าที่ 2 อย่างคือ 1) failover และ trunking 2) เพิ่มการใช้งาน bandwidth ของโหนดที่เซิร์ฟเวอร์ใช้ NIC เป็นคู่ ก็เมื่อตัวใดตัวหนึ่งเสียหายอีกตัวต้องทำหน้าที่แทนได้ต่อไป ส่วน trunking หรือ aggregation โดย NIC สามารถ plug-in ได้โดยจะแบ่งเป็น symmetric และ asymmetric aggregation ระหว่างฮับ (hub) และเซิร์ฟเวอร์ โดย symmetric trunking คือ NIC ทั้งคู่ต้องรับส่งข้อมูลโดยมีไอพีเดียวแต่แตกต่างกันทั้งอีเทอร์เน็ต และ MAC

ส่วน Asymmetric trunking คือ NIC ทั้งคู่สามารถส่วนข้อมูลได้ทั้งสองทางโดยใช้ IP เดียวกันแต่รับได้เฉพาะตัวใดตัวหนึ่ง

4.8.5.3 Node Load-based Balancing

อุปกรณ์บาลานซ์จะมีซอฟต์แวร์ เอเจนต์บนโหนดซึ่งคอยดูโหลดที่เกิดขึ้นจากซีพียู หรือโหลดของระบบ โดยเมื่อได้ข้อมูลจากแต่ละโหนด อุปกรณ์บาลานซ์จะใช้วิธีการบาลานซ์อย่างง่ายเพื่อการกระจายงานที่รับเข้ามา ถ้าไม่คำนึงถึงแบนด์วิดท์จะไม่ใช้วิธี Least Connections และ Fastest Response

4.8.5.4 Load-balancing Domain Name Resolution

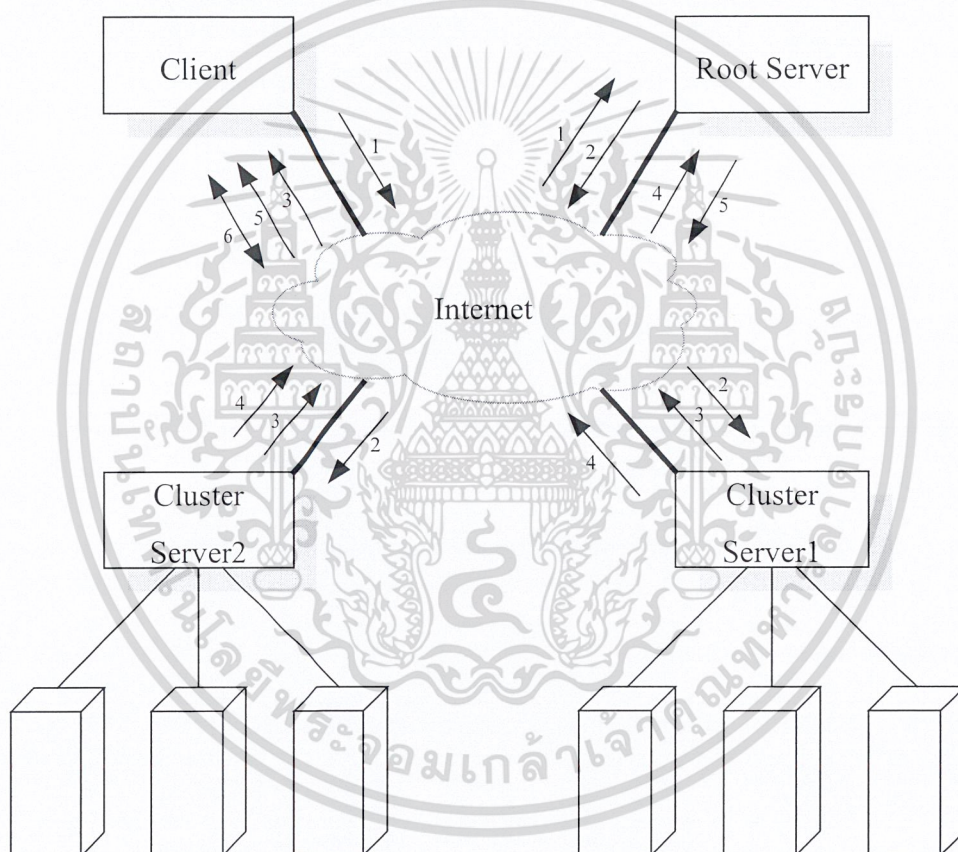
ใช้วิธีการ Round-Robin กับ DNS servers ซึ่งมักจะรวม weighting หรือ Hashing function ด้วยการโหลดบาลานซ์เกิดขึ้นภายใน DNS server ซึ่งไม่ขึ้นกับโปรแกรมที่ใช้งาน ปัจจุบันเป็นที่นิยมนำมาใช้เพราะง่าย โดยเพิ่มอัลกอริทึมที่เป็น โปรแกรมมาตรฐานใน DNS server เช่น BIND สำหรับยูนิกซ์ที่ใช้เพิ่มประสิทธิภาพโหลดบาลานซ์ให้เครือข่ายคลัสเตอร์และควรจะใช้ระบบปฏิบัติการเดียวกันเพื่อการตอบสนองที่เร็วขึ้น อาทิเช่น การคลัสเตอร์เว็บเซิร์ฟเวอร์ (cluster web server) บน เอ็นที และ ยูนิกซ์

4.8.5.5 Topology-based Redirection

ใช้งานในกรณีที่มียุทธศาสตร์ข้ามเครือข่าย ในเครือข่ายจะ redirect traffic ไปยังคลัสเตอร์ที่ใกล้ที่สุดให้ผู้ใช้ในรูปของ network topology ซึ่งมีวิธีการวัดอยู่ 2 วิธี คือ การนับฮอป (hop count) และ network latency

Hop count คือ นับจำนวนเราเตอร์ที่แพ็กเก็ตถึงปลายทางโดยค่าจำนวนฮอปที่เหลือน้อยที่สุดจะถูกใช้ ส่วน Network latency คือ จำนวนของเวลาหน่วยเป็นวินาทีที่ใช้ระหว่างไคลเอ็นต์ และคลัสเตอร์อุปกรณ์ปลายทาง

วิธีการที่ใช้ใน ทีซีพี/ไอพี ใน network latency คือ การ ping ส่งแพ็กเก็ตเพื่อทดสอบการตอบกลับ วิธีการที่ใช้ในการ ping คือ ping triangulation ดังรูป



รูปที่ 4-23 แสดง Ping triangulation ใน topology-based redirection

การทำงาน 6 ขั้นตอน

1. ไคลเอ็นต์ติดต่อไปยังเราเตอร์เมื่อเชื่อมต่อใหม่ไปยังเว็บไซต์อินเทอร์เน็ต
2. เราเตอร์จะส่งไปยังเซิร์ฟเวอร์คลัสเตอร์และส่งไอพีของไคลเอ็นต์
3. เซิร์ฟเวอร์คลัสเตอร์จะตรวจสอบระยะทางถึงไคลเอ็นต์
4. รายงานไปยังเราเตอร์บอกระยะทางถึงไคลเอ็นต์

5. รูตเซิร์ฟเวอร์จะปรับปรุงตารางในตัวเองและกำหนดให้ไคลเอนต์ที่จะติดต่อดำเนินการ redirect ไปยังคลัสเตอร์นั้นในรูปแบบเป็น คลัสเตอร์เซิร์ฟเวอร์ 1
6. ไคลเอนต์จะติดต่อโดยตรงกับ คลัสเตอร์เซิร์ฟเวอร์ 1 โดยใช้ไอพีตัวเอง (ใช้โดย Cisco และ RND)

4.8.5.6 Policy-based Redirection

เป็นโปรแกรมที่ใช้คณิตศาสตร์หรือ function set of rules ซึ่งกำหนดพฤติกรรมการทำงานของคลัสเตอร์โดยอุปกรณ์ บาลานซึ่งจะกำหนด rule set แต่ละ โหนดและเก็บไว้ในหน่วยความจำ การเข้าใช้งานจะถูกทดสอบว่าไม่ขัดแย้งกับกฎที่ตั้งไว้และถูกส่งไปยังโหนดอย่างเหมาะสม ถ้า rule set มีความซับซ้อนอาจจะใช้ไอพี หรือ บัญชีรายชื่อผู้ใช้งานเป็นคีย์ในตาราง hash ของ rule set ซับซ้อนขึ้นไปอีก ใช้การเปรียบเทียบแอดเดรสโดยใช้ซอฟต์แวร์หรือใช้ ASICs (Application-Specific Integrated Circuits) แต่โดยทั่วไปจะไม่ใช้ rule set ที่ซับซ้อนเพื่อให้ประมวลผลได้เร็วและให้การบาลานซ์ได้ดีขึ้น

โดยปกติจะเป็นเหมือนระบบ overarching balancing มีการใช้ทั้งวิธีการบาลานซ์อย่างง่ายและขั้นสูงขึ้น นโยบายจะสร้างและให้น้ำหนักทั้งแบบอัตโนมัติและกึ่งอัตโนมัติให้กับวิธีการที่เลือกใช้

4.8.5.7 Bandwidth Allocation Policies

เป็นการเพิ่มการกำหนดระดับความสำคัญ (priorities) ให้กับการใช้งานกับคลัสเตอร์เช่น ผู้ดูแลเครือข่ายและการควบคุมความปลอดภัยให้กับข้อมูล ปกติจะให้ค่าสำคัญกว่าการเข้าใช้งานเว็บ

วิธีการที่ใช้คือ Fastest response และ Least Connections กับเซิร์ฟเวอร์ที่ให้บริการ

4.8.5.8 Administrative Policies

ปกติจะใช้งานสัมพันธ์กับ work flow และธุรกิจมากกว่าการใช้เท่ากันกับโหนดบาลานซ์ในการร้องขอ นโยบายที่เข้มงวดจะขึ้นอยู่กับงานและการบริหารองค์กร งานบางอย่างจะแบ่งความสำคัญเป็นโหนดเดียวหรือเป็นกลุ่มๆ , หรือเฉพาะบุคคลหรือแผนก , ชนิดของโปรแกรม , ช่วงเวลาของวันหรือร้องขอจากบางแหล่ง

4.8.5.9 Security Policies

ตามจริงเป็นส่วนย่อยของนโยบายการบริหารและการจัดการแต่จะเน้นไปใช้งานในหลายเครือข่ายซึ่งเน้นและปรับใช้เป็นนโยบายสำคัญ โดยที่นโยบายด้านความปลอดภัยต้องการที่จะให้มีการแสดงตนก่อนเข้าใช้งานต่างๆ ซึ่งมักจะรวมเวลาเพื่อกำหนดความสำคัญการร้องขอไว้ด้วย ซึ่งมักจะมีการเข้ารหัสโดย Private Key และถ้าเราพูดถึงฐานข้อมูลมักจะใช้ RSA เป็นต้น

4.8.5.10 Application-specific Redirection

เป็นคุณลักษณะนำมาใช้งานขึ้นอยู่กับโปรแกรมและแหล่งที่ไคลเอนต์พยายามจะเข้าใช้ซึ่งถ้าไม่ได้มีการกำหนดไว้ในชั้น เซสชัน ของ ทีซีพี/ไอพี บางโปรแกรมสร้างระบบเซสชันเองโดยอุปกรณ์บาลานซ์จะต้องสนับสนุนด้วย ฐานข้อมูล (Database) และเว็บ (Web) นำไปใช้งานมาก ในฐานข้อมูลจะประกอบด้วยโหนดกระจายกันอยู่เมื่อมี query ไปยังเซิร์ฟเวอร์ฐานข้อมูลจะถูกแบ่งเป็น multiple query และส่งไปยังคลัสเตอร์ต่างๆเพื่อประมวลผล และรูตเซิร์ฟเวอร์จะตรวจสอบและส่งกลับไปยังไคลเอนต์

ในส่วนของเว็บ , HTTP ใช้โปรโตคอล ทีซีพี ต้องการไฟล์ข้อมูลซึ่งเอกสารถูกแบ่งเป็นหลายๆ ไฟล์รวมทั้งสคริปต์ใน HTML เมื่อไคลเอนต์อ่านหน้าหลักแล้วสามารถร้องขอไปยังหลายเซิร์ฟเวอร์ ทำให้ง่ายในการแบ่งไปยัง multiple เซิร์ฟเวอร์ โดยใช้ URL หรือข้อมูลแต่ละโหนด

4.8.6 ความผิดพลาด (Common Errors)

ความผิดพลาด 4 ประการ คือ

4.8.6.1 Overflow

4.8.6.2 Underflow

4.8.6.3 Routing Errors

4.8.6.4 Induced Network Errors

4.8.6.1 Overflow

เกิดขึ้นเมื่อมีการใช้งานเครือข่ายจำนวนมาก ทำให้เกิด lost packet หรือ แพ็กเก็ตเกิดไปไม่ถึงโหนด เกิดขึ้นได้ทั้งกับอุปกรณ์ บาลานซ์ และเฉพาะโหนด โดยมีการเกิดกระบวนการใช้งานที่อุปกรณ์บาลานซ์ ซึ่งมากกว่าเกิดขึ้นกับเฉพาะโหนด หรืออาจจะเกิดที่อุปกรณ์บาลานซ์ซึ่งยังผลให้ data stream เกิดการเสียหายหรือถูกสับออกก่อนจะถึงโหนด บางครั้งเกิดการสูญหายของข้อมูลและกระบวนการใน ทีซีพี จะเกิด time out ในกรณีไม่ได้รับการตอบกลับจากผู้รับ

4.8.6.2 Underflow

เป็นปัญหาเกิดภายในคลัสเตอร์เอง คือ โหนดใดโหนดหนึ่งได้รับทราฟฟิกไม่พอเมื่อเทียบกับโหนดอื่นๆ ผลให้เกิด underutilized หรือ starved ของโหนดอื่นๆ จำนวนโหนดลดลงแสดงให้เห็นถึงการลดประสิทธิภาพในการกระจายงาน ซึ่งมักเป็นที่ อัลกอริทึมหรือระบบไม่เหมาะสม

บางกรณีเป็นปัญหาของโหนดเป็น non-symmetric เป็นความไม่เท่าเทียมกันของประสิทธิภาพในการคำนวณและให้ทรัพยากรมากกว่าโหนดอื่นๆ เช่น มีการให้ค่า weight แต่ละโหนดไม่เท่ากันทำให้โหนดมีค่ามากได้รับงานมากกว่าโหนดอื่นๆ วิธี Least Connections และ Fastest Response ทำให้เกิดปัญหานี้ได้ในกรณีเป็น asymmetric cluster ซึ่งมีความสามารถในการตอบสนองหรือกระบวนการร้องขอได้รวดเร็วกว่า

4.8.6.3 Routing Errors

เกิดขึ้นระหว่างอุปกรณ์บาลานซ์และโหนดคลัสเตอร์ หรือระหว่าง ไคลเอนต์และโหนดคลัสเตอร์ซึ่งมักเป็นปัญหาจากการกำหนดคุณลักษณะหรือการเชื่อมต่อขาดหาย ซึ่งอาจจะเป็นอุปกรณ์บาลานซ์ซึ่งให้ค่าผิดพลาดหรือโหนดอาจจะใส่ค่าไอพีปลายทางผิด หรือการหาเส้นทางผิดพลาด การ disconnect link ปกติจะตรวจสอบโดยเราเตอร์บอกเป็น “No route to host” ทำให้ทุกวิธีโหนดบาลานซ์ซึ่งเกิด fail จนกว่าการเรตจ์จะถูกกู้คืนกลับมา (Restore) หรือโหนดคลัสเตอร์ทำงานแต่ไม่มีเส้นทางไปยังโหนดทำให้เกิด starvation ผู้ออกแบบเครือข่ายบาลานซ์คลัสเตอร์จะอนุญาตให้ตัดการเชื่อมต่อหรือปฏิบัติการทรานส์

แอ็กชันถ้าบริการ โปรแกรมสมบูรณ์ต้องการเชื่อมต่อสองระบบ ซึ่งคุณลักษณะจะต้องถูกรวมไว้ทั้งในอุปกรณ์บาลานซ์และ โหนดด้วย

4.8.6.4 Induced Network Errors

ต่างจาก Routing Errors ตรงที่เป็นความผิดพลาดจากสถานะปกติ และไม่ถูกต้องหรือสถานะเครือข่ายไม่คงที่ (unstable network state) ความเป็นจริงความผิดพลาดที่ไม่เกิดแต่ผลจากความสำเร็จในการส่งผ่านแพ็กเก็ตตามหาเส้นทางเครือข่าย แพ็กเก็ตที่ถูกส่งผ่านเครือข่ายจะล่าช้าเพิ่มขึ้นระหว่าง 2 จุดขึ้นอยู่กับเส้นทางระหว่างต้นทางและปลายทางที่ใช้งานด้วย ถ้ามีการใช้งานมากทำให้เกิดคอขวดในการหาเส้นทางเครือข่ายทำให้เกิดความผิดพลาดขึ้นได้ซึ่งมักเกิดกับวิธีการ Fastest Response และ Topology-based redirection

4.8.7 Dynamic Load Balancing

4.8.7.1 พิจารณาที่กระทำ คือ

- แบบ static ใช้ตอน Compile-time
- แบบ dynamic ใช้ตอน runtime

4.8.7.2 ข้อคิดที่เกี่ยวข้อง

- Static Scheduling จัดการตอน Compile-time
- Dynamic Scheduling จัดการตอน Run-time
- Predicting The Future คาดการณ์ล่วงหน้า
- Task Queue Model จัดการคิว
- Diffusion Model จัดตั้งค่าเริ่มต้นให้กระจายงานและการย้ายงานระหว่างโพรเซสเซอร์

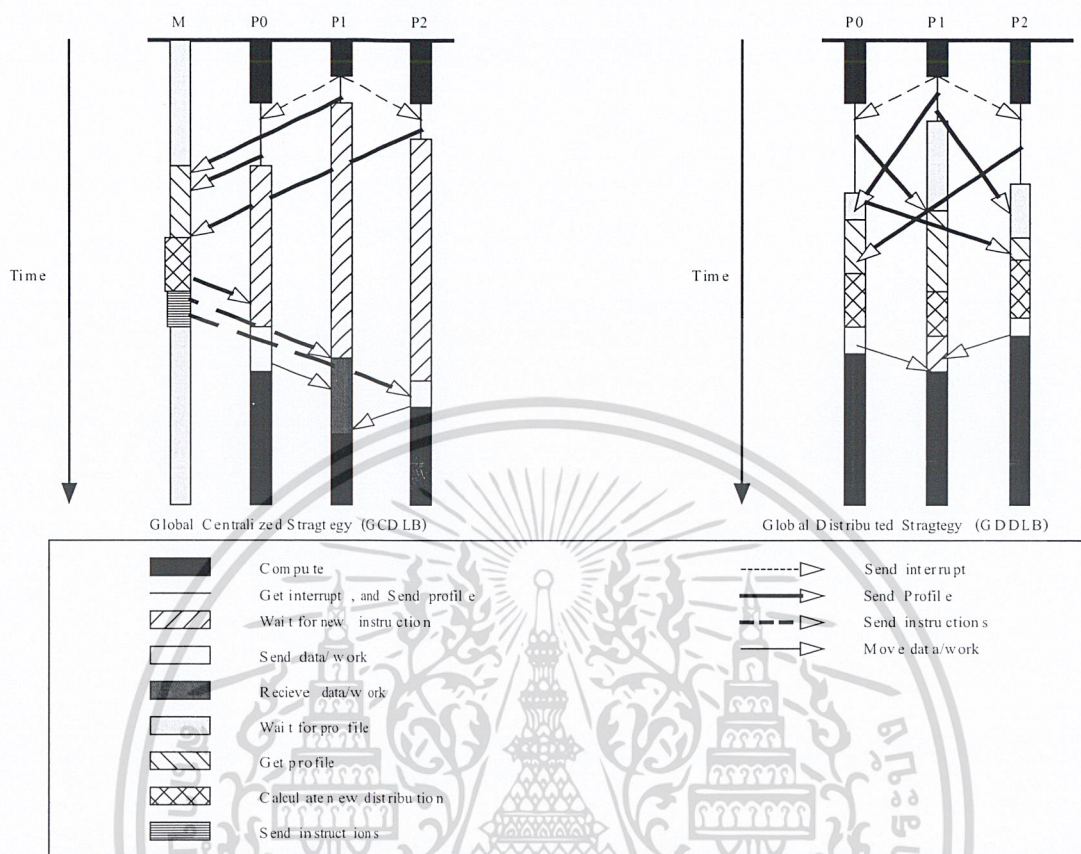
4.8.7.3 Dynamic Load Balancing (DLB)

จุดประสงค์เพื่อจัดการงานในแต่ละกระบวนการงานของโหนดให้ทำงานอย่างเหมาะสมกับประสิทธิภาพหลังจากตั้งค่าเริ่มต้น

ให้แล้ว DLB จะจัดการอย่าง

1. ดูประสิทธิภาพของตัวประมวลผล
2. การแลกเปลี่ยนข้อมูลระหว่างตัวประมวล
3. คำนวณการกระจายงานใหม่และตัดสินใจกระจายงาน
4. เคลื่อนย้ายข้อมูลจริง

4.8.7.4 กลวิธีการทำโหลดบาลานซ์ (Load Balancing Strategies)



รูปที่ 4-24 แสดง Centralized vs. distributed strategies

4.8.7.4.1 Global Strategies

ใน global knowledge เช่น ทุกๆ โพรเซสเซอร์จะทำการ Synchronization และส่งโปรไฟล์แสดงประสิทธิภาพให้โหลดบาลานซ์เซอร์

Global Centralized DLB (GCDLB)

ค่าโหลดบาลานซ์เซอร์จะอยู่ที่มาสเตอร์โพรเซสเซอร์ (Centralized) เมื่อหลังจากคำนวณคำสั่งใหม่แล้วจะส่งคำสั่งออกไปยัง โพรเซสเซอร์ อื่นๆ

Global Distributed DLB (GDDLb)

ตัวโหลดบาลานซ์เซอร์จะรองรับงานและกระจายงานไปยังทุกโพรเซสเซอร์

4.8.7.4.2 Local Strategies

จะทำงานโดยการแบ่งโพรเซสเซอร์เป็นหลายกลุ่มเรียก “K-block” หรือใช้การสุ่มเลือกเอา

Local Centralized DLB (LCDLB)

คล้าย GCDLB โดยโพรเซสเซอร์ที่เร็วที่สุดในกลุ่มจะเป็นตัวจัดการและในต่างกลุ่มจะใช้ Asynchronous handle จัดการ

Local Distributed DLB (LDDLb)

ตัวโหนดบาลานเซอร์จะรับคำสั่งจากทุกๆ โพรเซสเซอร์แต่โพรไฟล์ของข้อมูลจะถูกตั้งเฉพาะในกลุ่มเดียวกันเท่านั้น

4.8.7.5 ค่าพารามิเตอร์ของโมเดล

พารามิเตอร์ของโพรเซสเซอร์

- จำนวนของโพรเซสเซอร์
- จำนวนกลุ่มข้างๆ
- ความเร็วของโพรเซสเซอร์

พารามิเตอร์ของโปรแกรม

- ขนาดของข้อมูล
- จำนวนการวนรอบของลูป
- จำนวนงานในแต่ละรอบ
- การสื่อสารข้อมูล
- เวลาต่อรอบ

พารามิเตอร์ของเครือข่าย

- Network Latency
- ช่องว่างการสื่อสาร
- วิธีการเชื่อมต่อของเครือข่าย
- โมเดลของโหนดภายนอก
- จำนวนโหนดสูงสุด
- ระยะเวลาของการมีชีวิตอยู่

บทที่ 5

เครื่องมือที่ใช้ในการตรวจสอบโปรแกรมประเภท DDoS

5.1 DDoSPing

โปรแกรม **DDoSPing** เป็นโปรแกรมที่ใช้ในการตรวจหาเครื่องที่เป็นเอเจนต์ของโปรแกรมประเภทที่ใช้โจมตีเพื่อให้ปิดบริการแบบกระจาย (DDoS : Distributed Denial Of Service) เช่น Wintrinoo, Trinoo, Stacheldraht และ TFN ซึ่งที่ได้ทดลองใช้นี้จะเป็นเวอร์ชัน 2.0 มีรูปแบบการทำงานเป็นแบบ GUI (Graphical User Interfaces) ซึ่งสามารถทำงานได้บน Win9x , WinNT

5.1.1 ลักษณะการทำงานของโปรแกรม

โปรแกรมจะทำการค้นหาเครื่องที่มีโปรแกรมประเภท DDoS อยู่ตามหมายเลขไอพีแอดเดรสที่ได้ระบุไว้เริ่มจากไอพีแอดเดรสเริ่มต้นไปจนถึงไอพีแอดเดรสสุดท้ายที่ได้ระบุเอาไว้ โดยสามารถจะระบุความเร็วของการสแกนตามประเภทความเร็วของเครือข่ายที่ใช้งานอยู่ได้ด้วย เมื่อมีเครื่องที่มีการทำงานของโปรแกรมประเภท DDoS อยู่ก็จะแสดงผลออกมาว่ามีเครื่องใดบ้างที่ทำงานอยู่ จากนั้นจะจัดเก็บผลของการรันโปรแกรมเอาไว้เป็นล็อกไฟล์ด้วย เพื่อสามารถนำมาวิเคราะห์ตรวจสอบได้ในภายหลัง

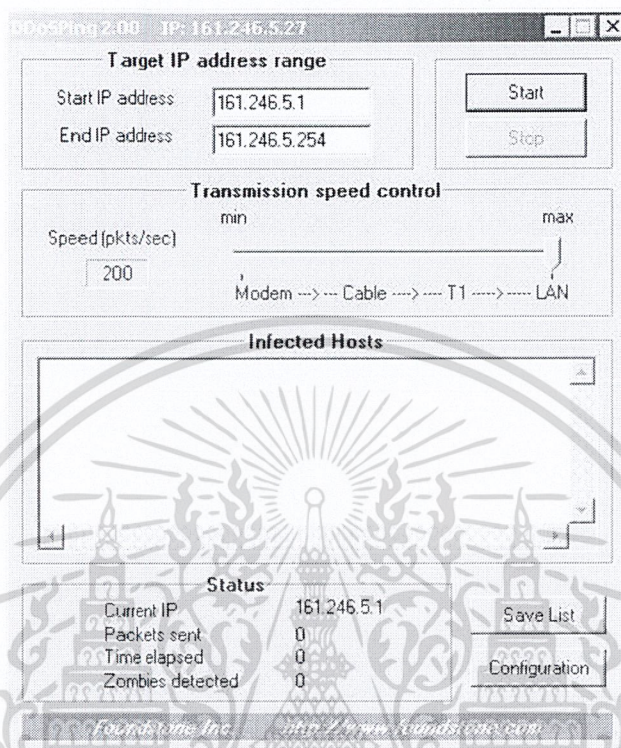
5.1.2 รูปแบบในการใช้งานโปรแกรม

เริ่มจากส่วนของ Target IP address range ซึ่งจะให้มีการระบุหมายเลขไอพีแอดเดรสโดยที่

- Start IP address : เป็นการให้ใส่หมายเลขไอพีแอดเดรสเริ่มต้นที่ต้องการสแกน
- End IP address : เป็นการให้ใส่หมายเลขไอพีสุดท้ายที่ต้องการสแกน

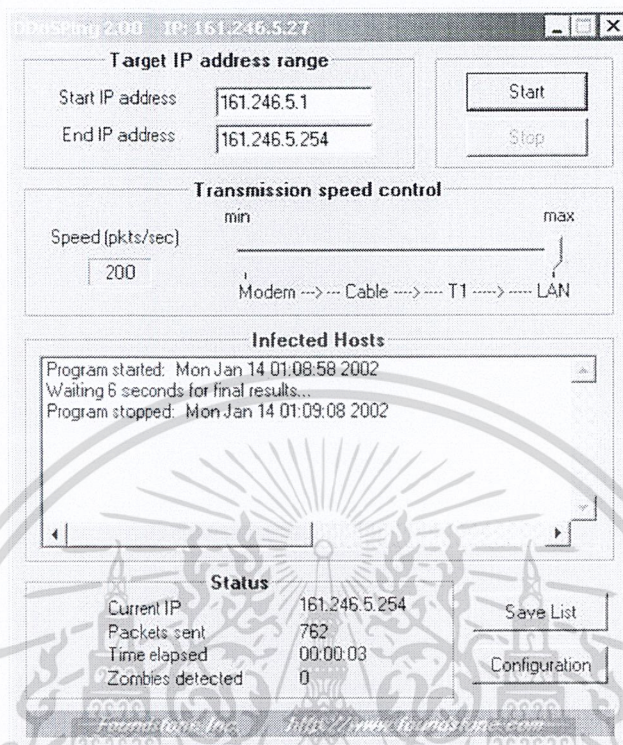
ในส่วนของ Start จะเป็นการเริ่มต้นให้โปรแกรมเริ่มทำงาน ส่วน Stop จะหยุดการทำงานของโปรแกรม

ส่วนของ Transmission speed control จะบอกถึงความเร็วในการติดต่อสื่อสารกับเครื่องที่ต้องการตรวจสอบโดยมีให้เลือกระดับของความเร็ว



รูปที่ 5-1 แสดงรูปแบบของโปรแกรม DDoSPing

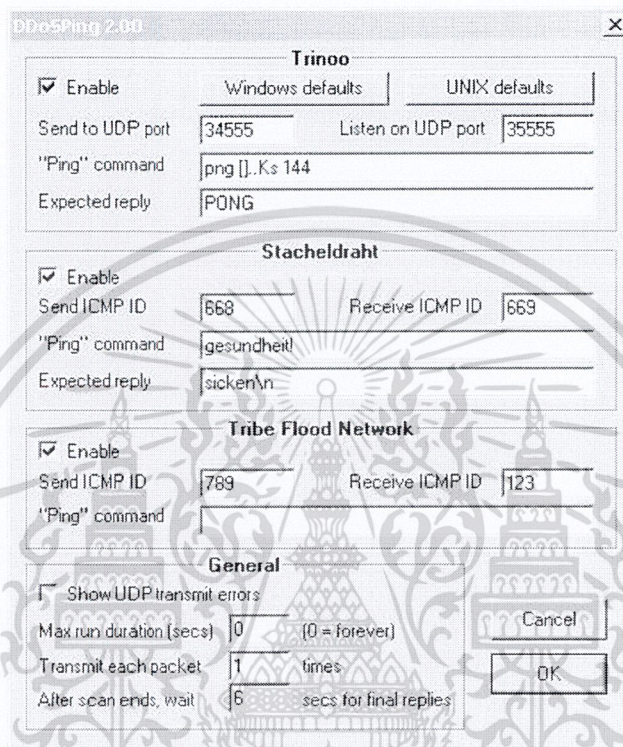
ส่วนของ Infected Hosts จะเป็นส่วนที่บอกถึงรายละเอียดที่เกิดขึ้นโดยจะบอกถึงเวลาเริ่มต้นการใช้งานโปรแกรมว่าเริ่มวันเวลาที่เท่าไรแล้วจะแสดงรายชื่อของโฮสต์ที่เป็นเอเจนต์ของโปรแกรมประเภท DDoS จากนั้นก็จะแสดงวันเวลาสิ้นสุดที่ได้มีการสแกนหา



รูปที่ 5-2 แสดงผลจากการสแกนของโปรแกรม DDoSPing

ส่วนสุดท้ายก็คือส่วนของ Status ซึ่งเป็นส่วนที่บอกถึงสถานะการทำงานของโปรแกรม จะมีส่วนของ

- Current IP : หมายเลขไอพีที่ได้ทำการติดต่อในขณะนั้น
- Packets sent : จำนวนแพ็กเก็ตที่ส่งไปทั้งหมดในขณะนั้น
- Time elapsed : เวลาที่เหลืออยู่ที่ใช้ในการสแกน
- Zombies detected : จำนวน zombie ทั้งหมดที่ได้สแกนเจอ
- Save List : สามารถที่จะทำการจัดเก็บผลของการสแกนไว้เป็นไฟล์
- Configuration : สามารถทำการ Configure ส่วนที่ต้องการค้นหาได้



DDosPing 2.00

Trinoo

☒ Enable Windows defaults UNIX defaults

Send to UDP port: 34555 Listen on UDP port: 35555

"Ping" command: png [].Ks 144

Expected reply: PONG

Stacheldraht

☒ Enable

Send ICMP ID: 668 Receive ICMP ID: 669

"Ping" command: gesundheit!

Expected reply: sicken\n

Tribe Flood Network

☒ Enable

Send ICMP ID: 789 Receive ICMP ID: 123

"Ping" command:

General

☐ Show UDP transmit errors

Max run duration (secs): 0 (0 = forever)

Transmit each packet: 1 times

After scan ends, wait: 6 secs for final replies

Cancel

OK

รูปที่ 5-3 แสดงการเลือกทำงานแบบ Windows defaults

จากรูปจะเป็นการเลือกการทำงานแบบ Windows defaults โดยจะมีอปชั่นต่างๆให้สามารถแก้ไขได้ตามการตรวจสอบประเภทของทูล โดยสามารถทำการเปลี่ยนพอร์ตที่ต้องการตรวจสอบ, คำสั่ง และการตอบกลับจากทูลได้

BDoSPing 2.00

Trinoo

☒ Enable Windows defaults UNIX defaults

Send to UDP port: 27444 Listen on UDP port: 31335

"Ping" command: png 144ads! Expected reply: PONG

Stacheldraht

☒ Enable

Send ICMP ID: 668 Receive ICMP ID: 669

"Ping" command: gesundheit! Expected reply: sicken\n

Tribe Flood Network

☒ Enable

Send ICMP ID: 789 Receive ICMP ID: 123

"Ping" command: Expected reply:

General

☐ Show UDP transmit errors

Max run duration (secs): 0 (0 = forever)

Transmit each packet: 1 times

After scan ends, wait: 6 secs for final replies

Cancel OK

รูปที่ 5-4 แสดงการเลือกทำงานแบบ *Unix defaults*

จากรูปจะเป็นการเลือกการทำงานแบบ UNIX defaults ซึ่งการทำงานจะคล้ายกันกับการตรวจสอบบนวินโดวส์

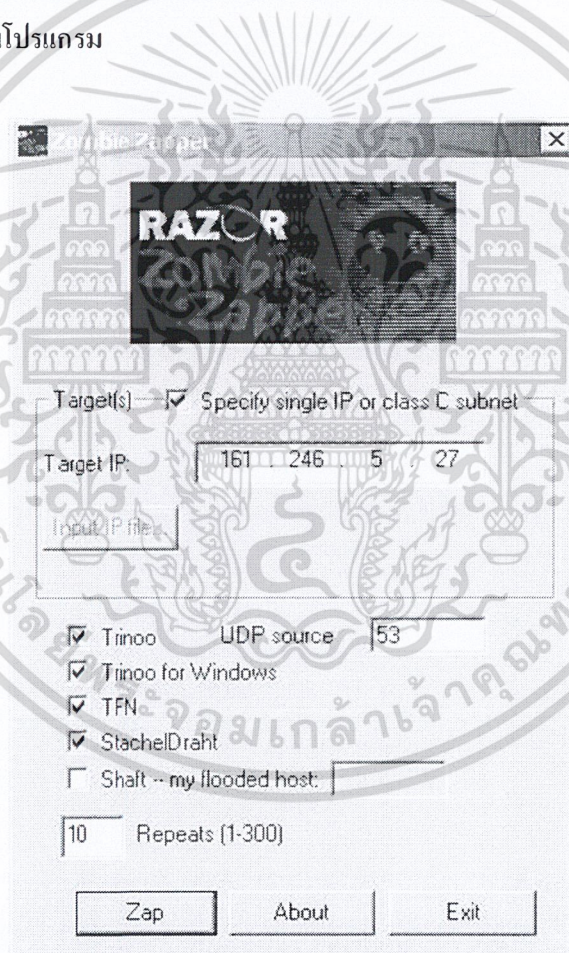
5.2 Zombie Zapper

โปรแกรม **Zombie Zapper** เป็นโปรแกรมที่ใช้ในการหยุดการทำงานของเครื่องที่เป็นเอเจนต์ของเครื่องมือที่ใช้ในการโจมตีประเภท DDoS ซึ่งสามารถหยุดการทำงานของ Trinoo , TFN , Stacheldraht ,และ Shaft

5.2.1 ลักษณะการทำงานของโปรแกรม

โปรแกรมจะทำการส่งคำสั่งหยุดการโจมตีไปยังเครื่องที่ได้ทำการระบุหมายเลขไอพีแอดเดรสไว้ซึ่งหากเครื่องนั้นเป็นเอเจนต์และได้มีการโจมตีไปยังเครื่องอื่นหรือไม่ก็ตาม เครื่องที่เป็นเอเจนต์จะหยุดการโจมตีในทันที สามารถใช้งานบนวินโดวส์ได้ด้วย

5.2.2 รูปแบบในการใช้งานโปรแกรม

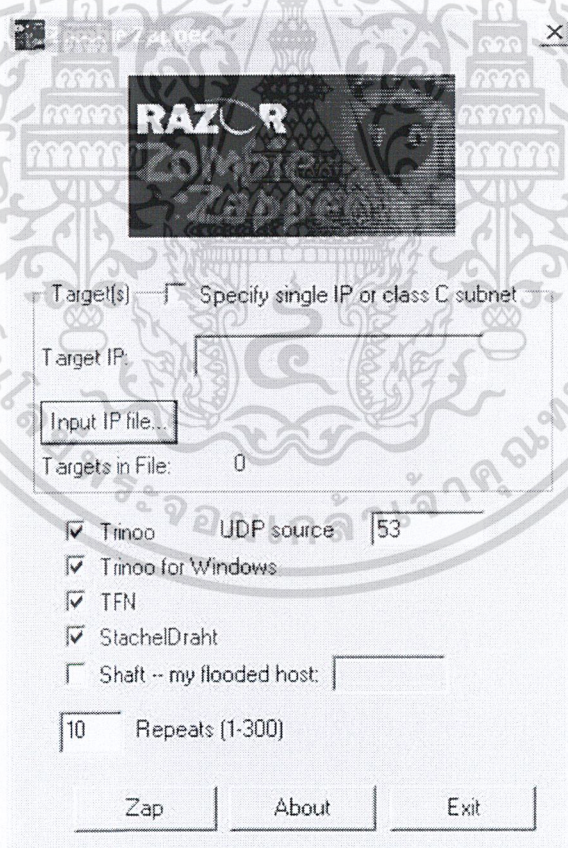


รูปที่ 5-5 แสดงการใช้งานของโปรแกรม *Zombie Zapper*

จากรูปจะเห็นว่ามีส่วนที่สามารถระบุหมายเลขไอพีแอดเดรสที่ต้องการตรวจหาโดยตรงได้ หรือไม่ หากต้องการทำการตรวจหาในหลายๆเครื่องก็ต้องมีรายชื่อเครื่องที่ต้องการค้นหาเก็บไว้เป็นไฟล์ จากนั้นจึงคลิกเช็คบ็อกซ์ตรงส่วนของ Specify single IP or class C subnet ออก แล้วโปรแกรมจะให้มีการเลือก Input IP file... ดังนั้นเราจึงมาสารดที่จะเลือกไฟล์ที่ได้สร้างรายชื่อของหมายเลขไอพีแอดเดรสทั้งหมดที่ต้องการตรวจสอบได้

ต่อไปเป็นส่วนของการเลือกว่าต้องการตรวจหาโปรแกรมประเภท DDoS ไດบ้าง ซึ่งจะเห็นว่ามีเช็คบ็อกซ์ที่ทำการเลือกได้โดยจะมี Trinoo , Trinoo for Windows , TFN , Stacheldraht และ Shaft เมื่อได้เลือกเสร็จแล้วก็จะมีส่วนที่ระบุว่าการส่งจำนวนของแพ็กเก็ตเป็นจำนวนเท่าไร โดยที่จำนวนที่สูงที่สุดที่สามารถทำได้คือ 300

- Zap : เริ่มการทำงานของโปรแกรม Zombie Zapper
- About : อธิบายเกี่ยวกับโปรแกรม Zombie Zapper
- Exit : ออกจากการทำงานของโปรแกรม Zombie Zapper



รูปที่ 5-6 แสดงการเลือกใช้งานในแบบ Input IP file

จากรูปนี้จะอธิบายถึงกรณีของการที่ต้องการระบุหมายเลขไอพีแอดเดรสโดยการอ่านจากไฟล์ที่ได้ทำการเก็บรายชื่อหมายเลขไอพีของเครื่องที่ต้องการค้นหาไว้แล้ว ซึ่งจะต้องคลิกเครื่องหมายถูกหน้า Specify single IP or class C subnet ออก

5.3 Find_ddos

โปรแกรม **find_ddos** เป็นอีกหนึ่งโปรแกรมที่ใช้สำหรับการสแกนหาเครื่องที่เป็นเอเจนต์และมาสเตอร์ของโปรแกรมประเภท DDoS ในการสแกนจากโปรเซสจะทำงานได้บนระบบ Solaris 2.6 ขึ้นไป และการสแกนจากไฟล์สามารถที่จะใช้งานบนระบบ Solaris 2.x ขึ้นไป

5.3.1 ลักษณะการทำงานของโปรแกรม

โปรแกรม **find_ddos** จะทำการค้นหาโดยการเริ่มจากการสแกนในโปรเซสการทำงานของเครื่องนั้นๆ ที่ได้มีการรันโปรแกรม **find_ddos** ว่าเป็นโปรเซสใดบ้างที่มีการทำงานของโปรแกรมประเภท DDoS อยู่หรือไม่ ถ้าหากพบก็จะแสดงผลออกมาให้ดูว่ามีโปรเซสใดบ้างที่เป็นโปรแกรมประเภท DDoS ทำงานอยู่ จากนั้นก็จะค้นหาจากไคลเอนต์ที่ทั้งหมดภายในเครื่องว่ามีโปรแกรมประเภท DDoS อยู่ในไคลเอนต์ไหนบ้าง

โปรแกรม **find_ddos** สามารถที่จะค้นหาเครื่องมือที่เป็น DDoS ได้ดังนี้

- mstream master
- mstream server
- stacheldraht client
- stacheldraht daemon
- stacheldraht master
- tfn-rush client
- tfn client
- tfn daemon
- tfn2k client
- tfn2k daemon
- trinoo daemon
- trinoo master

5.3.2 รูปแบบในการใช้งานโปรแกรม

การใช้งานของโปรแกรม **find_ddos** ในการรันโปรแกรมจะต้องเป็น root ของเครื่องเท่านั้น โดยมีรูปแบบของคำสั่งดังนี้

```
./find_ddos [-g grabdir] [-l logfile] [-p] [-v] [-V] [-x exclude1] [scandir]
```

โดยอธิบายพารามิเตอร์ต่างๆ ได้ดังนี้

- g grabdir : เลือกไดเรกทอรี, ชื่อไฟล์ ที่ไม่ได้ถูกใช้งานตามปกติ
- l logfile : ใช้สำหรับระบุชื่อไฟล์ที่ต้องการเก็บผลของการรันโปรแกรม find_ddos
- p : ใช้สำหรับบอกให้โปรแกรมทำการสแกนเฉพาะจากโปรเซสเท่านั้น
- v : สั่งให้ทูลบ่งบอกว่า ให้ตอบกลับมาว่า ขณะนี้ทำงานอยู่ในขั้นตอนใด
- V : แสดงเวอร์ชันของข้อมูล
- x exclude1 : ระบุให้สแกนข้ามไดเรกทอรีที่ไม่ต้องการสแกนได้
- scandir : ระบุประเภทของไฟล์ระบบที่ต้องการสแกนได้

จากนั้น โปรแกรมจะทำการถามว่าต้องการจะทำงานหรือไม่ ดังนั้นจึงต้องพิมพ์อักษร y แล้วก็กด Enter โปรแกรมก็ทำการค้นหาในส่วนของโปรเซสก่อน แล้วจึงค้นหาไฟล์จากไดเรกทอรีทั้งหมดของเครื่อง ซึ่งจะใช้เวลาพอสมควรในการที่โปรแกรมจะสามารถรันจนเสร็จ




```
[root@Techlinux find_ddos]# ./find_ddos
Running with the following default options:
./find_ddos -g files -l LOG -p /tmp /
```

This material and tool is furnished on an "as is" basis. There are no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. There is no warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

If the program "find_ddos" generates a log file revealing unlawful access of a tested system, system administrators are encouraged to contact their local FBI field office or the NIPC.

Please review the README file before continuing.

Do you accept these terms? y

In the future, you can run this tool with the "-y" option from the command line to acknowledge these terms

```
Logging output to: LOG
Scanning running processes...
/proc/1928/exe: tfn daemon
/proc/1933/exe: mstream master
/proc/1935/exe: mstream server
/proc/1948/exe: stacheldraht master
/proc/1949/exe: stacheldraht master
/proc/1952/exe: stacheldraht daemon
/proc/1961/exe: trino daemon
/proc/2563/exe: stacheldraht daemon
/proc/2585/exe: stacheldraht master
/proc/2586/exe: stacheldraht master
Scanning "/tmp"...
Scanning "/"...
/home/gatealcus/td: tfn daemon
/home/nakae/stachelantigl/client/td: stacheldraht daemon
/home/nakae/stachelantigl/telnetc/client: stacheldraht client
/home/nakae/stachelantigl/mserv: stacheldraht master
/home/nakae/Mstream/a.out: mstream master
/root/TFN/tfn: tfn client
/root/TFN/tfn-rush: tfn-rush client
/root/TFN/td: tfn daemon
/root/TFN/tx: tfn daemon
/root/tfn2k/src/td: tfn2k daemon
/root/tfn2k/src/tfn.o: tfn2k client
/root/tfn2k/src/tfn: tfn2k client
/root/trino/daemon/TrinoDaemon: trino daemon
```

รูปที่ 5-7 แสดงการทำงานของโปรแกรม find_ddos

บทที่ 6

โปรแกรมที่พัฒนาขึ้น

6.1 อัลกอริทึมการทำงานของโปรแกรม

โปรแกรมจะมีอัลกอริทึมการทำงาน โดยอาศัยการทำงานเริ่มต้นของการติดต่อสื่อสารของโปรแกรม DDoS ต่างๆมาทำการตรวจหาโปรแกรมประเภทนี้ดังนี้

ตรวจหาโปรแกรม Trinoo จะส่งคำสั่ง “png” มีพาสเวิร์ดเป็น “144ads1” และจะมีการตอบกลับด้วย “PONG”

ตรวจหาโปรแกรม TFN จะใช้การตรวจสอบ โดยการส่ง “789” และจะมีการตอบกลับด้วย “123”

ตรวจหาโปรแกรม stacheldraht จะใช้การส่งคำสั่งด้วย “skillz” และจะมีการตอบกลับด้วย “ficken” จะมีพอร์ตการส่งเป็น 666 และตอบกลับผ่านพอร์ต 667

6.2 ฟังก์ชันต่างๆของโปรแกรม

โปรแกรมที่พัฒนาขึ้นมีความสามารถในการตรวจสอบเครื่องมือการปิดการให้บริการแบบต่างๆ โดยมีฟังก์ชันการทำงานหลักๆดังนี้

โปรแกรม IsagDD.c

- unsigned short ip_sum()

จุดมุ่งหมาย : ตรวจสอบสถานะของไอพีแอดเดรสที่รับเข้ามา

อาร์กิวเมนต์ : แอดเดรสของไอพีที่ตรวจสอบ, ความยาวของไอพีแอดเดรสนั้น

ขั้นตอนการทำงาน :

- (1) ถ้าความยาวที่รับเข้ามาไม่เกิน 1 ให้เพิ่มค่าบิตที่เข้ามา
- (2) ถ้าความยาวที่รับเข้ามาเป็น 1 ให้เพิ่มไอพีแอดเดรสเข้ากับค่าที่กำหนด
- (3) ทำการแปลงเป็นเลขไบนารี 32 บิต ในการตรวจสอบไอพี

ผลลัพธ์ : ไอพีแอดเดรส 32 บิตส่งค่าขึ้นให้ฟังก์ชันอื่นๆ

- void listener()

จุดมุ่งหมาย : รอรับแพ็กเก็ตตอบกลับจากไอพีแอดเดรสที่กำหนด

อาร์กิวเมนต์ : แพ็กเก็ตประเภท ไอซีเอ็มพี, ยูดีพี, ไอพีแอดเดรสที่รับเข้ามา

ขั้นตอนการทำงาน :

- (1) ตรวจสอบแพ็กเก็ตที่รับเข้ามาว่าเป็นแพ็กเก็ตที่ถูกต้องหรือไม่ ถ้าใช่ให้ตรวจสอบขั้นต่อไป และถ้าไม่ใช่ ให้ยกเลิกการตรวจสอบ และออกจากโปรแกรม

- (2) วนลูการตรวจสอบว่า มีการตอบกลับจากไอพีแอดเดรสตรงตามที่กำหนดไว้หรือไม่ ถ้าใช้พิมพ์แสดงบอกว่าไอพีแอดเดรสนั้น มีเครื่องมือประเภท DDos อยู่ แต่ถ้าไม่ใช่ให้วนรอการตรวจสอบต่อไป

ผลลัพธ์ : ตรวจสอบไอพีแอดเดรสที่ตอบกลับ และแสดงผลลัพธ์จากการตรวจสอบนั้น

- int usage()

จุดมุ่งหมาย : แสดงวิธีการใช้งานโปรแกรม

อาร์กิวเมนต์ : ไม่มี

ขั้นตอนการทำงาน :

- (1) พิมพ์แสดงการใช้งานโปรแกรม
- (2) ออกจากโปรแกรม

ผลลัพธ์ : แสดงการใช้งานโปรแกรม

- main()

จุดมุ่งหมาย : เป็นฟังก์ชันหลักในการเรียกใช้งานการตรวจสอบไอพีแอดเดรสที่กำหนด

อาร์กิวเมนต์ : จำนวนอาร์กิวเมนต์ที่รับ, ตัวแปรอาร์กิวเมนต์ต่างๆ

ขั้นตอนการทำงาน :

- (1) กำหนดค่าเริ่มต้นให้แฟล็กต่างๆที่ใช้งาน
- (2) ตรวจสอบอาร์กิวเมนต์ และค่าออฟชั่นต่างๆที่เข้ามา ถ้าถูกต้องให้ทำงานต่อไป แต่ถ้าไม่ถูกต้องให้แสดงการทำงาน และออกจากโปรแกรม
- (3) ตรวจสอบ ไอพีแอดเดรสที่รับเข้ามา ถ้าถูกต้องให้ทำงานต่อไป แต่ถ้าไม่ถูกต้องให้ออกจากโปรแกรม
- (4) เรียกการตรวจสอบไอพีแอดเดรส และรอการตอบกลับด้วยฟังก์ชัน listen()
- (5) ออกจากโปรแกรม

ผลลัพธ์ : รับค่าแฟล็กต่างๆและส่งไปยังฟังก์ชันต่างๆในการตรวจสอบ

6.3 คุณสมบัติของโปรแกรม

โปรแกรมนี้สามารถตรวจสอบเครื่องมือประเภท DDos ต่างๆได้ดังนี้

- (1) ตรวจสอบเครื่องมือของ TRINOO / WINTRINOO ได้
- (2) ตรวจสอบเครื่องมือของ TFN / TFN2K / TFN3K ได้
- (3) ตรวจสอบเครื่องมือของ STACHELDRAHT ได้
- (4) ตรวจสอบเครื่องมือของ MSTREAM ได้
- (5) ตรวจสอบเครื่องมือของ CARKO ได้

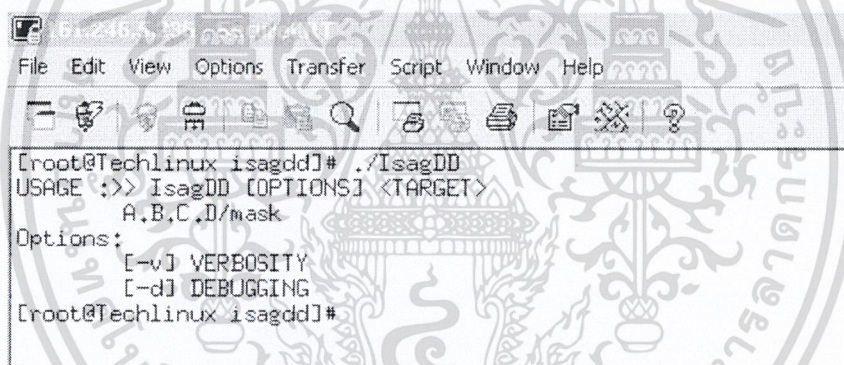
6.4 ข้อจำกัดของโปรแกรม

โปรแกรมที่พัฒนาขึ้นมีข้อจำกัดต่าง ๆ ดังนี้

- (1) โปรแกรมนี้สร้างขึ้นบนระบบปฏิบัติการลินุกซ์ ซึ่งอาจจะแตกต่างถ้านำไปใช้งานในระบบอื่น
- (2) ผู้ที่มีสิทธิ์รูท (root) เท่านั้นที่จะสามารถใช้งาน โปรแกรมนี้ได้
- (3) โปรแกรมนี้จะทำการตรวจสอบจากไอพีแอดเดรสที่กำหนดเท่านั้น

6.5 การใช้งานโปรแกรม

โปรแกรมที่พัฒนาขึ้นนี้ สามารถเรียกใช้ได้โดยการเรียก **IsagDD** โดยสามารถเลือกฟังก์ชันใช้งานได้ ดังนี้

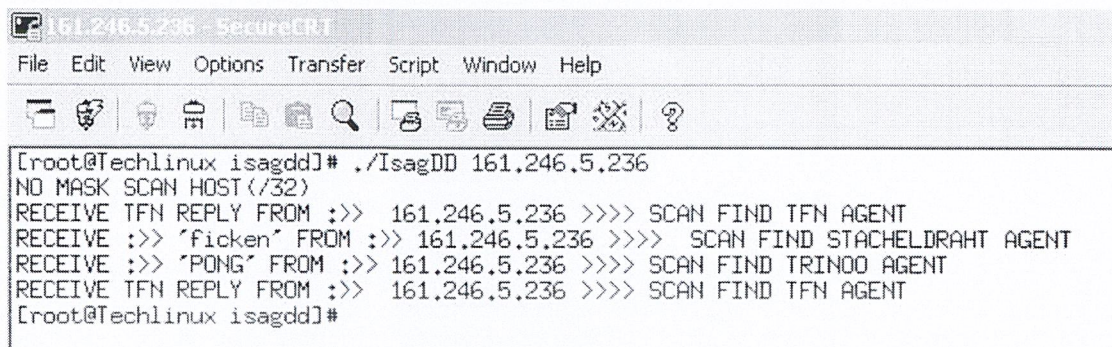


```

[61.246.5.138]
File Edit View Options Transfer Script Window Help
[root@Techlinux isagdd]# ./IsagDD
USAGE :>> IsagDD [OPTIONS] <TARGET>
          A.B.C.D/mask
Options:
  [-v] VERBOSITY
  [-d] DEBUGGING
[root@Techlinux isagdd]#
  
```

รูปที่ 6-1 แสดงการใช้งานโปรแกรม IsagDD

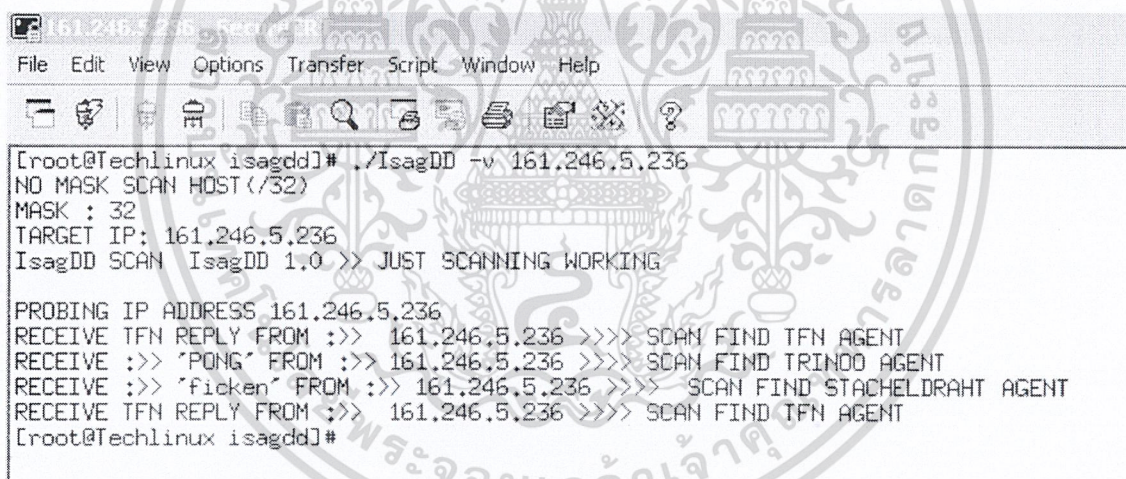
จากรูปแสดงการใช้งานโปรแกรม IsagDD ซึ่งเป็นหน้าจอช่วยเหลือในการใช้งานโปรแกรมที่ได้พัฒนาขึ้นมา ซึ่งจากรูปจะเห็นได้ว่า จะมีการใช้งานโดยการเรียกโปรแกรม IsagDD ตามด้วย Option ซึ่งมี 2 อย่าง คือ **-v** เป็นการตรวจสอบการแสดงผลการทำงานของโปรแกรม ส่วน **-d** จะเป็นการดีบั๊ก แสดงขั้นตอนการทำงาน และ **target** ซึ่งเป็น ไอพี ที่ต้องการตรวจสอบ



```
[root@Techlinux isagdd]# ./IsagDD 161.246.5.236
NO MASK SCAN HOST (/32)
RECEIVE TFN REPLY FROM :>> 161.246.5.236 >>>> SCAN FIND TFN AGENT
RECEIVE :>> "ficken" FROM :>> 161.246.5.236 >>>> SCAN FIND STACHELDRAHT AGENT
RECEIVE :>> "PONG" FROM :>> 161.246.5.236 >>>> SCAN FIND TRINOO AGENT
RECEIVE TFN REPLY FROM :>> 161.246.5.236 >>>> SCAN FIND TFN AGENT
[root@Techlinux isagdd]#
```

รูปที่ 6-2 แสดงตัวอย่างการใช้งานโปรแกรม IsagDD

จากรูปแสดงถึงความสามารถของโปรแกรมที่สามารถตรวจสอบหุลได้สองประเภท คือ trinoo เอเจนต์ และ tfn เอเจนต์ ซึ่งเป็นหุลสองตัวหลักๆของการโจมตีเพื่อปิดการให้บริการแบบกระจาย

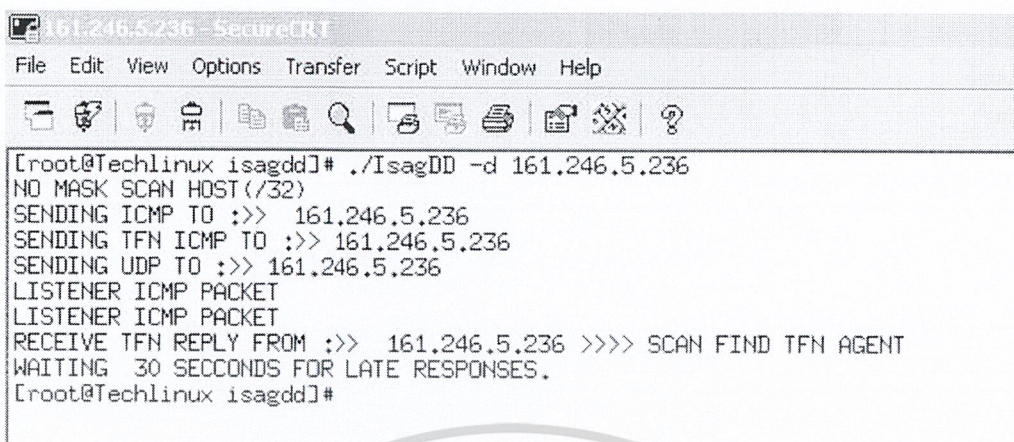


```
[root@Techlinux isagdd]# ./IsagDD -v 161.246.5.236
NO MASK SCAN HOST (/32)
MASK : 32
TARGET IP: 161.246.5.236
IsagDD SCAN IsagDD 1.0 >> JUST SCANNING WORKING

PROBING IP ADDRESS 161.246.5.236
RECEIVE TFN REPLY FROM :>> 161.246.5.236 >>>> SCAN FIND TFN AGENT
RECEIVE :>> "PONG" FROM :>> 161.246.5.236 >>>> SCAN FIND TRINOO AGENT
RECEIVE :>> "ficken" FROM :>> 161.246.5.236 >>>> SCAN FIND STACHELDRAHT AGENT
RECEIVE TFN REPLY FROM :>> 161.246.5.236 >>>> SCAN FIND TFN AGENT
[root@Techlinux isagdd]#
```

รูปที่ 6-3 แสดงตัวอย่างการเพิ่มเติมออพชั่น -v

จากรูป แสดงให้เห็นการตรวจสอบการทำงานของโปรแกรม



```

161.246.5.236 - SecureCRT
File Edit View Options Transfer Script Window Help

[root@Techlinux isagdd]# ./IsagDD -d 161.246.5.236
NO MASK SCAN HOST(/32)
SENDING ICMP TO :>>> 161.246.5.236
SENDING TFN ICMP TO :>>> 161.246.5.236
SENDING UDP TO :>>> 161.246.5.236
LISTENER ICMP PACKET
LISTENER ICMP PACKET
RECEIVE TFN REPLY FROM :>>> 161.246.5.236 >>>> SCAN FIND TFN AGENT
WAITING 30 SECONDS FOR LATE RESPONSES.
[root@Techlinux isagdd]#

```

รูปที่ 6-4 แสดงตัวอย่างการเพิ่มเติมออพชั่น -d

จากรูปเป็นการแสดงขั้นตอนการทำงานของโปรแกรมที่ละขั้นตอน โดยบ่งบอกถึงประเภทข้อบกพร่องที่ใช้ในการทำงานไปยังไอพีที่ตรวจสอบ ในที่นี้คือ 161.246.5.236 และรอการตอบกลับ

6.6 จุดเด่นและจุดบกพร่องในการทำงานของโปรแกรม

6.6.1 จุดเด่น

โปรแกรมที่พัฒนาขึ้นนี้มีลักษณะเด่นดังต่อไปนี้

- (1) สามารถตรวจสอบการทำงาน โดยอาศัยการเริ่มต้นการติดต่อสื่อสารของเครื่องมือต่างๆ เหล่านั้น
- (2) มีฟังก์ชันการทำงานที่ง่ายในการพัฒนาต่อไปได้

6.6.2 จุดบกพร่อง

โปรแกรมที่พัฒนาขึ้นนี้ยังมีข้อบกพร่องอยู่บ้าง ได้แก่

- (1) ไม่สามารถตรวจสอบเครื่องมือบางตัวที่นิยมใช้กันอยู่ได้
- (2) เครื่องมือบางตัวมีความซับซ้อนเพิ่มมากขึ้น ดังนั้นจึงจำเป็นต้องอาศัยการตรวจสอบด้วยวิธีการอื่นๆ ประกอบกันเพื่อให้ได้การตรวจสอบได้อย่างครบถ้วน

บทที่ 7

วิเคราะห์ผลการทดลองและสรุปผล

7.1 ปัญหาและอุปสรรค

ในการพัฒนาโปรแกรมตรวจจับเครื่องมือที่ใช้ในการปิดการให้บริการแบบกระจาย มีปัญหาและอุปสรรคในการพัฒนาหลายประการ ได้แก่

- (1) โปรแกรมที่ใช้ในโจมตีเพื่อปิดการให้บริการแบบกระจายนั้นมีมาก และแต่ละตัวก็มีการพัฒนาปรับปรุงเพื่อyakต่อการป้องกันในการโจมตี
- (2) ในการพัฒนายังต้องการเพิ่มรูปแบบในการป้องกันในรูปแบบอื่นๆเพิ่มเติมประกอบ

7.2 แนวทางการวิจัยและพัฒนาต่อ

โปรแกรมที่พัฒนาขึ้นยังสามารถพัฒนาต่อเพื่อให้มีความสามารถครอบคลุมการป้องกัน ค้นหาในรูปแบบอื่นๆได้อีกมาก อีกทั้งพัฒนาขึ้นใช้งานเฉพาะในระบบปฏิบัติการลินุกซ์เท่านั้น ดังนั้นจึงสามารถนำไปพัฒนาต่อได้โดย

- (1) เพิ่มความสามารถในการทำงาน ในครอบคลุมทั้งคลาสไอพีได้
- (2) เพิ่มการพัฒนาเพื่อติดต่อกับอุปกรณ์เครือข่ายอื่นๆได้ อาทิ เช่น เราเตอร์, ไฟร์วอลล์ เป็นต้น
- (3) เพิ่มความสามารถในการตรวจสอบ และป้องกันในรูปแบบอื่นๆประกอบกัน
- (4) พัฒนาต่อในรูปแบบออปเจ็กโอเรียนเท็ด เพื่อให้สามารถพัฒนาต่อได้ง่าย
- (5) พัฒนาให้โปรแกรมสามารถทำงานบนแพลตฟอร์มอื่นได้ เช่น Windows 9x, Windows NT, Windows 2000, Windows XP และ Windows Whistler เป็นต้น

7.3 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นกับเครื่องมืออื่นๆที่มีอยู่

ปัจจุบันเครื่องมืออื่นๆที่ออกมาสนับสนุนการตรวจจับการปิดการให้บริการในลักษณะนี้เท่าที่พบมี 3 โปรแกรม ได้แก่ DDosPing, Zombie Zapper และ Find_ddos ถึงแม้ว่าจะมีการตรวจจับและขอบเขตไม่เหมือนกันกับโปรแกรมที่ได้พัฒนาขึ้นนี้ แต่ก็มีจุดประสงค์ในการทำงานเหมือนกัน คือ ค้นหาและตรวจสอบเครื่องมือที่ปิดการให้บริการแบบกระจายได้ ซึ่งสามารถเปรียบเทียบกับโปรแกรมที่ได้พัฒนาขึ้นดังตารางนี้

โปรแกรมที่ใช้งาน	ผลการทดสอบ			
	DDosPing	Zombie Zapper	Find_ddos	IsagDD
1. Trinoo	✓	✓	✓	✓
2. Wintrinoo	✓	✓	✓	✓
3. TFN (2K,3K)	⊖	✓	✓	✓
4. Stacheldraht	⊖	✗	✓	✓
5. Shaft	✗	✗	✗	✗
6. Mstream	✗	✗	✓	✓
7. Carko	✗	✗	✗	✓

ตารางที่ 7-1 แสดงการเปรียบเทียบโปรแกรมกับเครื่องมืออื่น

ความหมายของสัญลักษณ์

- ✓ คือ สามารถตรวจสอบเครื่องมืออื่น ได้อย่างถูกต้อง
- ✗ คือ ไม่สามารถตรวจสอบได้อย่างถูกต้อง
- ⊖ คือ ตรวจสอบได้แต่ไม่ชัดเจน

จากตัวอย่างของการทดสอบเครื่องมือตรวจสอบต่างๆ ด้วยโปรแกรมที่นิยมใช้ในการปิดการให้บริการแบบกระจายแบบต่างๆ ปรากฏว่าโปรแกรมส่วนใหญ่ไม่สามารถตรวจสอบในการค้นหาเครื่องมือในรูปแบบของ Shaft ได้ เนื่องจากโปรแกรมที่พัฒนายังไม่ได้มีการปรับปรุงให้สามารถตรวจสอบการทำงานของเครื่องมือใหม่ๆ ที่ได้นำวิธีการปิดการให้บริการแบบกระจายนี้มาใช้

อีกทั้งรูปแบบการโจมตีเพื่อปิดการให้บริการแบบกระจาย ยังได้มีการขยายขีดความสามารถในการสร้างเครือข่ายในการโจมตีไปอีก อาทิ เช่น หนอนโค้ดเรด ที่ได้อาศัยช่องโหว่ของ IIS ของไมโครซอฟท์ วินโดวส์ ในการใช้เป็นเครือข่ายในการโจมตีได้

ซึ่งเป็นการยากที่จะพัฒนาโปรแกรมใดๆ ที่ให้มีความสามารถครอบคลุมการโจมตีเพื่อปิดการให้บริการแบบกระจายได้ในทุกรูปแบบ

บรรณานุกรม

- [1] Behrouz Forouzan, *“Introduction to Data Communications and networking”*, International Edition, McGraw-Hill, 1998.
- [2] Andrew S. Tanenbaum, *“Computer Network”*, 3th Edition, Prentice-Hall, 1996.
- [3] George Coulouris, Jean Pollimore and Tim Kindberg, *“Distributed Systems Concepts and Design”*, 3th Edition, Addison-Wesley, 2001.
- [4] Pankaj Mehra, Benjamin W Wah, *“Load Balancing an automated Learning Approach”*, World Scientific, 1964.
- [5] Rajkumar Buyya, *“High Performance Cluster Computing Volume 1 Programing and Application”*, PTR/PH, 1999.
- [6] Rajkumar Buyya, *“High Performance Cluster Computing Volume 2 Architecture and Systems”*, PTR/PH, 1999.
- [7] Randall K. Nichols, *“ICSA guide to CRYPTOGRAPHY”*, McGraw Hill, 1999.
- [8] Richard L. Petersen, Morgan Kaufmann, *“Unix Networking clearly explained”*, 1999.
- [9] Warren W. Gay, *“Linux Socket Programming by example”*, QUE, 2000.
- [10] W. Richard Stevens, *“Advanced Programming in the UNIX Environment”*, Addison Wesley, 1992.
- [11] W. Richard Stevens, *“UNIX Network Programming”*, Original U.S. edition, Prentice-Hall, 1990.
- [12] สันติ ศรีลาศักดิ์, วรุฒิ เทียงธรรม, *“เจาะประเด็นงานเขียนโปรแกรมบนลินุกซ์”*, ออฟเซ็ท เพรส, 2542
- [13] สุวัฒน์ ปุณณชัยยะ, ดัน ดันห์สุทธิวงศ์ และ สุพจน์ ปุณณชัยชนะ, *“เปิดโลกของ TCP/IP และโปรโตคอลของอินเทอร์เน็ต”*, โปรวิชั่น, 2543
- [14] เอกสิทธิ์ วิริยจารี, *“ปิดทางแฮกเกอร์”*, ซีเอ็ด, 2544

เว็บไซต์อ้างอิง

- [1] <http://www.cert.org>
- [2] <http://www.ibm.com>
- [3] <http://www.microsoft.com/security>
- [4] <http://www.packetstorm.com>
- [5] <http://www.programmersheaven.com>
- [6] <http://www.rootshell.com>
- [7] <http://www.securityfocus.com>

