ระบบตรวจจับบอทเน็ตจากพฤติกรรมการใช้งานเครือข่าย

NETWORK BEHAVIOR-BASED BOTNET DETECTION SYSTEM

จตุภัทร ชำชัยภูมิ

ชัยณรงค์ ไชยกุลวงศ์สันติ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบตรวจจับบอทเน็ตจากพฤติกรรมการใช้งานเครือข่าย

NETWORK BEHAVIOR-BASED BOTNET DETECTION SYSTEM

จุตฏิกูร ขำชัยภูมิ
ชัยณรงค์ ไชยกูลวงศ์สันติ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับบอทเน็ตจากพฤติกรรมการใช้งานเครือข่าย

NETWORK BEHAVIOR-BASED BOTNET DETECTION SYSTEM

ผู้จัดทำ

1. นายจุตฏิกูร ขำชัยภูมิ รหัสนักศึกษา 53010247
2. นายชัยณรงค์ ไชยกูลวงศ์สันติ์ รหัสนักศึกษา 53010335

.....อาจารย์ที่ปรึกษา

(ดร. ธนัญชัย ตรีภาค)

ระบบตรวจจับบอทเน็ตจากพฤติกรรมการใช้งานเครือข่าย

นายจุตฏิกูร ขำชัยภูมิ 53010247
นายชัยณรงค์ ไชยกูลวงศ์สันติ์ 53010335
ดร. ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2556

บทคัดย่อ

ในปัจจุบันเครื่องคอมพิวเตอร์ของผู้ใช้ตามบ้านหรือในองค์กรกำลังถูกแฮคเกอร์เข้าควบคุมและแปรสภาพเครื่องดังกล่าวให้ทำงานเป็น “bots”, “zombie” หรือ “drones” กล่าวคือทำการติดตั้งซอฟต์แวร์ที่ทำให้แฮคเกอร์สามารถควบคุมระยะไกลได้ โดยจะเรียกเครื่องคอมพิวเตอร์ต่างๆ ที่ ถูกแฮคเกอร์ควบคุมว่า “บอทเน็ต (Botnet)” ซึ่งวัตถุประสงค์ในการรวบรวมบอทเน็ตคือเพื่อใช้ในการประกอบกิจกรรมที่ขัดต่อกฎหมาย อย่างเช่น ส่งสแปมเมลล์ หรือ โจมตีโดยวิธี Denial of Service (DOS attack) จากปัญหาดังกล่าวผู้พัฒนาจึงทำการศึกษาและทำความเข้าใจกับบอทเน็ตรวมถึงรูปแบบพฤติกรรมที่บอทเน็ตกระทำกับเครื่องที่ตกเป็นเป้าหมาย ทั้งระบบการทำงานต่างๆ ของบอทเน็ต และพัฒนาโปรแกรมสำหรับวิเคราะห์และตรวจหาทราฟฟิคที่อาจจะเป็นของบอทเน็ตในเครือข่ายที่ดูแลได้

NETWORK BEHAVIOR-BASED BOTNET DETECTION SYSTEM

Mr. Juttikhun Khamchaiyaphum 53010247
Mr. Chainarong Chaikoonvongsun 53010335
Dr. Thanunchai Threepak Advisor
Academic Year 2013

ABSTRACT

Home or office computers are under threats from hackers who take over the control and alter the system to operate as "bots", "zombie", or "drones" by installing some remote control software enabling the hackers to control from afar. The controlled computers will be called "Botnet" and used by hackers collectively for illegal purposes such as sending spam mails or attacking by denial of service. From such problems, our team of developers have studied and understood about Botnet, including behavioral pattern that Botnet has on the victimized system, various working system of Botnet, and then has developed an application for analysis and searching for traffic that has a possibility to be Botnet in the network under supervision.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถดำเนินงานได้อย่างราบรื่นได้ด้วยความกรุณาของ อาจารย์ ดร. ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษาให้คำแนะนำ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาโดยตลอด

ขอขอบคุณห้องวิจัย ISAG สถานที่ที่ผู้วิจัยได้ทำงานวิจัยนี้และคำแนะนำต่างๆ ของเพื่อนๆ สมาชิกห้อง ISAG ทุกคน

ขอกราบขอบพระคุณอาจารย์คณะวิศวกรรมศาสตร์ สาขา วิศวกรรมคอมพิวเตอร์ ทุกท่านที่คอยประสิทธิ์ประสาทวิชาความรู้ทางด้านคอมพิวเตอร์ให้ข้าพเจ้าตั้งแต่อดีตจนถึงปัจจุบัน สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และผู้ปกครองที่ได้อบรมสั่งสอนให้ความรู้และเป็นกำลังใจให้เสมอมา

จุตฏิกูร ขำชัยภูมิ
ชัยณรงค์ ไชยกูลวงศ์สันติ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ปัจจุบันเครือข่ายคอมพิวเตอร์และอินเทอร์เน็ตมีการใช้งานเพิ่มขึ้นอย่างรวดเร็ว ระบบรักษาความปลอดภัยจึงเป็นเรื่องสำคัญอย่างมาก เพราะข้อมูลต่างๆ อาจเกิดการสูญหายหรือมีผู้อื่นทำการขโมยข้อมูลเหล่านั้นไป โดยใช้การโจมตีต่างๆ ซึ่งการโจมตีหรือภัยคุกคามมีรูปแบบที่แตกต่างทำให้สามารถจำแนกรูปแบบการโจมตีหรือภัยคุกคามเหล่านี้ได้ คือ Virus, Worm, Trojans, Backdoor, Spyware และ Phishing Botnets ซึ่งจากรูปแบบการโจมตีดังกล่าว บอทเน็ตนั้นถือว่าเป็นการโจมตีที่อันตรายและสร้างความเสียหายมากที่สุด

บอทเน็ตเป็นการโจมตีที่มีลักษณะวิธีการแพร่กระจายตนเองที่หลากหลาย ส่งผลให้เครื่องคอมพิวเตอร์ที่ถูกโจมตีทำงานภายใต้อำนาจควบคุมของผู้อื่น เมื่อได้รับคำสั่งจากผู้ที่ควบคุม หรือเรียกว่า บอทมาสเตอร์ เพื่อให้คอมพิวเตอร์ทำงานในลักษณะต่างๆ ให้บรรลุเป้าหมายตามที่ต้องการ โดยมีกลไกในการควบคุมและสั่งการที่แตกต่างกัน (เช่น IRC, HTTP, P2P) จะพบว่าบอทเน็ตไม่ได้เป็นเพียงภัยคุกคามต่อระบบเครือข่ายและอินเตอร์เน็ตเท่านั้น แต่ยังมีส่วนร่วมในรูปแบบอื่นๆ ของภัยคุกคามและการโจมตีอีกด้วย จากปัญหาต่างๆ ที่เกิดจากภัยคุกคามของบอทเน็ต ทำให้จุดประสงค์ในการพัฒนาโครงงานนี้คือ มีระบบที่สามารถตรวจจับบอทเน็ตโดยใช้หลักการวิเคราะห์พฤติกรรมของบอท ซึ่งในขั้นตอนการดำเนินงาน จะรวบรวมข้อมูลต่างๆ ในเครือข่าย แล้วทำการเก็บรวบรวมไว้ในฐานข้อมูลของเซิร์ฟเวอร์ จากนั้นจะนำข้อมูลมาตรวจสอบโดยใช้หลักการแบ่งกลุ่มเชิงลำดับชั้น (Hierarchical Clustering) เพื่อแบ่งลักษณะการใช้งานระหว่างกลุ่มพฤติกรรมของบอทกับกลุ่มพฤติกรรมของบุคคลใช้งานทั่วไปออกจากกัน

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบในการตรวจจับบอทต่างๆ โดยใช้หลักการวิเคราะห์พฤติกรรมของบอท
2) เพื่อให้ผู้ใช้ระบบสามารถล่วงรู้ทันเหตุการณ์ว่าเครื่องคอมพิวเตอร์นี้ถูกคุกคามโดยบอท

1.3 ขอบเขตของโครงงาน

1) พัฒนาแอพพลิเคชั่นสำหรับวิเคราะห์ทราฟฟิคที่ผิดปกติ โดยใช้สถิติของพฤติกรรมการใช้งานเครือข่าย เพื่อตรวจจับบอทเน็ตที่ทำงานบนวินโดว์เซิร์ฟเวอร์
2) ทำการตรวจจับบอทเน็ตที่ทำงานบนระบบเครือข่าย

1.4 วิธีการดำเนินการ

1) ศึกษาการทำงานของบอทเน็ต
2) ศึกษารูปแบบการตรวจจับแบบวิเคราะห์พฤติกรรม
3) ศึกษางานวิจัยของ Meisam Eslahi เรื่อง Improving HTTP-Based Botnet Detection by Using Network Behavior Analysis System
4) ศึกษาวิธีการจัดแบ่งกลุ่มของบอท Hierarchical Clustering (An Efficient Clustering Scheme to Exploit Hierarchical Data in Network Traffic Analysis : BIRCH)
5) จำลองการทำงานของบอทและจัดเก็บสถิติไว้ใช้ในการจัดกลุ่ม
6) พัฒนาโปรแกรมสำหรับคำนวณการจัดแบ่งกลุ่มของบอท
7) ทดลองใช้งานโปรแกรมที่พัฒนาขึ้นมา
8) สรุปผลการทดลองและแก้ไขปรับปรุงกรณีมีข้อผิดพลาด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้เกี่ยวกับบอทแต่ละประเภท
2) ได้รับความรู้เกี่ยวกับกระบวนการจัดแบ่งกลุ่มของบอทชนิดต่างๆ
3) ได้นำโปรแกรมตรวจจับบอทเน็ตไปใช้งานจริงในการตรวจจับบอทเน็ตบนระบบต่างๆ

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท โดยมีรายละเอียดดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึง ทฤษฎีพื้นฐานที่ใช้ในโคงงงานประกอบด้วยการแบ่งกลุ่มแบบเชิงลำดับชั้นด้วยกระบวนการของอัลกอริทึมเบิร์ชและการวิเคราะห์พฤติกรรมของเครือข่าย

บทที่ 3 การออกแบบและการพัฒนา กล่าวถึง รายละเอียดระบบที่พัฒนารูปแบบโครงสร้างของระบบและแนวคิดการวิเคราะห์ข้อมูล

บทที่ 4 การทดลองและผลการทดลองกล่าวถึง การทดลองโปรแกรมและการทดลองการทำงานของอัลกอริทึมที่ใช้

บทที่ 5 บทสรุป กล่าวถึงบทสรุปของโครงงาน ปัญหาและอุปสรรคต่างๆของโครงงาน แนวทางการแก้ไขและแนวทางการพัฒนาต่อไป

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บอทเน็ต

บอทเป็นโปรแกรมอัตโนมัติที่สามารถดำเนินการและทำงานด้วยคำสั่งซ้ำๆ กันได้อย่างรวดเร็ว มีจุดประสงค์หลักเพื่อดำเนินกิจกรรมที่เป็นอันตรายในเครือข่ายคอมพิวเตอร์ บอทจำนวนมากจะแพร่กระจายตนเองไปยังเครื่องคอมพิวเตอร์ต่างๆ และเชื่อมต่อตนเองกับผู้ใช้อื่นๆ ผ่านทางการใช้งานอินเทอร์เน็ต เมื่อมีบอทจำนวนมากทำงานในลักษณะเป็นกลุ่ม จะเรียกพฤติกรรมดังกล่าวว่า “บอทเน็ต (Botnet)” เครือข่ายของบอทหรือบอทเน็ตจะมีขนาดเครือข่ายตั้งแต่ขนาดเล็กๆ ระดับเพียงพันบอท จนถึงเครือข่ายขนาดใหญ่ซึ่งอาจมีจำนวนบอทมากถึงล้านบอท

บอทถูกออกแบบมาเพื่อทำให้เครื่องคอมพิวเตอร์ในเครือข่ายติดไวรัสหรือถูกคุกคาม เครื่องคอมพิวเตอร์ที่ถูกคุกคามจะกลายเป็นส่วนหนึ่งของบอทโดยที่เจ้าของนั้นไม่รับรู้ถึงภัยคุกคามที่เกิดขึ้นได้เลย ผู้ที่ควบคุมบอทเหล่านี้ เรียกว่า บอทมาสเตอร์ (Bot Master) จะทำการส่งคำสั่งประสงค์ร้ายต่างๆ ไปยังบอททั้งหมดผ่านทางอินเทอร์เน็ตและเซิร์ฟเวอร์ ที่มีชื่อว่า command and control (เรียกย่อๆว่า ซีแอนซี) เซิร์ฟเวอร์

วงจรชีวิตของบอทเน็ตสามารถแบ่งออกได้เป็น 4 ช่วง โดยช่วงสัปดาห์แรกของการถูกคุกคามเป้าหมายหลักของบอท คือ การรวบรวมข้อมูลที่สามารถเข้าถึงได้ทั้งหมดของเครื่องที่ถูกคุกคาม เช่น รหัสผ่านทางเว็บไซต์, อีเมลล์, บัญชีธนาคาร เป็นต้น โดยจะใช้วิธีการต่างๆ ในการรวบรวมข้อมูล เช่น การเปลี่ยนเส้นทางการเดินของข้อมูลไปยังฟิชชิ่งเว็บไซต์ (phishing websites) เป็นต้น ในช่วงสัปดาห์ที่ 2 บอทจะรวบรวมข้อมูลในส่วนของโลคอลเน็ตเวิร์ค (Local Network) เช่น ข้อมูลการใช้งานเครือข่ายในองค์กร หรือบริษัท ช่วงสัปดาห์ที่ 3 จะเป็นช่วงแพร่กระจายตนเองของบอท เช่น การส่งสแปม หรือ การแนบตัวเองไปพร้อมกับการส่งข้อความผ่านทางสื่อออนไลน์ต่างๆ และช่วงสัปดาห์ที่ 4 บอททั้งหมดจะถูกควบคุมโดยบอทตัวหลัก เพื่อรอรับคำสั่งสำหรับการโจมตีต่อไป

2.2 ลักษณะของบอทเน็ต

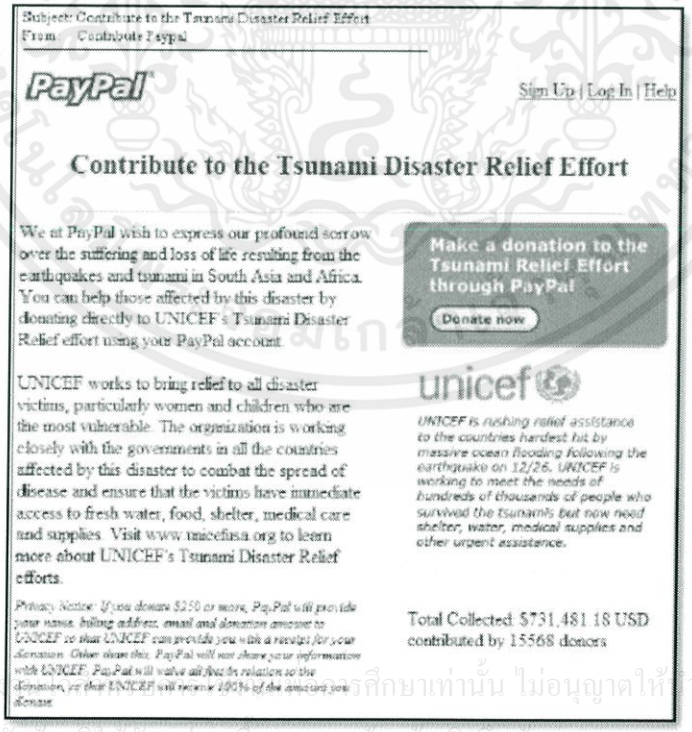
ลักษณะหลักของบอทเน็ต ระบุโดย เทร็นไมโคร (Trend Micro) และ Jose Nazario มีดังต่อไปนี้

2.2.1 การแพร่กระจายตนเอง

บอทเน็ตมีกรรมวิธีที่หลากหลายในการแพร่กระจายตนเองไปยังเครื่องคอมพิวเตอร์ โดยทั่วไปเรียกว่า เวิร์ม (Worm) ซึ่งลักษณะการแพร่กระจายตนเองของบอทมักจะมีพฤติกรรมแบบซ้ำเดิม Jose Nazario จึงแบ่งประเภทของการแพร่กระจายออกเป็น 4 รูปแบบดังต่อไปนี้

1) อีเมลล์ (Email)

อีเมลล์จำนวนมากอาจมีไฟล์บางไฟล์ที่บอทแฝงตัวแนบไปกับอีเมลล์ เพื่อจะทำการแพร่กระจายตนเอง หากไฟล์นั้นถูกเปิดใช้งาน อาจทำให้เครื่องคอมพิวเตอร์เครื่องนั้นถูกควบคุมได้ทันที ในอดีตนั้นเราสามารถทำการป้องกันได้ง่าย เนื่องจากอีเมลล์จำพวกนี้มักถูกส่งจากบุคคลที่เราไม่รู้จัก แต่ในปัจจุบันอีเมลล์เหล่านี้ใช้ความรู้ทางวิศวกรรมสังคม (Social Engineering) แฝงตัวเข้ามาโดยอาจอ้างว่าอีเมลล์นี้ถูกส่งมาจากแหล่งที่มาที่มีความน่าเชื่อถือ เช่น อีเมลล์มาจากธนาคาร หรือ บุคคลที่รู้จัก บ่อยครั้งที่เครื่องคอมพิวเตอร์ที่ถูกบุคคลอื่นขโมยข้อมูลที่อยู่ และส่งต่ออีเมลล์ไปยังที่อยู่นั้นๆ เพื่อทำการแพร่กระจายบอทออกไป รูปที่ 2.1 แสดงตัวอย่าง ฟิชชิ่งอีเมลล์ ที่ส่งไปยังเครื่องเหยื่อ โดยอ้างว่าเป็นการระดมเงินสำหรับภัยพิบัติสึนามิ จะเห็นได้ว่าเป็นเรื่องที่ยากที่จะสามารถแยกความแตกต่างระหว่างอีเมลล์จริงออกจากฟิชชิ่งอีเมลล์จำพวกนี้


Subject: Contribute to the Tsunami Disaster Relief Effort
From: Contribute Paypal

PayPal

Sign Up | Log In | Help

Contribute to the Tsunami Disaster Relief Effort

We at PayPal wish to express our profound sorrow over the suffering and loss of life resulting from the earthquakes and tsunami in South Asia and Africa. You can help those affected by this disaster by donating directly to UNICEF's Tsunami Disaster Relief effort using your PayPal account.

UNICEF works to bring relief to all disaster victims, particularly women and children who are the most vulnerable. The organization is working closely with the governments in all the countries affected by this disaster to combat the spread of disease and ensure that the victims have immediate access to fresh water, food, shelter, medical care and supplies. Visit www.unicefusa.org to learn more about UNICEF's Tsunami Disaster Relief efforts.

Privacy Notice: If you donate $250 or more, PayPal will provide your name, billing address, email and donation amount to UNICEF so that UNICEF can provide you with a receipt for your donation. Other than this, PayPal will not share your information with UNICEF. PayPal will waive all fees in relation to the donation, so that UNICEF will receive 100% of the amount you donate.

Make a donation to the Tsunami Relief Effort through PayPal

Donate now

unicef

UNICEF is rushing relief assistance to the countries hardest hit by massive ocean flooding following the earthquake on 12/26. UNICEF is working to meet the needs of hundreds of thousands of people who survived the tsunamis but now need shelter, water, medical supplies and other urgent assistance.

Total Collected: $731,481.18 USD
contributed by 15568 donors

รูปที่ 2.1 ตัวอย่างฟิชชิ่งอีเมลล์

2) ลิงค์สแปม (Link Spam)

ลิงค์สแปม คือ ยูอาร์แอล (URL) ที่มีเนื้อหาที่เป็นอันตรายซึ่งอาจได้มาโดยหลากหลายกรรมวิธี เช่น อีเมลล์, เว็บไซต์ออนไลน์ต่างๆ และแชทต่างๆ ทั้งนี้ทุกกรรมวิธีต้องมีการพยายามที่จะโน้มน้าวให้ผู้ใช้นั้นคิดว่าเป็นความจริง ข้อความเหล่านี้มักจะปรากฏขึ้นมาจากเพื่อนหรือองค์กรที่เชื่อถือได้ แต่ความจริงแล้วได้ถูกปลอมแปลงมาจากอีกบุคคลหนึ่ง

3) เว็บไซต์ (Website)

เว็บไซต์ที่ปลอมแปลงนั้นจะมีการติดตั้งตามเว็บที่เป็นที่รู้จักดี เช่น ยูทูป หรือ เว็บไซต์ที่เชื่อถือได้ที่มีการถูกแฮคและถูกฝังโค้ดที่เป็นอันตราย สามารถแบ่งเป็นการโจมตีได้ 2 ประเภท คือ การโจมตีฝั่งผู้ใช้ และการใช้ประโยชน์จากการดาวน์โหลด ในส่วนของการโจมตีฝั่งผู้ใช้นั้น ผู้ใช้จะมีการเข้าบริการเว็บไซต์ต่างๆ ที่ถูกฝังโค้ดที่เป็นอันตราย โค้ดจะทำงานโดยพยายามที่จะใช้ช่องโหว่ต่างๆ ของเว็บเบราเซอร์ในการเข้าถึงเครื่องของผู้ใช้งาน หากประสบผลสำเร็จ เครื่องของผู้ใช้งานจะถูกควบคุมโดยผู้อื่นทันที ส่วนการใช้ประโยชน์จากการดาวน์โหลดนั้น เมื่อเกิดการแจ้งเตือนในการดาวน์โหลดไฟล์ แต่ผู้ใช้นั้นยอมรับการดาวน์โหลดนั้น คอมพิวเตอร์ทำการรันไฟล์และส่งผลให้เครื่องคอมพิวเตอร์ถูกคุกคามในทันที

4) Exploits

วิธีการนี้จะคล้ายคลึงกับการโจมตีฝั่งผู้ใช้ แต่การใช้ช่องโหว่นั้นไม่ได้จำกัดว่าจะต้องมาจากเว็บเบราเซอร์เท่านั้น เมื่อเครื่องคอมพิวเตอร์ถูกคุกคาม ก็จะพยายามสแกนเครือข่ายเพื่อหาเครื่องคอมพิวเตอร์เครื่องอื่นๆ และจะทำการโจมตีหรือหารหัสผ่านของเครื่องที่ง่ายต่อการเข้าถึงต่อไป

2.2.2 พฤติกรรมการโจมตี

บอทเน็ตจะถูกสร้างขึ้นโดยผู้สร้างซึ่งมีการนำไปใช้งานในลักษณะที่แตกต่างกันดังต่อไปนี้

1) สแปมและฟิชชิ่งอีเมลล์

ในปี 2009 มีเมลล์ที่เป็นสแปมมากถึงแสนล้านฉบับถูกส่งทุกๆ วัน คิดเป็น 87.7 % ของอีเมลล์ทั้งหมด ซึ่ง 83% ของอีเมลล์สแปมนั้นถูกส่งมากจากบอทเน็ต สแปมเป็นปัญหาใหญ่ที่ไม่ใช่แค่เพียงสร้างความรำคาญให้กับผู้ใช้งานเพียงอย่างเดียว แต่ยังสามารถทำฟิชชิ่งข้อมูลต่างๆ ที่เป็นความลับได้อีกด้วย เช่น ชื่อผู้ใช้, รหัสผ่าน, หมายเลขบัตรเครดิต เป็นต้น

2) ดีดีโอเอส (DDoS)

การโจมตีแบบดีดีโอเอส (Distributed Denial of Service) เป็นปัญหาสำคัญอีกประการหนึ่งเพราะยากต่อการป้องกัน การโจมตีแบบดีโอเอส (DoS) จะเกิดขึ้นเมื่อเซิร์ฟเวอร์

ถูกร้องขอให้สร้างการเชื่อมต่อเป็นจำนวนมากจากไคลเอนท์ ทำให้มีปริมาณการไหลเวียนของข้อมูลบนเครือข่ายจำนวนมหาศาล จนกระทั่งเครื่องเซิร์ฟเวอร์ไม่สามารถตอบสนองได้ทั้งหมด ซึ่งแต่ก่อนนั้นการโจมตีชนิดนี้จะง่ายต่อการป้องกัน เพียงแค่ไฟร์วอลในการบล็อคไอพีแอดเดรสของผู้โจมตี แต่ในกรณีที่มีการร้องขอการเชื่อมต่อมาจากไคลเอนท์หลายที่ก็เป็นการยากที่จะทำการป้องกันได้อย่างเต็มประสิทธิภาพ

3) คีย์ล็อคกิ้ง (Key Logging)

บอทเน็ตจะได้รับข้อมูลจำนวนมากจากคอมพิวเตอร์ที่ถูกคุกคาม ข้อมูลทุกส่วนในเครือข่ายหรือแม้แต่ขณะที่พิมพ์ข้อมูลบางอย่างบนเครื่องคอมพิวเตอร์ก็สามารถถูกคุกคามได้ทันที หากเครื่องคอมพิวเตอร์นั้นมีการใช้งานคีย์ล็อคกิ้ง ซึ่งเป็นซอฟแวร์ที่จะบันทึกการกดแป้นพิมพ์ทุกอย่าง ไม่ว่าจะเป็นรหัสผ่านของเว็บไซต์ เช่น PayPal หรือ ธนาคาร เป็นต้น

4) Click Fraud

Click Fraud เป็นลักษณะการโกงของบอท เมื่อบอทเข้าไปแฝงตัวอยู่ในเว็บไซต์ต่างๆ จะแสร้งทำเป็นผู้ใช้ทั่วไป และจะคลิกโฆษณาต่างๆ โดยอ้างว่าทำกำไรให้กับเจ้าของเว็บไซต์ ซึ่งการฟลัดนี้จะไปนำเงินจากบริษัทโฆษณาออนไลน์ที่จ่ายเงินให้กับการคลิกโฆษณาแต่ละครั้ง ซึ่งเป็นการยากที่จะตรวจจับ เพราะทุกครั้งที่มีการคลิกนั้นจะมาจากต้นทางที่แตกต่างกัน เพราะมีการกระจายตัวไปทั่วอินเทอร์เน็ต

5) มัลแวร์ (Malware)

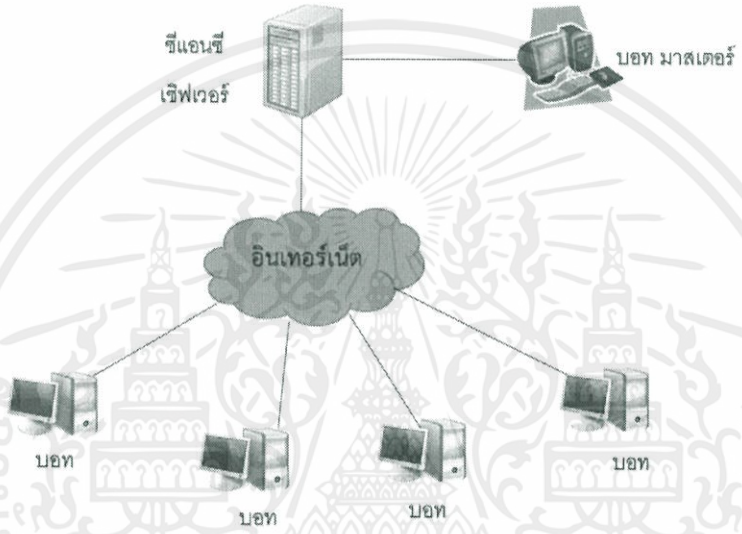
บอทเน็ตจะแพร่กระจายมัลแวร์เพื่อเพิ่มจำนวนเครื่องที่ถูกคุกคาม ซึ่งวิธีที่นิยมใช้คือผ่านทางอีเมลล์ แต่ในปัจจุบันนี้โซเชียลเน็ตเวิร์คและแอพพลิเคชั่นในการแชทต่างๆ เช่น เฟสบุ๊ค, ทวิตเตอร์ อาจมีการส่งข้อความที่ประกอบด้วยยูอาร์แอล ซึ่งจะนำไปสู่เว็บไซต์ที่อันตรายต่อผู้ใช้งาน

2.2.3 โครงสร้างในการใช้คำสั่งสื่อสารและควบคุมบอท

ซีแอนซีเซิร์ฟเวอร์ (Command and Control) จะคอยสั่งการและควบคุมการสื่อสารระหว่างบอทกับบอทมาสเตอร์ บอททั้งหมดในเครือข่ายจะติดต่อกันผ่านทางมาสเตอร์เซิร์ฟเวอร์ โดยการส่งข้อมูลที่เก็บรวบรวมได้ (collected information) เช่น รหัสผ่าน มายังมาสเตอร์เซิร์ฟเวอร์ นอกจากนี้ยังสามารถรับคำสั่งอื่นๆ จากเซิร์ฟเวอร์ได้ด้วย โครงสร้างของซีแอนซีที่ถูกใช้โดยบอทเน็ตสามารถแบ่งได้เป็น 4 ประเภทดังนี้

1) สตาร์ (Star)

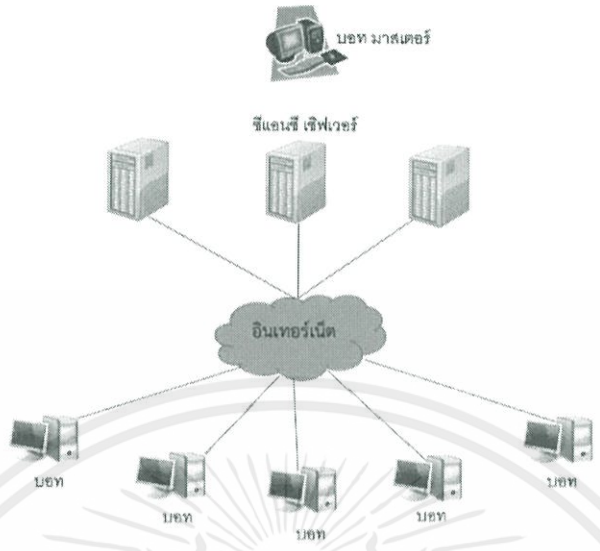
รูปแบบสตาร์จะทำให้บอททั้งหมดสื่อสารกันผ่านเซิร์ฟเวอร์ตัวเดียวกัน โครงสร้างแบบสตาร์จึงง่ายต่อการติดตั้งและการบำรุงรักษา ข้อเสียของรูปแบบสตาร์คือ ถ้าซีแอนซีเซิร์ฟเวอร์ไม่สามารถใช้งานได้ด้วยเหตุผลใดๆ ก็ตาม บอทจะยังสามารถทำงานได้บนเครื่องที่ถูกคุกคาม แต่จะไม่สามารถรับคำสั่งใดๆ และไม่สามารถส่งข้อมูลไปยังบอทมาสเตอร์ได้



รูปที่ 2.2 ตัวอย่างโครงสร้างสตาร์ของซีแอนซีบอทเน็ต

2) มัลติเซิร์ฟเวอร์ (Multi-Server)

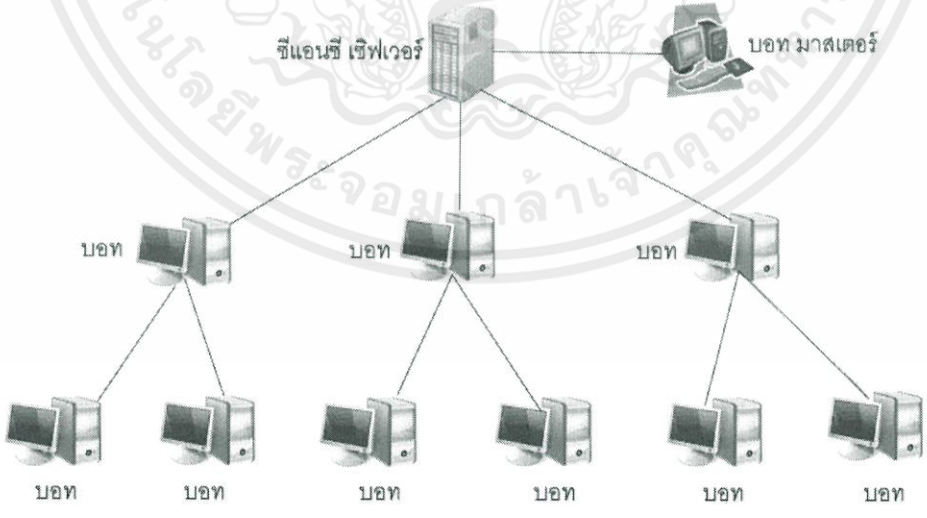
โครงสร้างมัลติเซิร์ฟเวอร์คล้ายกับโครงสร้างแบบสตาร์ แต่โครงสร้างมัลติเซิร์ฟเวอร์จะมีเซิร์ฟเวอร์อย่างน้อย 2 ตัวที่บอทนั้นสื่อสารด้วยแสดงดังรูปที่ 2.3 โครงสร้างแบบนี้จะลดปัญหาที่เกิดกับโครงสร้างแบบสตาร์ได้ เนื่องจากถ้าเซิร์ฟเวอร์หนึ่งไม่สามารถทำงานได้ด้วยเหตุผลใดก็ตาม เซิร์ฟเวอร์อื่นก็จะสามารถควบคุมบอทเหล่านั้นแทนได้



รูปที่ 2.3 ตัวอย่างโครงสร้างแบบมัลติเซิร์ฟเวอร์ของซีแอนซีบอทเน็ต

3) ลำดับชั้น (Hierarchical)

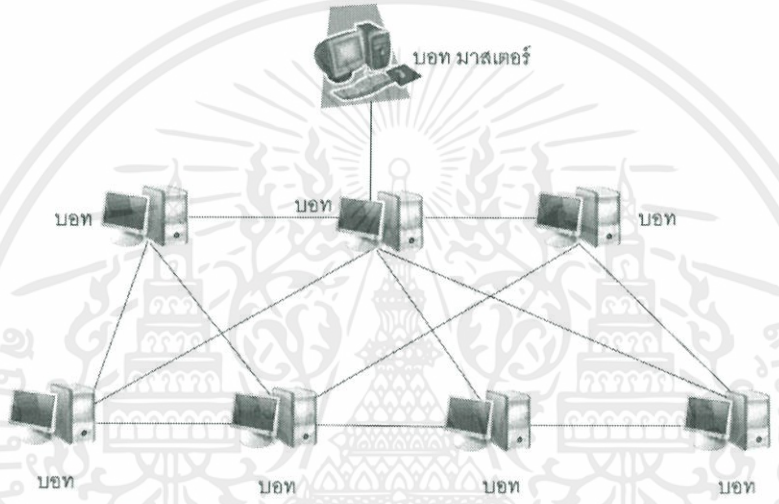
โครงสร้างแบบลำดับชั้นจะใช้โครงสร้างย่อยของผู้ที่อยู่ใต้บังคับบัญชาแทนการสั่งการผ่านซีแอนซีเซิร์ฟเวอร์เพียงตัวเดียว มีรูปแบบคล้ายทรี สามารถกระจายคำสั่งจากบอทตัวหนึ่งไปยังบอทตัวอื่นๆ ได้ดังรูปที่ 2.4 ข้อเสียของโครงสร้างแบบลำดับชั้น คือ บอทเพียงตัวเดียวไม่เพียงพอที่จะทราบสถานที่และจำนวนของบอททั้งหมดที่มีได้



รูปที่ 2.4 ตัวอย่างโครงสร้างแบบลำดับชั้นของซีแอนซีบอทเน็ต

4) สุ่ม (Random)

โครงสร้างแบบสุ่มเป็นโครงสร้างที่ไม่มีศูนย์กลางแต่จะติดต่อสื่อสารโดยตรงผ่านเซิร์ฟเวอร์ ดังแสดงในรูปที่ 2.5 ตัวอย่างนี้ของโครงสร้างแบบสุ่ม เช่น พีทูพี peer-to-peer (P2P) ซึ่งคำสั่งใน พีทูพี นั้นจะมีการแนะนำบอทในบอทเน็ตและสามารถแพร่กระจายไปยังบอททุกตัวแทนบอทตัวอื่นได้ โครงสร้างแบบสุ่มนี้ยากที่จะปิดตัวลง เนื่องจากไม่มีศูนย์กลางของเซิร์ฟเวอร์ซีแอนซี และมีข้อเสียคือการสื่อสารอาจล่าช้าและไม่สามารถคาดเดาได้



รูปที่ 2.5 ตัวอย่างโครงสร้างแบบสุ่มของซีแอนซีบอทเน็ต

2.2.4 โพรโทคอลในการสื่อสาร

การติดต่อสื่อสารกันระหว่างบอทเน็ตจะขึ้นอยู่กับโพรโทคอลที่ถูกนำมาใช้ รูปแบบและโครงสร้างของการสื่อสาร จะถูกใช้เพื่อส่งข้อมูลไปยังซีแอนซีเซิร์ฟเวอร์ และรอรับคำสั่งจากบอทมาสเตอร์ แต่เดิมมีการสื่อสารผ่านทางโพรโทคอลไออาร์ซี (Internet Relay Chat) โดยบอทจะเข้าไปทำงานร่วมกับไออาร์ซีเซิร์ฟเวอร์ที่มีการป้องกันแบบทั่วไปด้วยรหัสผ่าน บอทจะดักรับคำสั่งผ่านทางแชนแนลไออาร์ซี แต่การใช้งานเอสเอสแอล (SSL) ในไออาร์ซี เพื่อเข้ารหัสข้อมูลและคำสั่ง อาจสามารถหลีกเลี่ยงการตรวจจับของบอทได้ แต่ในปัจจุบันบอทส่วนมากใช้งานโพรโทคอลเอชทีทีพี (HTTP) เพราะไฟร์วอลล์นั้นมีการบล็อคทราฟฟิคที่มีการติดต่อกับซีแอนซีเซิร์ฟเวอร์ แต่เครือข่ายเกือบทั้งหมดจะอนุญาตให้โพรโทคอลเอชทีทีพีผ่านไปได้ เพราะใช้ในการเข้าถึงเว็บไซต์ และยิ่งไปกว่านั้น บางส่วนของบอทเน็ตมีการนำโพรโทคอลพีทูพี (Peer to Peer) มาใช้ในโครงสร้างแบบสุ่ม ทำให้ยากต่อการตรวจจับ และยากที่จะปิดตัวลงอีกด้วย ตัวอย่างบอทจำพวกพีทูพี ได้แก่ สปายบอท

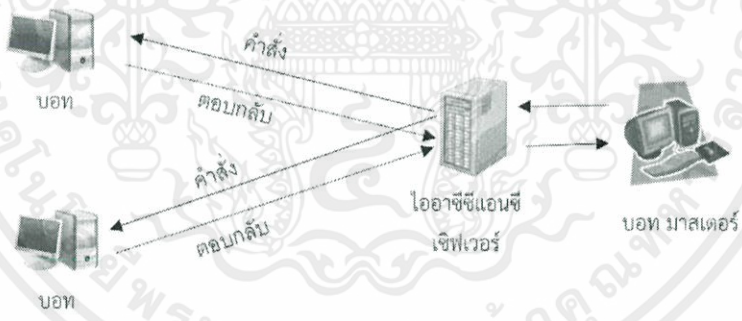
(SpyBot) และ อโกบอท (Agobot) อย่างไรก็ตามการใช้งานโพรโทคอลไออาร์ซีก็ยังเป็นที่นิยมมากที่สุด เนื่องมาจากมีสคริปต์ที่พร้อมใช้งาน และมีทรัพยากรวัสดุต่างๆ มากมาย

2.2.5 กลไกการควบคุมและสั่งการศูนย์กลาง

ซีแอนซีเซิร์ฟเวอร์ สามารถแบ่งประเภทโดยอยู่บนพื้นฐานของโพรโทคอลที่ใช้ในการสื่อสารและความสามารถในการสร้างการเชื่อมต่อระหว่างกันได้เป็น 2 ประเภท คือ ไออาร์ซี และ เอชทีทีพี

1) ไออาร์ซีบอทเน็ต (IRC-base Botnets)

ไออาร์ซีเป็นระบบที่ถูกใช้โดยผู้ใช้คอมพิวเตอร์ในการสื่อสารออนไลน์หรือพูดคุยกันแบบเรียลไทม์ วิธีนี้ถูกนำมาใช้ในบอทรุ่นแรก โดยบอทมาสเตอร์จะใช้ไออาร์ซีเซิร์ฟเวอร์และเชนแนลที่เกี่ยวข้องเพื่อกระจายคำสั่ง แต่ละบอทจะต้องติดต่อไปที่ไออาร์ซีเซิร์ฟเวอร์และแชนแนลที่บอทมาสเตอร์ได้เลือกและจะรอคำสั่งจากบอทมาสเตอร์ที่แชนแนลนี้ บอทมาสเตอร์จะสร้างการสื่อสารแบบเรียลไทม์ในทุกการเชื่อมต่อกับบอท และไออาร์ซีบอทจะคอยส่งคำสั่งอย่างเดียวซึ่งหมายความว่าเมื่อไออาร์ซีบอทเชื่อมต่อไปที่เชนแนลที่ได้เลือกไว้ บอทจะไม่ถูกตัดการเชื่อมต่อและยังคงสภาพในโหมดเชื่อมต่อตลอดเวลา ดังรูปที่ 2.6

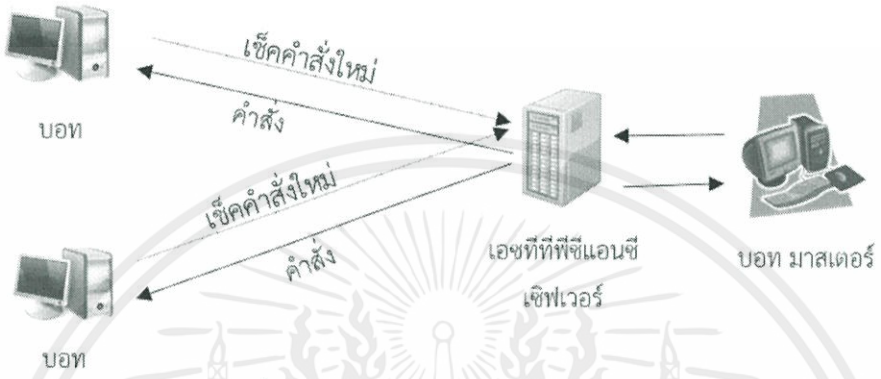


รูปที่ 2.6 ไออาร์ซีเบสซีแอนซีบอทเน็ต

2) เอชทีทีพีบอทเน็ต (HTTP-base Botnets)

เอชทีทีพีซีแอนซีเป็นเทคโนโลยีใหม่ที่ช่วยให้บอทมาสเตอร์สามารถควบคุมบอทโดยใช้เอชทีทีพีโพรโทคอล ในเทคนิคนี้บอทจะใช้ยูอาร์แอลที่เจาะจงหรือไอพีแอดเดรสที่กำหนดโดยบอทมาสเตอร์ เพื่อเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์เฉพาะ เอชทีทีพีบอทจะใช้วิธีการดึง (Pull) ซึ่งต่างจากวิธีการผลัก (Push) ของไออาร์ซีบอท ในวิธีการดึงนั้นเอชทีทีพีบอทจะไม่คงสถานะอยู่ในโหมดการเชื่อมต่อหลังจากที่สร้างการเชื่อมต่อไปที่ซีแอนซีเซิร์ฟเวอร์แล้ว ในขั้น

แรกในวิธีการดึง บอทมาสเตอร์จะเผยแพร่คำสั่งที่บางเซิร์ฟเวอร์ และบอทจะเข้าไปที่เว็บไซต์ของบอทเพื่ออัพเดทตัวเองหรือรับคำสั่งใหม่เป็นระยะๆ ด้วยกระบวนการที่ต่อเนื่องในช่วงเวลาปกติ ซึ่งจะถูกกำหนดโดยบอทมาสเตอร์



รูปที่ 2.7 เอชทีทีพีเบสซีแอนซีบอทเน็ต

2.3 เหตุผลที่เลือกเอชทีทีพีบอทเน็ตในการวิจัย

ในรุ่นแรกของบอทเน็ตเทคโนโลยีไออาร์ซีถูกใช้โดยบอทมาสเตอร์เพื่อควบคุมบอทเพราะระบบไออาร์ซีมีข้อได้เปรียบหลายอย่าง เช่น การใช้งานที่สะดวกทั้งการควบคุม,การจัดการ อย่างไรก็ตามจุดอ่อนของไออาร์ซีบอทเน็ตคือกลไกการควบคุมศูนย์กลางบอทเน็ตทั้งหมดสามารถถูกทำลายได้โดยการบล็อคไออาร์ซีเซิร์ฟเวอร์หรือบล็อคพอร์ตไออาร์ซี ดังนั้นบอทแบบพีทูพีจึงถูกออกแบบมาเพื่อแก้ปัญหานี้ในบอทแบบพีทูพีจะไม่มีศูนย์กลางตัวควบคุมและสั่งการ แต่จะมีการกระจายเซิร์ฟเวอร์คำสั่งจะถูกส่งไปหาบอทโดยบอทเอง นอกจากนี้บางวิธีการถอดรหัสถูกนำมาใช้เพื่อการสื่อสารที่ปลอดภัยเทคนิคนี้ทำให้ยากขึ้นสำหรับตรวจจับพีทูพีบอทเน็ตเมื่อเทียบกับไออาร์ซีบอทเน็ต อย่างไรก็ตามพีทูพีบอทเน็ตไม่ได้ถูกนำมาใช้อย่างกว้างขวางอย่างไออาร์ซีบอทเน็ตเพราะการดำเนินการและการควบคุมบอทเน็ตของพีทูพีนั้นค่อนข้างยากและซับซ้อน นอกจากนี้ยังมีความล่าช้าในการส่งคำสั่งระหว่างพีทูพีบอทด้วยกันและบอทมาสเตอร์ยังไม่สามารถที่จะรู้เกี่ยวกับสถานะของคำสั่งด้วย

ในปัจจุบันบอทมาสเตอร์ได้เริ่มใช้ศูนย์กลางเป็นซีแอนซีอีกครั้ง เอชทีทีพีโพรโทคอลถูกใช้แทนที่ไออาร์ซีโพรโทคอลและใช้พอร์ต 80 เพราะว่าความหลากหลายของการบริการ ทำให้ยากที่จะบล็อคศูนย์กลางซีแอนซีเซิร์ฟเวอร์นี้ นอกจากนี้ในการใช้เอชทีทีพีโพรโทคอลบอทจะซ่อนการสื่อสารผ่านทางกระแสของเอชทีทีพีปกติและจะหลีกเลี่ยงการตรวจจับโดยตัวตรวจจับเน็ตเวิร์ค อย่างเช่นไฟร์วอล เป็นต้น

จากการทบทวนในลักษณะของ ไออาร์ซี,พีทูพี และ เอชทีทีพีบอทเน็ตแล้วเป็นที่ชัดเจนว่า ซีแอนซีของเอชทีทีพีนั้นเป็นเทคโนโลยีที่ใหม่ที่เป็นที่ต้องการของบอทเน็ตเมื่อเทียบกับไออาร์ซีและพีทูพีบอทเน็ตแล้ว เอชทีทีพีเบสบอทเน็ตจะมีเซตของคุณสมบัติที่ทำให้เป็นเรื่องยากในการตรวจจับแต่เป็นเรื่องน่าแปลกใจที่จำนวนของงานวิจัยที่มุ่งเน้นไปที่การตรวจสอบของบอทเน็ตที่ใช้ เอชทีทีพี นั้นค่อนข้างต่ำเมื่อเทียบกับจำนวนของงานวิจัยเกี่ยวกับวิธีการตวจสอบสำหรับบอทเน็ตที่ที่ใช้ ไออาร์ซีหรือพีทูพี

2.4 วิธีการที่ใช้ในการตรวจสอบบอทเน็ต

วิธีการที่ใช้ในการตรวจสอบบอทเน็ตมีดังต่อไปนี้

2.4.1 ฮันนี่พอทและฮันนี่เน็ต

ฮันนี่พอทและฮันนี่เน็ต (Honeypot และ Honeynet) เป็นเครื่องมือที่ใช้เป็นกับดักสำหรับดักจับบอทที่สามารถตรวจจับได้ หรือรวบรวมข้อมูลของกิจกรรมของบอทได้ ข้อมูลที่สามารถนำมาใช้เพื่อทำความเข้าใจเกี่ยวกับพฤติกรรมของบอทหรือเจตนาของบอทมาสเตอร์ จะใช้วิธีการรวบรวมไบนารี่โค้ดของบอทและข้อมูลอื่นๆ ที่เกี่ยวกับบอท

2.4.2 ตรวจจับโดยลายเซ็น

ลายเซ็นหรือซิคเนเชอร์ (Signature) หมายถึงรูปแบบที่รู้จักกันหรือลักษณะของภัยคุกคามจากผู้บุกรุกเข้าสู่ระบบคอมพิวเตอร์ โดยจะวิเคราะห์และเปรียบเทียบรูปแบบเหล่านี้หรือลักษณะเหล่านี้ ซึ่งมีความเป็นไปได้ที่จะแยกแยะกิจกรรมที่เป็นภัยคุกคามจากกิจกรรมทั่วไป

การตรวจจับแบบวิธีลายเซ็นนั้นเป็นวิธีที่ประสิทธิภาพไม่ดีเท่าที่ควร เนื่องจากวิธีนี้ไม่สามารถระบุรูปแบบพฤติกรรมหรือลักษณะใหม่ๆ ของบอทได้ วิธีการนี้จะขึ้นอยู่กับการเปรียบเทียบข้อมูลที่เก็บรวบรวมไว้สำหรับที่รูปแบบที่รู้จักอยู่แล้ว ดังนั้นวิธีการนี้เป็นวิธีการที่ดีสำหรับตรวจสอบบอทที่รู้จักกันดี แต่ค่อนข้างไร้ประโยชน์สำหรับการตรวจสอบบอทใหม่

2.4.3 ตรวจจับโดยการตรวจสอบดีเอนเอส

การตรวจสอบและวิเคราะห์ทราฟฟิคด้วยดีเอนเอส (DNS Monitoring) ถูกนำมาใช้เป็นเทคนิคในการตรวจจับบอท โดยพบว่าบอทจะสร้างทราฟฟิคดีเอนเอสในบางสถานการณ์ เช่น เมื่อมีการระบุเซิร์ฟเวอร์ซีแอนซีหรือมีการเตรียมการโจมตี เช่น ดีดีโอเอส เป็นต้น

2.5 การตรวจจับโดยการวิเคราะห์พฤติกรรมของเครือข่าย

การตรวจจับโดยการวิเคราะห์พฤติกรรมของเครือข่าย (Detection Based on Network Behavior Analysis) เรียกย่อๆ ว่า เอนบีเอ (NBA) เป็นวิธีที่สามารถใช้ในการเก็บรวบรวมความหลากหลายของข้อมูล และสถิติเกี่ยวกับทราฟฟิคของเครือข่าย จากนั้นจะนำข้อมูลมาวิเคราะห์เพื่อตรวจจับสัญญาณที่เป็นภัยคุกคามหรือกิจกรรมที่เป็นอันตรายต่างๆ วิธีเอนบีเอประกอบไปด้วยองค์ประกอบที่หลากหลายรวมไปถึงเซ็นเซอร์และตัวจัดการเซิร์ฟเวอร์ด้วย โดยระบบเอบีเอนั้นจะรวบรวมข้อมูล เช่น ไอพีแอดเดรส, ระบบปฏิบัติการ, การบริการที่เปิดให้ใช้บริการ และข้อมูลที่ถูกบันทึกไว้ เช่น Timestamp, ประเภทของเหตุการณ์ (event type), โพรโทคอลเครือข่าย, พอร์ตโฮส และ ฟิลด์ของเฮดเดอร์แพ็คเก็ตสำหรับแต่ละไคลเอนต์

2.5.1 ตัวอย่างการตรวจจับโดยการวิเคราะห์พฤติกรรมตามช่วงของเวลา (Period)

พฤติกรรมของเอชทีทีพีเบสบอทเน็ต (HTTP-base Botnets) จะมีการเริ่มการเชื่อมต่อและร้องขอเอชทีทีพีเก็ต (HTTP GET) ไปยังเว็บไซต์ เพื่อร้องขอคำสั่ง หลังจากที่เว็บไซต์ของซีแอนซีได้รับคำร้องขอแล้วก็จะส่งคำสั่งกลับมาพร้อมกับเครื่องคอมพิวเตอร์ที่ติดบอท ข้อดีของวิธีนี้ คือ เว็บไซต์ซีแอนซีไม่จำเป็นต้องติดแทร็ก (Track) และเครื่องคอมพิวเตอร์ที่ติดบอทเพียงแค่รอเครื่องคอมพิวเตอร์ที่ติดบอทด้วยกันติดต่อกลับมาแทน ข้อเสียของวิธีนี้ คือ เว็บไซต์นั้นไม่สามารถควบคุมหรือรักษาการเชื่อมต่อของเครื่องคอมพิวเตอร์ที่ติดบอทได้จนกว่าเครื่องนั้นๆ จะติดต่อกลับมาที่เซิร์ฟเวอร์เอง

กลไกที่ทำให้เครื่องคอมพิวเตอร์ติดบอท คือ การฝังโค้ดที่เป็นอันตรายลงไป และมีการเรียกใช้คำสั่ง sleep() (คำสั่งที่ใช้ในการรอตามระยะเวลาที่กำหนดก่อนจะมารับคำสั่งใหม่ที่เว็บไซต์ซีแอนซี) ใน while loop ซึ่งแสดงได้ดังรูปที่ 2.8 แสดงตัวอย่างโค้ดที่อาจจะถูกนำมาใช้ และตารางที่ 2.1 แสดงตารางบันทึกเวลา (Timestamps) การเชื่อมต่อแอบเอชทีทีพีของแต่ละการเชื่อมต่อ

```
while (1) {
        get_and_process_cnc_commands();
        sleep(60);
}
...
void get_and_process_cnc_commands()
{
        // Insert code here to connect to C&C website, poll for a command
        // and process the command
}
```


รูปที่ 2.8 โค้ดแสดงกลไกการใช้คำสั่ง sleep() อย่างง่าย

ตารางที่ 2.1 ตารางเวลาในแต่ละช่วงเวลาที่บอทมีการติดต่อกับซีแอนซีเซิร์ฟเวอร์

Date/Time	Time Interval (Seconds)
2010-05-01 21:47:46	-
2010-05-01 22:47:50	3604
2010-05-01 23:47:53	3603
2010-05-01 00:47:55	3602
2010-05-01 01:47:58	3603
2010-05-01 02:47:01	3603
2010-05-01 03:47:04	3603
2010-05-01 04:47:07	3603
2010-05-01 05:47:10	3603
2010-05-01 06:47:13	3603
2010-05-01 07:47:16	3603
2010-05-01 08:47:19	3603
2010-05-01 09:47:22	3603
mean = 3603 std. dev = 0.43	

จากการสังเกตเครื่องคอมพิวเตอร์ที่ติดบอทจะมีพฤติกรรมในการติดต่อกลับมาที่ซีแอนซีเซิร์ฟเวอร์เป็นระยะๆ ทำให้สามารถนำพฤติกรรมการติดต่อดังกล่าว ซึ่งมีช่วงระยะเวลาเกือบจะคงที่มาใช้ในการคำนวนและตรวจจับการเชื่อมต่อที่มีระยะเวลา

2.5.2 เหตุผลที่เลือกวิธีการตรวจจับโดยใช้การวิเคราะห์พฤติกรรมเครือข่าย

- ความสามารถในการตรวจจับภัยคุกคามที่ไม่รู้จัก

  บอทมาสเตอร์จะทำการอัพเดทข้อมูลวันต่อวัน เพื่อหลีกเลี่ยงการตรวจจับ แต่ระบบเอนบีเอนั้นสามารถขัดขวางกลยุทธ์ต่างๆ ของบอทมาสเตอร์เหล่านี้ได้ และด้วยคุณสมบัติของเอนบีเอทำให้สามารถเพิ่มประสิทธิภาพในการตรวจจับบอทเน็ตได้

- ความสามารถในการตรวจจับภัยคุกคามที่มีการเข้ารหัส

  บอทเน็ตจะพยายามซ่อนทราฟฟิคของบอทกับทราฟฟิคทั่วไป หรือใช้วิธีการเข้ารหัส แต่วิธีเอนบีเอสามารถมองรูปแบบของการไหลของข้อมูลในเครือข่ายที่ผิดปกติได้ นอกจากนี้รายงานมาตรฐานจากกลุ่มอเบอร์ดีนชี้ให้เห็นว่าการตรวจจับแบบเอนบีเอสามารถให้ผลลัพธ์ที่ดีเมื่อมีการทำงานร่วมกับวิธีอื่น

2.6 เทคนิคการแบ่งกลุ่มแบบเชิงลำดับชั้น

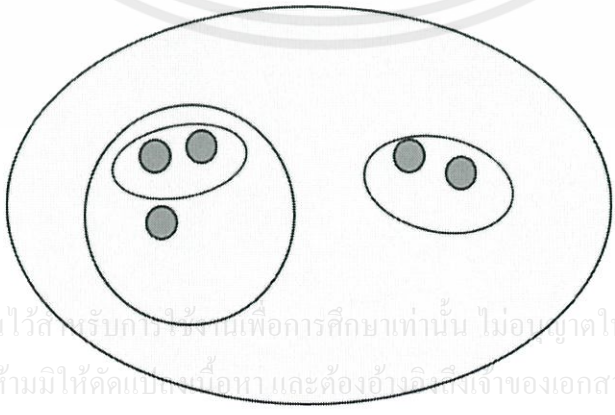
เทคนิคการแบ่งกลุ่มแบบเชิงลำดับชั้น (Hierarchical Cluster) มีเครื่องมือในการวิเคราะห์อย่างแพร่หลายเป็นจำนวนมาก แนวคิด คือ การสร้างบาลานซ์ทรีของข้อมูลที่เป็นลำดับ และรวมกลุ่มของจุดที่คล้ายกัน เพื่อสรุปการใช้งานของข้อมูล ข้อดีของเทคนิคแบบเชิงลำดับชั้นเมื่อเปรียบเทียบกับเทคนิคแบบเคมีน คือ ข้อมูลจำเป็นสำหรับเทคนิคแบบเค-มีน (K-mean) คือ จำนวนของกลุ่ม, การมอบหมายข้อมูลกับกลุ่ม, ระยะทางระหว่างข้อมูล แต่ข้อมูลสำหรับเทคนิคแบบเชิงลำดับชั้นต้องการเพียงตัวชี้วัดของความคล้ายคลึงกันระหว่างจุดของกลุ่มข้อมูล ในงานวิจัยนี้จึงได้เลือก เบิร์ชอัลกอริทึม (BIRCH Algorithm) ในการแบ่งกลุ่มของข้อมูล

2.6.1 ประโยชน์ของอัลกอริทึมเบิร์ช

1) อัลกอริทึมเบิร์ชอาศัยการจัดกลุ่มโดยการใช้ข้อมูลที่ตนมีอยู่ โดยไม่จำเป็นต้องสแกนข้อมูลทั้งหมดของระบบใหม่อีกครั้ง
2) อัลกอริทึมเบิร์ชสามารถสร้างกลุ่มของข้อมูลได้โดยอาศัยการสแกนระบบทั้งหมดเพียงครั้งเดียว
3) อัลกอริทึมเบิร์ชสามารถบริหารจัดการการใช้หน่วยความจำสำหรับการแบ่งกลุ่มข้อมูลได้

2.6.2 หลักการของอัลกอริทึมเบิร์ช

อัลกอริทึมเบิร์ชจะเริ่มแบ่งกลุ่มของข้อมูลที่กลุ่มของจุดๆ หนึ่ง (ทุกจุดในฐานข้อมูลในกลุ่ม) หลังจากนั้นภายในแต่ละกลุ่มจะหาจุดที่ใกล้ตนที่สุด แล้วรวมกันเป็นอีกกลุ่มหนึ่ง และทำแบบนี้ไปเรื่อยๆ จนกระทั่งเหลือเพียงกลุ่มใหญ่สุดกลุ่มเดียว การคำนวณของกลุ่มจะทำด้วยระยะทางเมทริค ($O(n^2)$large) และ เวลา $O(n^2)$



รูปที่ 2.9 การแบ่งกลุ่มแบบเชิงลำดับชั้น

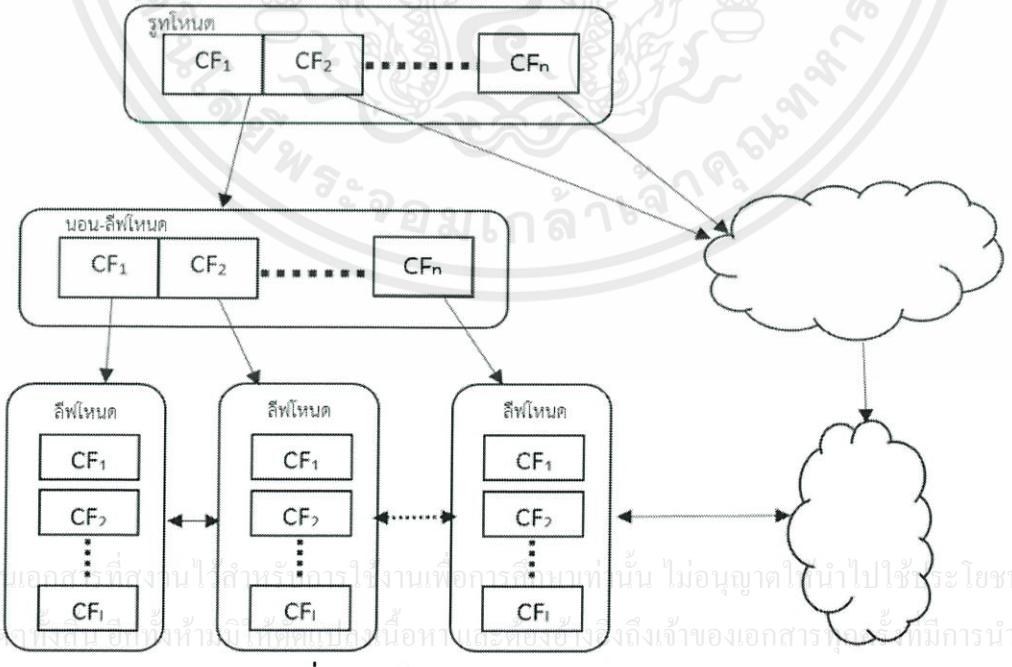
2.6.2.1 ลักษณะการแบ่งกลุ่ม

อัลกอริทึมเบิร์ชจะสร้างซีเอฟทรี (Clustering Feature Tree) ในขณะที่มีการสแกนชุดข้อมูล ซึ่งแต่ละเอนทรีในซีเอฟทรีจะแสดงกลุ่มของออปเจ็ค และลักษณะที่โดดเด่น 3 ประการ (N,LS,SS) โดยกำหนดให้ N เป็นเวกเตอร์ของจุดข้อมูลในกลุ่มที่มี d-มิติ และกำหนดให้ X_i (i = 1, 2, 3,..., N) เป็นซีเอฟเวกเตอร์ของกลุ่ม รูปแบบของ CF เป็น (N,LS,SS) โดยที่ N คือ จำนวนจุดของข้อมูลในกลุ่ม, LS คือ ผลรวมเชิงเส้นของ N จุดข้อมูล และ SS คือ ผลรวมตารางของ N จุดข้อมูล

2.6.2.2 ซีเอฟทรี

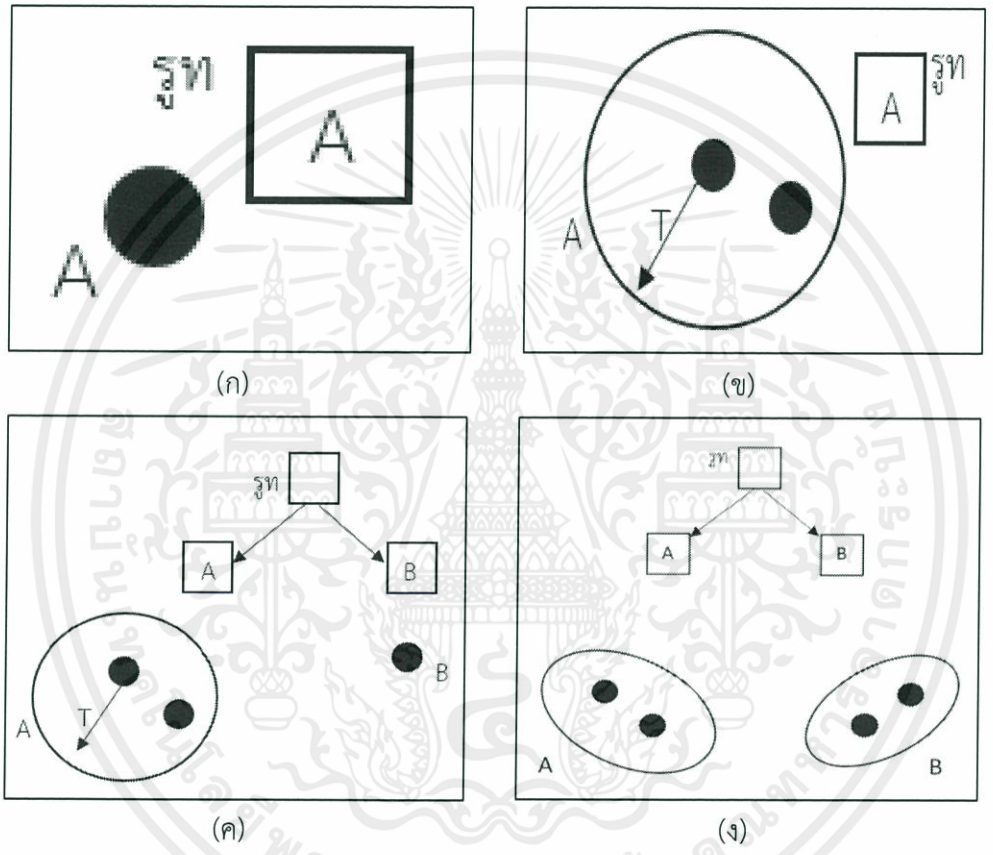
ความสูงของบาลานซ์บรีจะมี 2 ค่า คือ บรานช์เฟคเตอร์ B และ เทรชโฮล T ซึ่งแต่ละนอนลีฟโหนดจะมีเอนทรีได้มากสุด B เอนทรี ในรูป [CF_i, $child_i$] โดยที่ $child_i$ เป็นจุดที่ชี้ไปยังโหนดที่ i และ CF_i เป็นซีเอฟของกลุ่มย่อยที่ถูกแทนด้วย child นี้ ดังนั้น นอนลีฟโหนดจะแสดงให้เห็นถึงกลุ่มที่สร้างขึ้นจากกลุ่มย่อยทั้งหมดที่แสดงถึงเอนทรีของตัวเอง

ลีฟโหนดจะมีเอนทรีมากสุด L เอนทรี แต่ละโหนดจะมีรูปแบบ [CF_i] โดยที่ i = 1,2,...,L นอกจากนี้ยังมีตัวชี้อยู่ 2 ตัวคือ ก่อนหน้าและถัดไป ซึ่งถูกใช้ในการเชื่อมลีฟโหนดเข้าด้วยกันเพื่อให้เวลาสแกนมีประสิทธิภาพ ลีฟโหนดยังแสดงให้เห็นถึงกลุ่มที่สร้างมาจากกลุ่มย่อยทั้งหมดที่แสดงถึงเอนทรีตัวเอง แต่ทุกเอนทรีในลีฟโหนดต้องมีการตอบสนองค่าเทรชโฮลด้วยการพิจารณาถึงค่าเทรชโฮล T ด้วย โดยที่ค่ากลางนั้นต้องมีค่าน้อยกว่า T



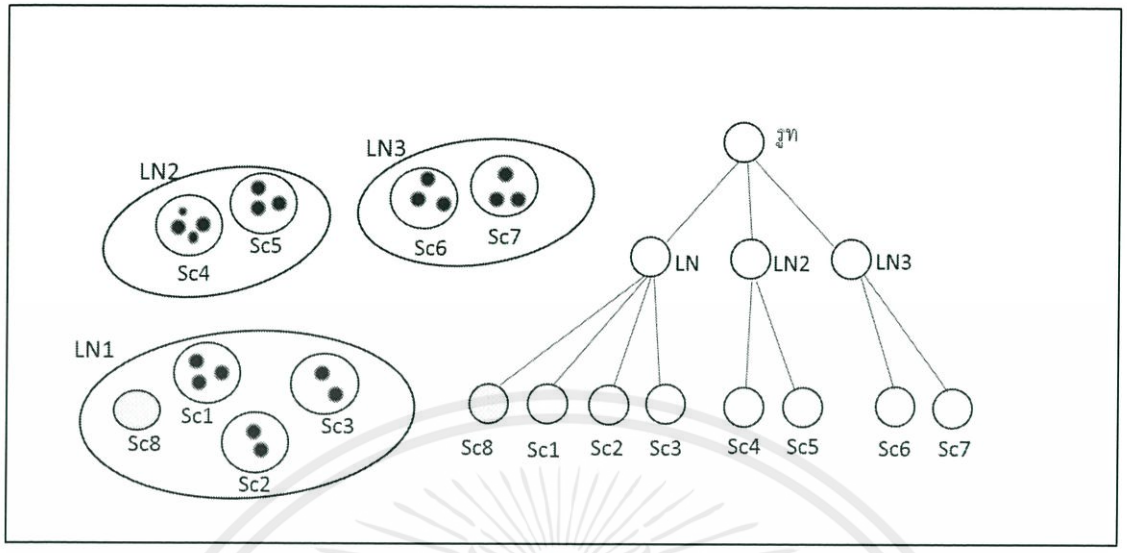
รูปที่ 2.10 ซีเอฟทรีของแต่ละโหนดย่อย

ตัวอย่างของซีเอฟทรี ในช่วงเริ่มต้นของจุดข้อมูลในกลุ่มหนึ่ง ดังรูปที่ 2.11(ก) เมื่อมีข้อมูลใหม่เข้ามาจะทำการตรวจสอบว่าระยะห่างนั้นมีค่ามากกว่า T หรือไม่ ดังรูปที่ 2.11(ข) ถ้าขนาดของกลุ่มนั้นมีขนาดใหญ่เกินกว่าค่า T กลุ่มนั้นจะถูกแบ่งออกเป็น 2 กลุ่ม และจะกระจายจุดออกจากกัน ดังรูปที่ 2.11(ค) ที่แต่ละโหนดของทรี ซีเอฟทรีจะเก็บข้อมูลเกี่ยวกับค่าเฉลี่ยของกลุ่มและเฉลี่ยผลรวมที่ได้ เพื่อให้สามารถคำนวณขนาดของกลุ่มได้อย่างมีประสิทธิภาพ ดังรูปที่ 2.11(ง)

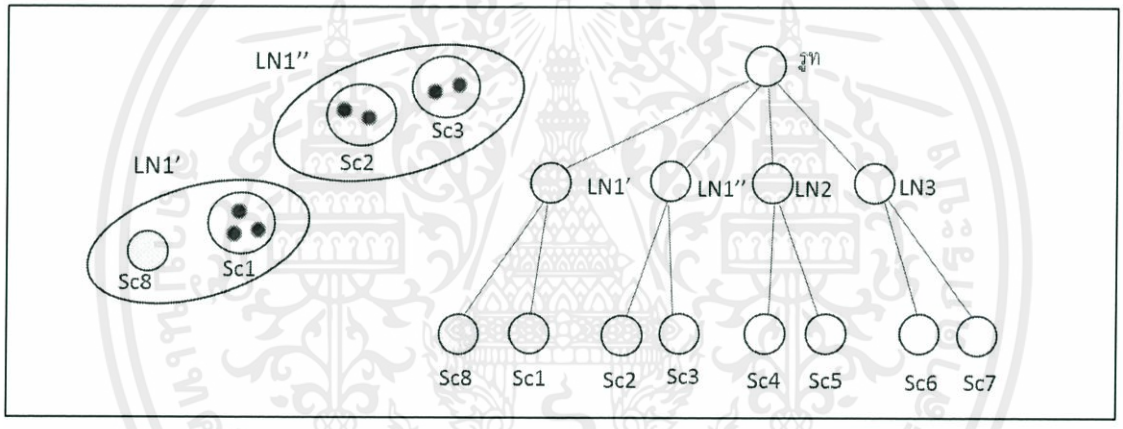


รูปที่ 2.11 ตัวอย่างซีเอฟทรี

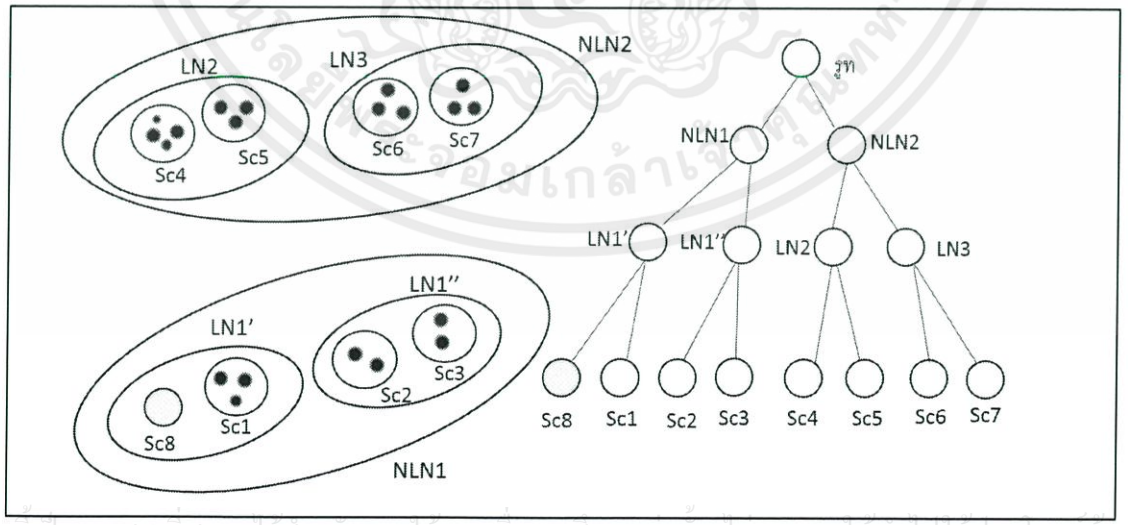
ตัวอย่างอื่นๆ ของซีเอฟทรี แสดงในรูปที่ 2.12 จะเห็นได้ว่ามีข้อมูล sc8 เข้ามาทำให้กลุ่ม LN1 มีมากกว่าสมาชิก 3 จึงต้องทำการแบ่งกลุ่ม LN1 ออกเป็น 2 กลุ่ม ดังรูปที่ 2.12(ค)



(ก)



(ข)



(ค)

รูปที่ 2.12 ตัวอย่างอื่นๆของซีเอฟทรี

จากรูปที่ 2.12(ข) จะเห็นว่านอนลีฟโหนดนั้นมีข้อมูลมากกว่า 3 จึงต้องให้ Root นั้นแบ่งกลุ่มย่อยออกไปเป็นดังรูปที่ 2.11(ค)

สรุปขั้นตอนของอัลกอริทึมเบิร์ชได้ดังนี้

- ขั้นตอนที่ 1 : สแกนข้อมูลทั้งหมดและสร้างจุดเริ่มต้นในหน่วยความจำในซีเอฟทรี
- ขั้นตอนที่ 2 : ทำการสแกนลีฟเอนทรีเพื่อสร้างทรีใหม่ที่ให้มีขนาดเล็กลง
- ขั้นตอนที่ 3 : จัดกลุ่มที่ใหญ่ที่สุด
  - จะใช้ Global หรือ Semi-Global อัลกอริทึมเพื่อแบ่งกลุ่มเอนทรีลีฟโหนดทั้งหมด
  - Adapted agglomerative hierarchical clustering algorithm ถูกนำไปใช้โดยตรงกับกลุ่มย่อยที่แสดงโดยเวคเตอร์ซีเอฟนั้นๆ
- ขั้นตอนที่ 4 : ปรับปรุงผลลัพธ์การแบ่งกลุ่ม (ทางเลือก)
  - เพื่อตรวจสอบข้อมูลและแก้ไขข้อมูลที่ไม่ถูกต้องและปรับกลุ่มต่อไป
  - ใช้จุดศูนย์กลางของกลุ่มในการสร้างที่ขั้นตอนที่ 3 เป็น Seed และ รีดิสทริบิวจุดข้อมูลไปยัง Seed ใกล้เคียงเพื่อเซ็ตกลุ่มข้อมูลใหม่

สรุปอัลกอริทึมเบิร์ช

- อัลกอริทึมเบิร์ชจะดำเนินการได้รวดเร็วกว่าอัลกอริทึมอื่นๆ ที่มีอยู่ และสามารถใช้งานกับชุดข้อมูลที่มีขนาดใหญ่ได้
- อัลกอริทึมเบิร์ชจะสแกนข้อมูลทั้งหมดภายในครั้งเดียว
- สามารถตรวจจับค่าผิดปกติได้ดี
- เมื่อเปรียบเทียบกับอัลกอริทึมอื่นพบว่าอัลกอริทึมเบิร์ชมีความเสถียรและยืดหยุ่นกว่า

2.7 ระยะห่างเชิงเลขคณิต, ระยะห่างเชิงประเภท และระยะห่างเชิงลำดับชั้นสำหรับอัลกอริทึมเบิร์ช

หัวข้อนี้จะกล่าวถึงระยะห่างเชิงเลขคณิต, ระยะห่างเชิงประเภท และระยะห่างเชิงลำดับชั้นสำหรับอัลกอริทึมเบิร์ช (Numerical, Categorical and Hierarchical Distance with BIRCH Algorithm) เนื่องจากปัญหาสำคัญอย่างหนึ่งในการนำข้อมูลของการใช้งานเครือข่ายมาคำนวณทางคณิตศาสตร์คือข้อมูลที่ได้มาจากการเก็บรวบรวมจากอุปกรณ์เครือข่ายนั้นส่วนมากมักจะเป็นข้อมูลที่ไม่สามารถนำมาคำนวณได้โดยตรงยกตัวอย่างข้อมูล เช่น หมายเลขไอพีต้นทาง/ปลายทาง, หมายเลขพอร์ตต้นทาง/ปลายทาง หรือ ชนิดของโพรโทคอล เป็นต้น การนำข้อมูลหรือคุณสมบัติของแพ็คเก็ต

ดังกล่าวมาวิเคราะห์หรือคำนวณทางคณิตศาสตร์จำเป็นต้องมีการแทนค่าข้อมูลนั้นๆและกำหนดวิธีการคำนวณอย่างเหมาะสมแทนการคำนวณแบบปกติ จากการศึกษาพบว่าข้อมูลดังกล่าวนั้นสามารถถูกแทนค่าและกำหนดการดำเนินการทางคณิตศาสตร์ที่เหมาะสมให้ได้

กำหนดให้ R แทนเซตของจำนวนจริง (Real Number), Z แทนเซตของจำนวนเต็ม (Integers), R^d แทนเวคเตอร์ซึ่งมี d มิติบนเซตของ R, C แทนกลุ่มของคุณสมบัติ (Cluster) ซึ่งมีจำนวนสมาชิกในกลุ่มเป็น N เขียนได้เป็น $C = \{X_j : j =,..., N\}$, โดยที่ X เป็นเวคเตอร์ซึ่งมี d มิติแสดงถึงโหนดหรือจุดในกลุ่มคุณสมบัติซึ่งเขียนได้เป็น $X = \{x[i] = 1,..., d\}$

นอกจากนั้นต้องมีการกำหนดการคำนวณค่ากึ่งกลางให้กับชุดคุณสมบัติแต่ละประเภทดังนี้ $\overline{C} = \{\bar{c}[i] : i = 1,..., d\}$ แทนจุดกึ่งกลางของกลุ่มคุณสมบัติ C โดยที่ $\bar{c}[i]$ แทนค่ากึ่งกลางของคุณสมบัติที่เป็นชนิดเดียวกันคือชนิดที่ i

ในการแบ่งกลุ่มของสมบัติที่เกี่ยวข้องกับการจราจรของแพ็คเก็ตในเครือข่ายจะต้องคำนึงถึงประเภทของคุณสมบัติที่กำลังพิจารณาโดยแบ่งประเภทของคุณสมบัติเป็น 3 ประเภทดังนี้ คุณสมบัติเชิงเลขคณิต (Numerical Attribute), คุณสมบัติเชิงประเภท (Categorical Attribute), คุณสมบัติเชิงลำดับชั้น (Hierarchical Attribute)

1) คุณสมบัติเชิงเลขคณิต

คุณสมบัติเชิงเลขคณิต คือ คุณสมบัติที่สามารถนำข้อมูลมาทำการคำนวณได้เลยโดยไม่ต้องผ่านการแทนค่าหรือเปลี่ยนแปลงค่าก่อน ตัวอย่างคุณสมบัติที่ถือเป็นคุณสมบัติเชิงเลขคณิตคือ จำนวนไบต์ข้อมูล คุณสมบัติเชิงเลขคณิตสามารถเขียนได้เป็นปริมาณสเกลาร์ $x[i] \in R$ โดยมี $\bar{c}[i]$ แทนค่ากึ่งกลางของคุณสมบัติที่เป็นคุณสมบัติเชิงเลขคณิตของกลุ่มคุณสมบัติ C ที่มีจำนวนสมาชิกข้อมูล N ชุดคือ

$$\bar{c}[i] = \frac{1}{N}\sum_{j=1}^{N} x_j[i] \qquad (1)$$

การคำนวณระยะห่างเชิงเลขาคณิตคือการคำนวณระยะห่างระหว่างค่ากึ่งกลางข้อมูลของที่เป็นข้อมูลชนิดเลขคณิตเหมือนกันโดยอาศัยหลักการของยูคลิดคือการใช้ระยะห่างเมตริก $\|\bullet\|$:

$$d_n^l(C_1, C_2) = \left\|\bar{c}_1[i] - \bar{c}_2[i]\right\| = \left[\left(\bar{c}_1[i] - \bar{c}_2[i]\right)^2\right]^{\frac{1}{2}} \qquad (2)$$

2) คุณสมบัติเชิงประเภท

คุณสมบัติเชิงประเภทคือคุณสมบัติที่ทำหน้าที่บ่งบอกชนิดหรือประเภทของแพ็คเก็ต ตัวอย่างคุณสมบัติเชิงประเภทเช่น โพรโทคอลและแฟล็กของแพ็คเก็ต ในกรณีของคุณสมบัติชนิดนี้จะถูกแทนค่าด้วย $x[i]$ ซึ่งเป็นเวคเตอร์ที่เป็นสมาชิกของจำนวนเต็ม Z^c โดยที่ c คือจำนวนค่าที่เป็นไปได้ของคุณสมบัตินั้น สำหรับเวคเตอร์ x ซึ่งเป็นเวคเตอร์ของคุณสมบัติที่มีจำนวนค่าที่เป็นไปได้ c ค่า จะสามารถเขียนแทนได้คือ $x[i]=\{a_1,a_2,a_3,...,a_c\}$ โดยที่ $a_c \in \{0,1\}$ ซึ่งจะทำให้ผลรวมของ a_k ในหนึ่งเรคคอร์ดมีค่าเป็น 1 เสมอตามสมการ

$$\sum_{k=1}^{c} a_k = 1 \text{ เมื่อ } a_k \in x[i] \tag{3}$$

ค่ากึ่งกลางข้อมูลของคุณสมบัติเชิงประเภท $\bar{c}[i]$ สามารถหาได้จากสมาการ

$$\bar{c}[i] = \frac{1}{N}\sum_{j=1}^{N} x_j[i] = \left\{\frac{A_1}{N}, \frac{A_2}{N}, ..., \frac{A_c}{N}\right\} \tag{4}$$

เมื่อ

$$A_k = \sum_{j=1}^{N} x_j[i] = \{a_{1,j}, a_{2,j}, ..., a_{c,j}\}$$

ระยะห่างระหว่างกลุ่มข้อมูล C1 กับ C2 สำหรับคุณสมบัตินี้สามารถหาได้จากสมการ

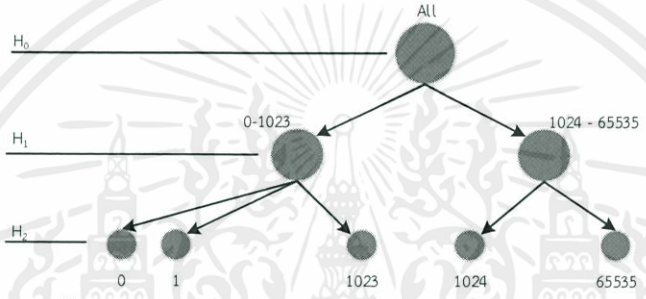
$$d_c^l(C_1,C_2) = \left\|\bar{c_1} - \bar{c_2}\right\| = \frac{1}{N}\left[\sum_{k=1}^{c}(A_{k,1} - A_{k,2})^2\right]^{\frac{1}{2}} \tag{5}$$

3) คุณสมบัติเชิงลำดับชั้น

ตัวอย่างคุณสมบัติของแพ็คเก็ตที่สามารถนำมาวิเคราะห์เป็นคุณสมบัติเชิงลำดับชั้นได้แก่ หมายเลขพอร์ตและหมายเลขไอพี การนำคุณสมบัติเชิงลำดับชั้นมาคำนวณหรือวัดความเหมือนกันนั้นจะพิจารณาคุณสมบัติเชิงลำดับชั้นในรูปแบบของต้นไม้ที่ L ระดับนั่นหมายความว่าต้นไม้จะมีความลึก L-1 โดยที่โดเมนที่อยู่ในระดับ l จะแทนด้วย H_l ดังนั้นโหนดใดๆที่ไม่ได้ถือว่าเป็นลีฟโหนดซึ่งมีตำแหน่งอยู่ในระดับที่ l จะต้องเป็นสมาชิกของ H_l ดังสมการ $h_{l,i} \in H_l$ สำหรับต้นหมายที่มี L ระดับ การพิจารณาถึงค่าๆหนึ่งซึ่งเป็นค่าที่บอกถึง

การที่ปมสองปมใดๆมีบรรพบุรุษเดียวกันจะเรียกค่านั้นว่า Lowest Common Ancestor หรือ LCA ระหว่าง n_1 กับ n_2 แทนด้วย $LCA(n_1,n_2)$ ส่วนความลึกของโหนด n ใดๆจะถูกนิยามว่าเป็นความยาวของเส้นทางตั้งแต่รากจนถึงโหนด n

กำหนดให้ $path(n_1,n_2)$ แสดงถึงเส้นทางที่สั้นที่สุดระหว่างโหนด n_1 ไปยัง โหนด n_2 โหนดราก แสดงด้วย n_r ลีฟโหนดแสดงด้วย l จะได้ว่า $|path(l_1,l_2)| = |path(l_1,l_r)| + |path(l_r,l_2)|$ โดยที่ $|path(n_1,n_2)|$ หมายถึงความยาวเส้นทางระหว่างโหนด n_1 ไปยังโหนด n_2



รูปที่ 2.13 ตัวอย่างต้นไม้ทวิภาคสำหรับหมายเลขพอร์ต

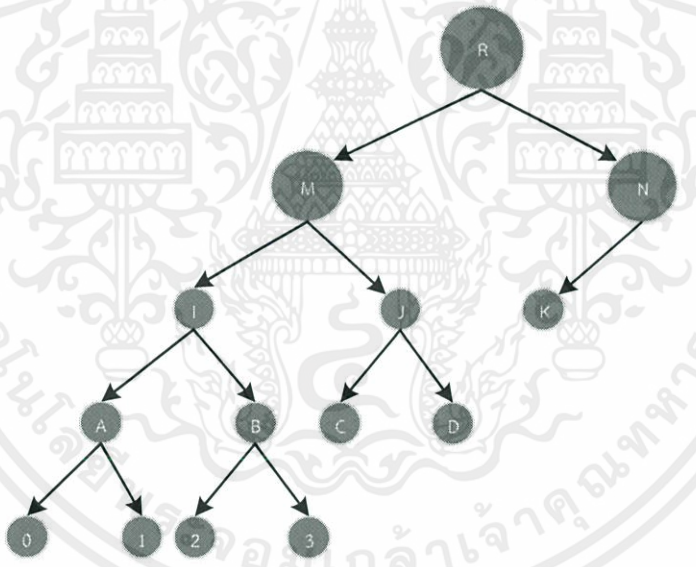
จากรูปแสดงการนำหมายเลขพอร์ตแสดงในรูปแบบของต้นไม้ หมายเลขพอร์ตสามารถแสดงในรูปแบบต้นไม้โดยมีเพียง 2 ระดับ (L=2) อยู่บนโดเมนของเลขจำนวนเต็ม จากรูปจะเห็นได้ว่าในระดับที่ 0 นั้น H_0 แสดงถึงทุกๆหมายเลขพอร์ต (ตั้งแต่ 0 - 2^{16}) $H_0 = \{All\}$ หมายความว่าทุกค่าหมายเลขพอร์ตถือเป็นกลุ่มเดียวกัน ส่วนในระดับที่ 1 นั้นจะมีคำสองคำที่บอกความแตกต่างของโหนดลูกทั้งสองคือ “Low” และ “High” ซึ่งก็คือ $H_1 = \{"Low","High"\}$ นั่นหมายความว่าถ้าหมายเลขพอร์ตเป็น 1 ซึ่งจะถูกจัดอยู่ในกลุ่ม “Low” ส่วนในกรณีของระดับที่ 2 นั้นจะได้ว่า $H_2 = \{0,1,2,...,65535\}$ หมายความว่าจะมี 65536 กลุ่มแต่ละกลุ่มมีสมาชิกก็คือตัวเองนั่นเอง ในกรณีของหมายเลขไอพีก็เช่นเดียวกันคือสามารถเขียนให้อยู่ในรูปแบบต้นไม้ได้

2.7.1 คุณสมบัติที่จำเป็นสำหรับการวัดความคล้ายคลึงระหว่างข้อมูลแบบลำดับชั้น

การนำข้อมูลมาจัดเรียงในรูปแบบต้นไม้ต้องทราบคุณสมบัติที่จำเป็นสำหรับการวัดความคล้ายคลึงระหว่างข้อมูลแบบลำดับชั้น หรือ Desired Properties of a Hierarchical Similarity Measure มิฉะนั้นจะไม่สามารถใช้กระบวนการทางคณิตศาสตร์แบบปกติวัดความคล้ายกันของแต่ละโหนดได้ ดังนั้นต้องมีการกำหนดกระบวนการในการวัดความคล้ายกันของแต่ละโหนดนั้น กำหนดให้ S

คือกระบวนการวัดความคล้ายคลึงระหว่างข้อมูลแบบลำดับชั้น คุณสมบัติที่จำเป็นของกระบวนการดังกล่าวมีดังนี้

1) S ควรจะเป็นกระบวนการที่สมมาตรสำหรับสองโหนด I และ J ใดๆ $S(I,J) = S(J,I)$
2) ความคล้ายคลึงระหว่างโหนด X กับโหนดพ่อ Y จะต้องมีค่ามากกว่าโหนด X กับโหนดข้างเคียงโหนด Y เช่น $S(0,A) > S(0,B)$
3) ค่าของ S จะต้องมีค่าลดลงเมื่อมีการวัดความคล้ายคลึงในแนวดิ่งเริ่มจากโหนด X กับตัวเองขึ้นไปเรื่อยๆจนถึงราก เช่น $S(0,0) > S(0,A) > S(0,I) > S(0,M)$
4) ความคล้ายคลึงของโหนดข้างกันที่มีพ่อเดียวกันในระดับที่ลึกกว่าจะต้องมีค่ามากกว่าความคล้ายคลึงกันของโหนดข้างกันในระดับที่ตื้นกว่า เช่น $S(0,1) \geq S(A,B)$
5) ความคล้ายคลึงของโหนดที่อยู่ในระดับลึกจะต้องมีค่ามากกว่าความคล้ายคลึงของโหนดในระดับที่ตื้นกว่า เช่น $S(0,2) \geq S(0,B)$



รูปที่ 2.14 ตัวอย่างข้อมูลที่จัดเรียงเชิงลำดับชั้น

ในการวัดความคล้ายคลึงระหว่างข้อมูลแบบลำดับชั้นนั้นจะต้องกำหนดการดำเนินการทางคณิตศาสตร์ที่รองรับคุณสมบัติที่กล่าวมาทั้งห้าข้อ จากการศึกษาพบว่า Hierarchical Similarity Measure 1 (HSM_1) ตามเอกสารอ้างอิง[1] มีคุณสมบัติครบถ้วนซึ่งเขียนเป็นสมการได้คือ

$$S_{HSM_1}(n_1,n_2) = \frac{|path(root,n_1) + path(root,n_2)|}{L}$$

2.7.2 การวิเคราะห์หมายเลขไอพีแอดเดรสแบบสมบัติเชิงลำดับชั้น

การวิเคราะห์หมายเลขไอพีสามารถทำได้โดยการจัดการกับหมายเลขไอพีด้วยต้นไม้แบบทวิภาค (Binary Tree) ของพรีฟิกส์ ซึ่งมี 32 ระดับตามรูปแบบของหมายเลขไอพีที่เป็นเลขฐานสองขนาด 32 บิตซึ่งทำให้หมายเลขไอพีมีทั้งสิ้น 2^{32} หมายเลข ค่าพรีฟิกส์ของหมายเลขไอพีจะแสดงถึงระดับในต้นไม้ทวิภาค

คำนิยาม 1 พรีฟิกส์ของหมายเลขไอพี (Prefix of an IP address)

กำหนดหมายเลขไอพีซึ่งเป็นเลขฐานสองขนาด 32 บิตและพรีฟิกส์เป็นเลขฐานสองขนาด p บิตจะได้ว่า พรีฟิกส์คือ p บิตแรกที่มีนัยสำคัญสูงที่สุดของหมายเลขไอพีเขียนแทนด้วย IP/p เมื่อ $0 \le p \le 32$

คำนิยาม 2 CommonPrefix and AggregateIP

CommonPrefix ถูกนิยามว่าเป็นพรีฟิกส์ p ที่ยาวที่สุดที่ทำให้ $IP_1/p = IP_2/p$ ซึ่งเปรียบเทียบได้กับสมการ $|path(root,n_1) \cap path(root,n_2)|$

AggregateIP ของหมายเลขไอพีสองหมายเลข IP_1 และ IP_2 ใดๆถูกนิยามว่าเป็น IP/p เมื่อ p เป็น CommonPrefix ของ IP_1 และ IP_2 ตัวอย่างเช่น

$$AggregateIP(203.192.32.127, 203.192.32.128) = 203.192.32.0/24$$

จากนิยามทำให้สามารถกำหนดการวัดระยะห่างเชิงลำดับชั้นของหมายเลขไอพีได้ดังสมการ

$$D_{HSM_1}(n_1,n_2) = 1 - S_{HSM_1}(n_1,n_2)$$
$$D_{HSM_1}(n_1,n_2) = (L - |path(root,n_1)| \cap |path(root,n_2)|)/L$$

ในกลุ่มของ C สำหรับสมบัติเชิงลำดับชั้นนั้นสามารถถูกพิจารณาค่ากึ่งกลางของข้อมูลได้คือค่า $\overline{IP}/p$ ซึ่งเป็นค่า AggregateIP ของกลุ่มหมายเลขไอพี ในทางปฏิบัติแล้วจะไม่สามารถหาค่าซับเน็ตที่มีค่าน้อยกว่า 8 ได้อยู่แล้วดังนั้นหมายเลขไอพีสองหมายเลขใดๆที่มี CommonPrefix p ที่น้อยกว่าหรือเท่ากับ 8 สามารถพิจารณาว่าไม่มีความคล้ายคลึงกันเลยได้และจะสามารถหาระยะห่างระหว่างข้อมูลสองกลุ่มระหว่าง C_1 และ C_2 ได้ดังนี้

$$d_n^I(C_1,C_2) = \begin{cases} (32-p)/24 & p > 8 \\ 1 & p \le 8 \end{cases}$$

2.7.3 การรวมผลต่างระยะห่างเชิงเลขคณิต, เชิงประเภท และเชิงลำดับชั้น

กำหนดให้ระยะห่างของข้อมูลคุณสมบัติเชิงเลขคณิตคือ d_n , คุณสมบัติเชิงประเภท d_c และคุณสมบัติเชิงลำดับชั้นคือ d_h จะสามารถสรุประยะห่างระหว่างกลุ่มข้อมูล C_1 และ C_2 ได้คือ

$$D_i(C_1,C_2)=\begin{cases} d_n^i(C1,C2), i \in numerical attribute, \\ d_c^i(C1,C2), i \in categorical\ attribute, \\ d_h^i(C1,C2), i \in hierarchical attribute. \end{cases} \quad (6)$$

และจะได้ระยะห่างระหว่างกลุ่มด้วยผลรวมของระยะห่างของทุกคุณสมบัติเป็น

$$D(C_1,C_2)=\left[\sum_{i=1}^{d} D_i(C_1,C_2)\right]^{\frac{1}{2}} \quad (7)$$

2.7.4 การคำนวณรัศมีของกลุ่มข้อมูล

ในการควบคุมความแปรปรวนของข้อมูลในกลุ่มข้อมูลนั้นเราต้องมีการคำนวณหรือกำหนดขอบเขตบางอย่างของข้อมูลไม่ให้ข้อมูลภายในกลุ่มมีค่ามากเกินไป ในกรณีของสมบัติเชิงเลขคณิตและเชิงประเภทนั้นเราสามารถใช้การคำนวณทางคณิตศาสตร์ซึ่งจะได้กล่าวต่อไปในการกำหนดขอบเขตของข้อมูลได้ แต่สำหรับสมบัติเชิงลำดับชั้น เช่น หมายเลขไอพีหรือหมายเลขพอร์ตจะต้องใช้กระบวนการที่แตกต่างออกไป ผลรวมทางคณิตศาสตร์ของรัศมีของสมบัติแต่ละประเภทนั่นก็คือรัศมีของกลุ่มข้อมูลนั้นเอง ขั้นตอนการคำนวณรัศมีของกลุ่มข้อมูลมีดังต่อไปนี้

1) รัศมี R_n สำหรับสมบัติเชิงเลขคณิตที่ i ในกลุ่มข้อมูลที่มี N เรคคอร์ดและค่ากึ่งกลางข้อมูล $\bar{c}[i]$ คือ

$$R_n=\sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_j[i])^2-(\bar{c}[i])^2}$$

2) รัศมี R_c สำหรับคุณสมบัติเชิงประเภทที่ i ซึ่งมีค่าที่สามารถเป็นไปได้ c ค่าคือ

$$R_n=\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\sum_{k=1}^{c}(a_{k,j})^2\right)-\frac{1}{c^2}\left(\sum_{k=1}^{c}A_k\right)^2}$$

3) สำหรับสมบัติเชิงลำดับชั้นนั้นไม่สามารถใช้การคำนวณเช่นเดียวกับข้อ ก. และ ข. เนื่องจากสมบัติประเภทนี้มีการจัดเรียงในรูปแบบเป็นลำดับชั้น การใช้ค่ามากสุดและน้อยที่สุดสำหรับพิจารณารัศมีจึงเป็นสิ่งที่เหมาะสม ในการพิจารณารัศมมีของกลุ่มข้อมูลประเภทนี้จะใช้สมการ

$$R_h = (32 - Common\,\mathrm{Pr}efix(\min IP, MaxIP))/32 \qquad (9)$$

ตัวอย่างเช่น

$C_1.range = (192.168.0.1, 192.168.0.2)$ Radius = (32-31)/32 = 1/32

$C_2.range = (192.168.0.1, 192.168.0.255)$ Radius = (32-24)/32 = ¼

2.8 เน็ตโฟลว์

เน็ตโฟลว์ (NetFlow) เป็นโพรโทคอลเครือข่ายที่ถูกพัฒนาโดยระบบของซิสโก้ (Cisco) ใช้ในการเก็บรวบรวมข้อมูลจากการทราฟฟิคไอพี เน็ตโฟลว์นั้นถูกใช้โดยผู้เชี่ยวชาญด้านไอทีในการวิเคราะห์การไหลของทราฟฟิคและกำหนดปริมาณทราฟฟิคที่กำลังมาว่าจะไปที่ไหนและมีจำนวนเท่าไหร่ที่ถูกสร้างขึ้น

เน็ตโฟลว์จะเปิดใช้งานเราเตอร์ในการนำข้อมูลทราฟฟิคและสถิติออกมาบันทึกไว้ซึ่งจะถูกเก็บรวบรวมโดยตัวเก็บรวบรวมเน็ตโฟลว์หรือเน็ตโฟลว์คอลเล็คเตอร์ (NetFlow Collector) ซึ่งตัวเก็บจะมีการวิเคราะห์ทราฟฟิคจริงและนำเสนอให้กับผู้ใช้ เน็ตโฟลว์ได้กลายเป็นมาตรฐานอุตสาหกรรมสำหรับตรวจสอบทราฟฟิคและได้รับการสนับสนุนจากหลายแพลตฟอร์ม การไหลของข้อมูลในเครือข่ายนั้นสามารถกำหนดได้หลายทาง ในซิสโก้มาตรฐานเน็ตโฟลว์เวอร์ชันที่ 5 มีการกำหนดการไหลลำดับของแพ็คเกจหลายค่า เช่น

- ไอพีแอดเดรสต้นทาง, ไอพีแอดเดรสปลายทาง
- ไอพีแอดเดรสของ next hop เราท์เตอร์
- ขนาดของแพ็คเกจ
- ทีซีพี / ยูดีพี ของพอร์ตต้นทาง และพอร์ตปลายทาง
- ประเภทการให้บริการของไอพี
- ชนิดของโพรโทคอลไอพี

2.9 เครื่องมือที่ใช้ในการดำเนินงาน

เครื่องมือที่ใช้ในการดำเนินงานมีดังนี้

1) เน็ตโฟลว์ อนาไลเซอร์ (NetFlow Analyzer)

เน็ตโฟลว์ อนาไลเซอร์ เป็นอุปกรณ์ที่ใช้ในการวิเคราะห์ทราฟฟิค และรายงานเกี่ยวกับที่เครือข่ายที่มีการใช้งาน เพื่อแสดงผลตามเวลาจริงในระบบเครือข่ายอย่างมีประสิทธิภาพ คุณสมบัติที่สำคัญ เช่น

- ตรวจสอบแอพพลิเคชั่นและโพรโทคอล ดูการใช้งานบนเครือข่ายที่มีการใช้งานบ่อยๆ
- รายงานทราฟฟิคตามเวลาจริง รับการแสดงผลเวลาจริงของทราฟฟิคในเครือข่าย
- การเตือนสามารถตั้งค่าการเตือนตามค่าเทรชโฮลสำหรับทราฟฟิคบนเครือข่าย

2) เอฟโพรบ (FProbe)

เป็นเครื่องมือลิบแคปที่ใช้ในการเก็บรวบรวมข้อมูลทราฟฟิคของเครือข่ายและปล่อยทราฟฟิคนั้นออกมาในรูปการไหลแบบเน็ตโฟลว์ไปยังที่เก็บข้อมูลที่ระบุไว้ เอฟโพรบรองรับการทำงานของ Netflow เวอร์ชั่นที่ 5 ซึ่งเป็นเวอร์ชั่นมาตรฐานสำหรับอุปกรณ์เครือข่ายทุกตัวที่สามารถทำงานกับ Netflow ได้ เอฟโพรบทำงานบนระบบปฏิบัติการลินุกซ์ ซึ่งจะทำการรวบรวมข้อมูลจากการใช้งานเครือข่ายที่ผ่านเครื่องคอมพิวเตอร์ที่ได้ติดตั้งเอฟโพรบไว้และจะทำการจัดทำข้อมูลสรุปเป็นแพ็คเก็ตส่งไปยังเครื่องปลายทางที่ได้ตั้งค่าไว้

บทที่ 3
การออกแบบและการพัฒนา

ในบทนี้จะกล่าวถึงการนำความรู้จากการค้นคว้ามาออกแบบระบบตรวจหาบอทเน็ตจากการวิเคราะห์พฤติกรรมของระบบเครือข่าย โดยระบบจะทำหน้าที่ในการบันทึกการรับส่งข้อมูลต่างๆบนเครือข่ายทีซีพี/ไอพีลงในฐานข้อมูล จากนั้นจะนำข้อมูลที่เก็บไว้มาแบ่งกลุ่มตามลักษณะของการรับส่งข้อมูลเพื่อเปรียบเทียบกับพฤติกรรมของบอทเน็ตในภายหลัง

3.1 รายละเอียดของระบบที่พัฒนา

การออกแบบและการพัฒนาระบบ จะต้องพิจารณาถึงรายละเอียดของระบบ ขั้นตอนการทำงานต่างๆ ของโปรแกรม รวมไปถึงการแสดงผลลัพธ์ของการวิเคราะห์ของโปรแกรม

3.1.1 รายละเอียดการนำเข้าข้อมูล (Input Specification)

การรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ในเครือข่ายกับอินเทอร์เน็ต เป็นข้อมูลนำเข้าพื้นฐานของระบบซึ่งจะอาศัยความสามารถของ เน็ตโฟลว์ Generator สร้างแพ็คเก็ตเน็ตโฟลว์ที่สรุปการส่งข้อมูลระหว่างแต่ละเครื่องคอมพิวเตอร์เพื่อนำข้อมูลที่ได้ไปทำการจัดกลุ่มข้อมูลแล้วเปรียบเทียบความใกล้เคียงกันกับข้อมูลของบอทเน็ต

3.1.2 รายละเอียดผลลัพธ์ของระบบ (Output Specification)

ผลลัพธ์ของระบบสามารถบ่งบอกได้ว่าการใช้งานของเครื่องคอมพิวเตอร์ใดในช่วงเวลาใด โดยผลลัพธ์ที่ได้อาจจะเป็นหรือไม่เป็นบอทเน็ตได้ เพราะการวิเคราะห์การใช้งานเครือข่ายอาศัยการแบ่งกลุ่มพฤติกรรมของการใช้งานเครือข่ายซึ่งต่างจากการตรวจหาบอทเน็ตจากลักษณะเฉพาะ (Signature) ซึ่งจะสามารถบ่งบอกได้เลยว่าเป็นบอทเน็ตหรือไม่ แต่ข้อดีของการวิเคราะห์พฤติกรรมคือสามารถประเมินได้ว่าเครื่องใดอาจเป็นบอทเน็ตโดยไม่ต้องอาศัยการอัพเดทฐานข้อมูลลักษณะเฉพาะของบอทเน็ต

3.1.3 ขอบเขตของระบบที่พัฒนา

1) ระบบตรวจจับบอทเน็ตนี้พัฒนาขึ้นเพื่อเป็นต้นแบบในการตรวจจับบอทเน็ตเชิงพฤติกรรมซึ่งจะต้องอาศัยการคำนวณและแบ่งกลุ่มของพฤติกรรมการทำงานของข้อมูลปริมาณมากซึ่งจำเป็นต้องใช้ความสามารถของเครื่องเซิร์ฟเวอร์ที่สูง
2) ระบบนี้ได้ถูกพัฒนามาโดยใช้การวิเคราะห์ข้อมูลจากการสังเกตุทำให้ไม่สามารถตรวจจับบอทเน็ตแบบทันที (Real Time) ได้
3) ระบบถูกพัฒนาขึ้นโดยใช้ภาษาจาวาซึ่งมีความเร็วน้อยกว่า Native Language เช่น C หรือ C++

3.1.4 ข้อจำกัดของระบบที่พัฒนา

1) ระบบนี้เป็นระบบที่สามารถตรวจจับบอทเน็ตได้จากพฤติกรรมที่ใกล้เคียงตัวอย่างบอทเน็ตที่ได้บันทึกไว้ซึ่งจะไม่สามารถตรวจจับบอทเน็ตที่มีพฤติกรรมที่แตกต่างจากที่ระบุไว้มากๆได้
2) ระบบนี้ไม่สามารถตรวจจับบอทเน็ตแบบ Real Time ได้
3) ไม่สามารถตรวจจับบอทเน็ตที่ไม่แสดงพฤติกรรมออกมาได้

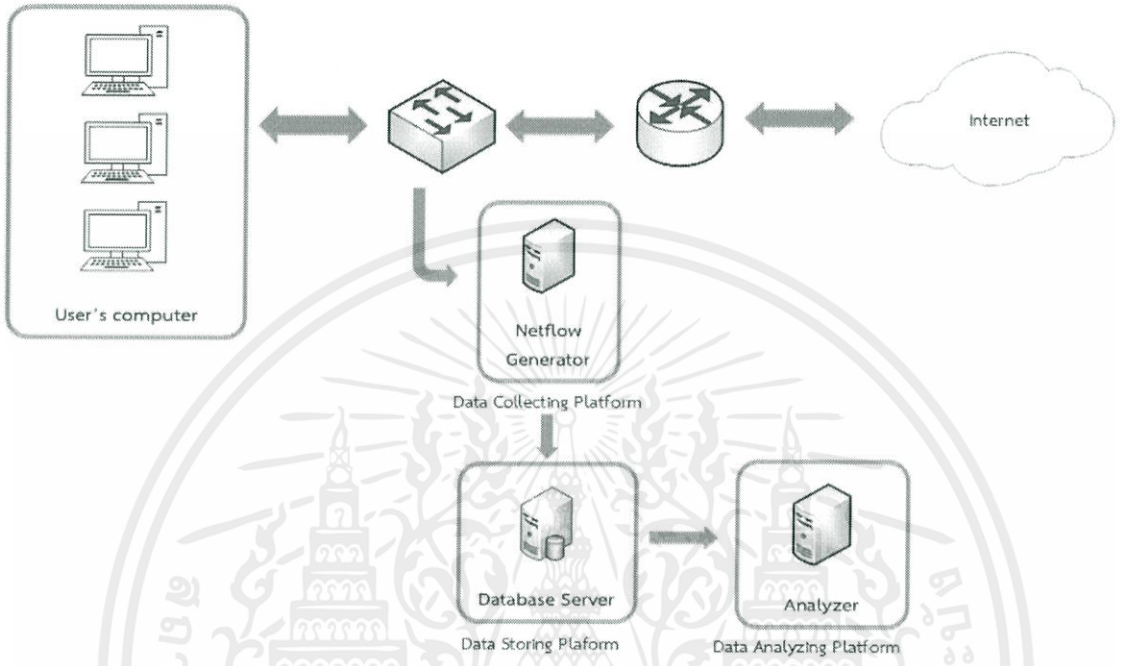
3.1.5 เครื่องมือที่ใช้ในการพัฒนา

1) สภาพแวดล้อมในการพัฒนา
    - หน่วยประมวลผล Intel® Core™ i5-2410M 2.30 GHz
    - หน่วยความจำหลัก 8 GB
    - หน่วยความจำรอง 640 GB
    - ระบบปฏิบัติการ Windows
2) ซอฟต์แวร์ที่ใช้พัฒนา
    - Eclipse Standard 4.3
    - mysql-connector-java-5.1.26-bin.jar
    - Fprobe
3) ภาษาที่ใช้ในการพัฒนา คือ ภาษา Java

3.2 รูปแบบโครงสร้างของระบบ

โครงสร้างของระบบประกอบไปด้วยโครงสร้างหลักสามส่วนคือส่วนรวบรวมข้อมูลทำหน้าที่ในการเก็บรวบรวมข้อมูลการรับ-ส่งข้อมูลของระบบเครือข่าย ส่วนเก็บรักษาข้อมูลทำหน้าที่ในการ

เก็บรักษาข้อมูลที่ได้จากการรวบรวมข้อมูลไว้ในฐานข้อมูล และส่วนวิเคราะห์ข้อมูลและแสดงผลทำหน้าที่นำข้อมูลจากฐานข้อมูลมาวิเคราะห์เปรียบเทียบกับข้อมูลของบอทเน็ตตัวอย่าง

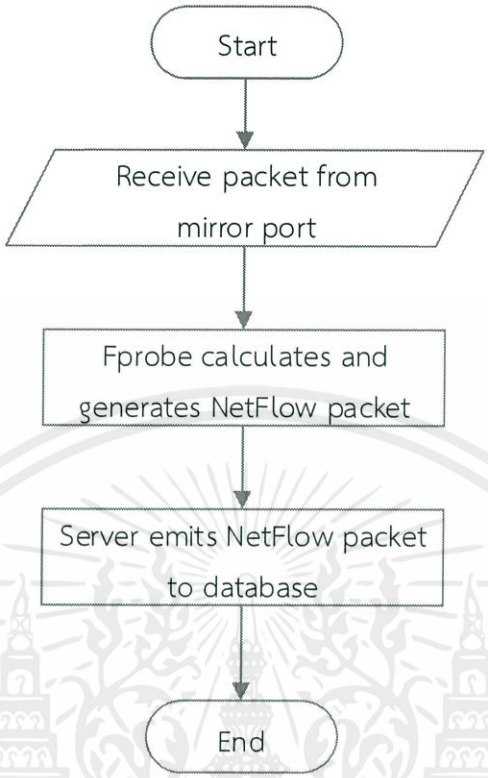


รูปที่ 3.1 ภาพรวมโครงสร้างของระบบ

3.2.1 ส่วนรวบรวมข้อมูล

ส่วนรวบรวมข้อมูล (Data Collecting Platform) สามารถใช้ฟังก์ชั่นพิเศษที่มีในสวิตช์หรือเราท์เตอร์ของซิสโกหรือของผู้ผลิตรายอื่นๆที่รองรับการใช้งานเน็ตโฟลว์ได้ แต่เนื่องจากอุปกรณ์เครือข่ายของสาขาวิชาไม่รองรับการใช้งานฟังก์ชั่นดังกล่าวจึงจำเป็นต้องหาวิธีอื่นในการสร้างแพ็คเก็ตเน็ตโฟลว์ขึ้นมา

ส่วนรวบรวมข้อมูลประกอบด้วยเครื่องเซิร์ฟเวอร์ที่ได้ทำการลงซอฟต์แวร์เอฟโพรปซึ่งเป็นซอฟต์แวร์ที่สามารถสร้างแพ็คเก็ตเน็ตโฟลว์ได้จากการวิเคราะห์แพ็คเก็ตต่างๆที่วิ่งผ่านเครื่องเซิร์ฟเวอร์ ตัวเซิร์ฟเวอร์ทำหน้าที่รวบรวมข้อมูลจากอุปกรณ์เครือข่ายสวิตช์หรือเราท์เตอร์ผ่านมิลเลอร์พอร์ตเพื่อนำข้อมูลที่ได้ไปสรุปและสร้างเป็นแพ็คเก็ตเน็ตโฟลว์และส่งต่อไปยังส่วนเก็บรักษาข้อมูล



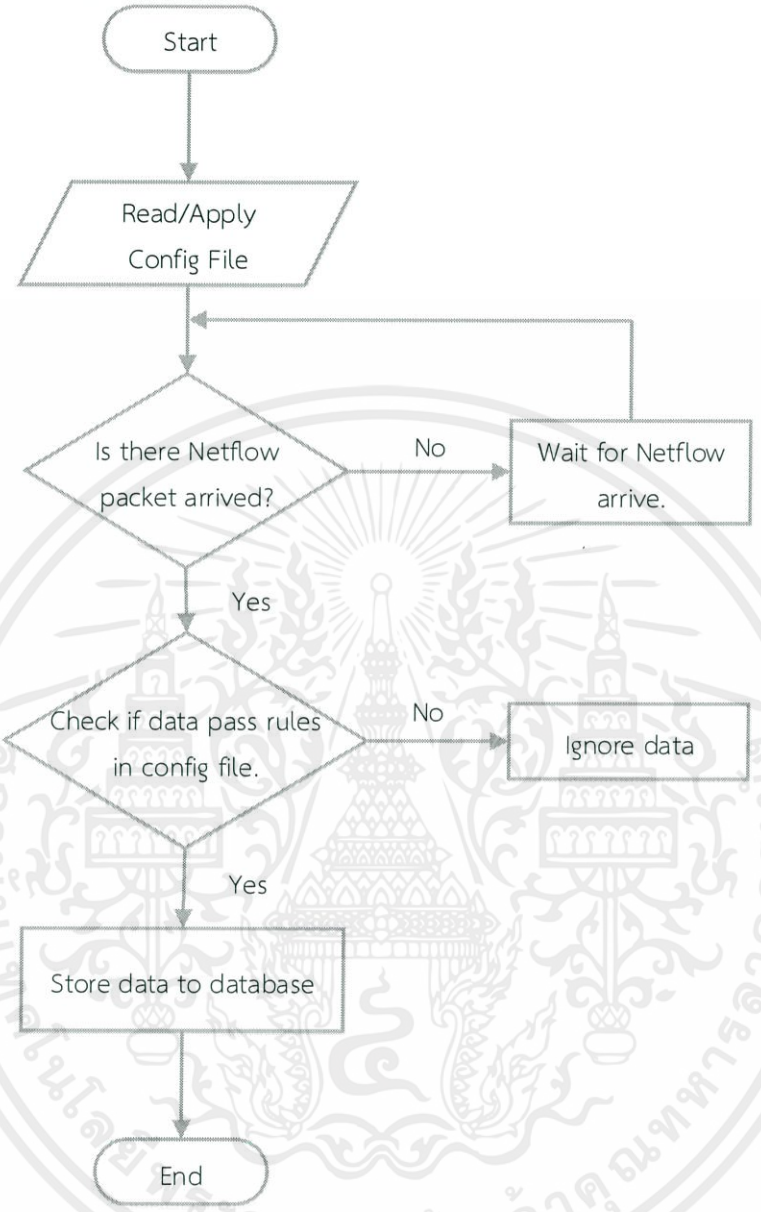
รูปที่ 3.2 ผังแสดงการทำงานของส่วนรวบรวมข้อมูล

3.2.2 ส่วนเก็บรักษาข้อมูล

หลังจากที่ส่วนรวบรวมข้อมูลได้ทำการรวบรวมข้อมูลการใช้งานเครือข่ายแล้วก็จะสร้างแพ็คเก็ตเน็ตโฟลว์ส่งมายังส่วนเก็บรักษาข้อมูล

ส่วนเก็บรักษาข้อมูล (Data Storing Platform) คือส่วนที่ทำหน้าที่ในการรักษาข้อมูลสองประเภท คือ ข้อมูลดิบที่ได้จากส่วนรวบรวมข้อมูลและข้อมูลที่ผ่านการคำนวณแล้ว ส่วนเก็บรักษาข้อมูลจะใช้ซอฟต์แวร์ Mysql ซึ่งเป็น รีเลชั่นเนล ดาตาเบส (Relational Database) ชนิดหนึ่งที่ได้รับความนิยมและมีความรวดเร็วในการเก็บและดึงข้อมูลมาใช้งานทำหน้าที่เก็บรักษาข้อมูลซึ่งเหตุผลที่เลือกใช้ฐานข้อมูลนี้ได้กล่าวไว้แล้วในบทที่สอง การที่เลือกใช้ รีเลชั่นเนล ดาตาเบส สาเหตุที่สำคัญอย่างหนึ่งคือการที่ต้องเก็บข้อมูลขนาดเล็กเป็นปริมาณมากซึ่งจำเป็นต้องใช้ความสามารถของ ดีบีเอ็มเอส ที่รวดเร็วในการดึงข้อมูลที่ต้องการมาใช้ในการคำนวณ

ในการเก็บรักษาข้อมูลที่ได้จากส่วนรวบรวมข้อมูล แทนที่จะเก็บข้อมูลทั้งหมดที่ได้จากส่วนรวบรวมข้อมูลระบบจะมีการกรองแพ็คเก็ต (Data Reduction Filters) ที่ไม่จำเป็นในการนำไปวิเคราะห์ออกไปก่อนเช่น แพ็คเก็ตที่มีการระบุที่อยู่ปลายทางเป็นที่อยู่ภายในเน็ตเวิร์คเดียวกันหรือเป็นที่อยู่ที่รู้จักดีอยู่แล้ว เช่น กูเกิ้ล, ยาฮู, ยูทูป และอื่นๆ

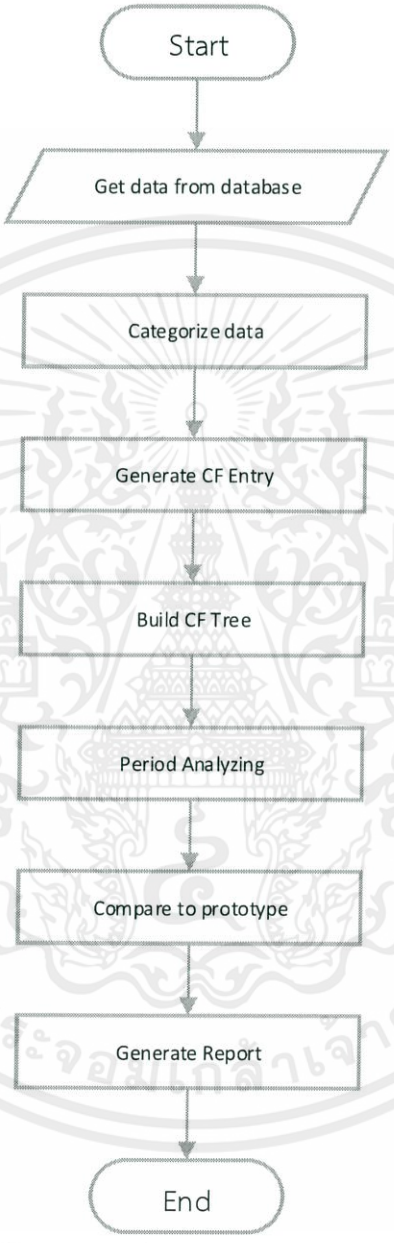


รูปที่ 3.3 ผังแสดงการทำงานของส่วนเก็บรักษาข้อมูล

3.2.3 ส่วนวิเคราะห์ข้อมูล

ส่วนวิเคราะห์ข้อมูล (Data Analyzing Plaform) จะทำงานภายหลังจากที่ได้ข้อมูลมาเก็บลงในฐานข้อมูลแล้วก็จะสามารถนำข้อมูลมาวิเคราะห์ได้ ซึ่งข้อมูลที่นำมาวิเคราะห์นั้นเป็นข้อมูลดิบที่ได้เก็บข้อมูลจากส่วนรวบรวมข้อมูล ปัญหาอย่างหนึ่งของการนำข้อมูลเกี่ยวกับการใช้งานเครือข่ายมาวิเคราะห์หรือคำนวณในทางคณิตศาสตร์คือข้อมูลที่ได้มานั้นส่วนใหญ่ไม่ได้เป็นข้อมูลเชิงคำนวณได้ (Numerical) แต่ถึงอย่างไรก็ตามก็ยังสามารถนำข้อมูลดังกล่าวมาคำนวณได้โดยผ่านกระบวนการ

แทนค่าข้อมูลด้วยชุดข้อมูลที่เหมาะสมและต้องอาศัยกระบวนการในการคำนวณที่แตกต่างไปจากการคำนวณแบบปกติ กระบวนการดังกล่าวได้ถูกกล่าวถึงแล้วในบทที่ 2



รูปที่ 3.4 ผังแสดงการทำงานของส่วนวิเคราะห์ข้อมูล

3.3 แนวคิดการวิเคราะห์ข้อมูล

จากกระบวนการทางคณิตศาสตร์ในบทที่ 2 ในหัวข้อนี้จะเป็นการนำเอาวิธีการต่างๆ มาใช้ในการวิเคราะห์การจราจรในเครือข่าย โดยใช้พื้นฐานของการแบ่งกลุ่มข้อมูลเชิงลำดับชั้น เพื่อสรุป

พฤติกรรมของแพ็คเก็ตส่วนใหญ่ที่สามารถสังเกตุได้จากการใช้งานเครือข่าย แนวคิดของโครงงานจะนำเอาข้อมูลของแต่ละแพ็คเก็ตมาวิเคราะห์และสร้างเป็นต้นไม้เชิงลำดับชั้นตามอัลกอริทึมเบิร์ชโดยแต่ละเรคอร์ดของข้อมูลที่ได้จากเครือข่ายจะถูกเพิ่มเข้าไปในกลุ่มข้อมูลที่มีความใกล้เคียงกับเรคคอร์ดที่พิจารณาที่สุดโดยใช้การวัดผลต่างเชิงเลขคณิต, เชิงประเภทและเชิงลำดับชั้นสำหรับทุกๆคุณสมบัติ โดยที่แต่ละเรคคอร์ดจะถูกรวมเข้าไปในกลุ่มข้อมูลหรือไม่นั้นขึ้นอยู่กับสองปัจจัยคือ รัศมีของกลุ่มข้อมูล (R) ซึ่งจะอธิบายวิธีการนวณในภายหลังและจำนวนข้อมูลมากที่สุดที่สามารถอยู่ในกลุ่มข้อมูลนั้นๆ ได้

3.3.1 รูปแบบการแบ่งกลุ่มข้อมูล

โครงงานนี้จะทำการแบ่งกลุ่มข้อมูลตามแนวทางของอัลกอริทึมเบิร์ชซึ่งจะทำให้ได้ผลลัพธ์เป็นต้นไม้เชิงลำดับชั้นหรือซีเอฟทรีลีฟโหนดของซีเอฟทรีจะแสดงถึงรายละเอียดของข้อมูลที่แบ่งย่อยที่สุดในแต่ละกลุ่มข้อมูลและรากของ ซีเอฟทรี จะแสดงถึงคุณลักษณะทั่วไปของกลุ่มข้อมูลนั้นๆ กลุ่มข้อมูลใดๆ C_l จะถูกแสดงด้วยเวคเตอร์ ซีเอฟ ซึ่งมีข้อมูลสรุปที่เพียงพอที่จะใช้คำนวณจุดกึ่งกลางของข้อมูล $\bar{c}_l$ และรัศมี R_l ของกลุ่มข้อมูล ในทำนองเดียวกับกรณีของโหนดพ่อก็สามาถใช้ข้อมูลของโหนดลูกๆในการคำนวณจุดกึ่งกลางข้อมูลและรัศมีของข้อมูลในกลุ่มข้อมูลเดียวกันได้ ขั้นตอนในการสร้างซีเอฟทรีคือเมื่อมีข้อมูลเรคคอร์ดใหม่ X จากเครือข่าย เรคคอร์ด X จะถูกดำเนินการเพิ่มเข้าไปในกลุ่มข้อมูล C_l ที่ใกล้เคียงกับเรคคอร์ด X ที่สุดโดยใช้การวัดระยะห่าง $D(X, C_l)$ ในสมการที่ (7) ถ้าการเพิ่ม X เข้าไปในกลุ่มข้อมูล C_l แล้วทำให้รัศมีของกลุ่มข้อมูลที่คำนวณใหม่มีค่าเกินกว่าค่าเทรชโฮล์ T ซึ่งมีค่าอยู่ในช่วง [0,1] จะต้องมีการสร้างกลุ่มข้อมูลใหม่ซึ่งอยู่บนพื้นฐานของข้อมูลเรคคอร์ด X แต่ถ้ารัศมีที่คำนวณได้ใหม่มีค่าอยู่ไม่เกินเทรชโฮล์ T ก็จะต้องมีการตรวจสอบว่าจำนวนข้อมูลในกลุ่มข้อมูลมีค่ามากที่สุด m หรือไม่ ถ้าไม่ก็สามารถเพิ่มเรคคอร์ด X เข้าไปในกลุ่มข้อมูลได้ ในกรณีที่ต้องมีการสร้างกลุ่มข้อมูลใหม่สำหรับเรคคอร์ด X แต่โหนดที่รองรับกลุ่มข้อมูลนั้นมีจำนวนกลุ่มข้อมูลเต็มที่แล้วก้ต้องมีการสร้างโหนดใหม่ขึ้นมาเพื่อรองรับเรคคอร์ด X แต่อย่างไรก็ตามเรคคอร์ค X จะไม่ถูกจัดให้อยู่ในกลุ่มข้อมูลใหม่ในโนหดใหม่ทันทีแต่ต้องมีการจัดเรียงกลุ่มข้อมูลใหม่โดยแบ่งแยกกลุ่มข้อมูลที่มีระยะห่างข้อมูลมากที่สุดออกมาอยู่คนละกลุ่มก่อนจากนั้นจะมีการเพิ่มเวกเตอร์ซีเอฟเข้าไปยังโหนดพ่อสำหรับโหนดใหม่ที่ถูกสร้างขึ้นมา

```
1.      find the nearest cluster C_l to record X in N
                                    2.      if N is a leaf node then
3.              C_l' ← insert X into C_l
4.              Calculate radius R_l of C_l'
5.              if R_l ≤ T, where T is a threshold in [0,1] then
6.                      update C_l ←C_l'
7.              else {R > T}
8.                      Create a new cluster C_new based on X
9.              endif
10.     else {N is a non-leaf node}
11.             C_new ←insert(X, C_l.nextnode)
                {C_l.nexttnode is the child node of CF entry C_l}
12.             Update statistics in C_l
13.     endif
14.     if C_new
        {a new CF entry needs to be inserted}
15.             if node N has space for C_new
16.                     add C_new to N
17.                     return nil
18.             else
                {need to split N}
19.                     create new node N_new, a and add C_new
20                      seed N and N_new with the two most
                                Distant clusters C_i and C_j in N
21.                     distribute C_new and the rest of C_k, k≠ i≠ j
                        According to their proximity to the seeds in N and N_new
22.                     return CF entry for N_new
23.             endif
24.     else
        {no new CF entry was created}
25.             return nil
26.     endif
```

รูปที่ 3.5 โค้ดเทียมสำหรับอัลกอริทึมเพิ่มเรคคอร์ด X ในซีเอฟทรี

3.3.2 การสรุปผล

เนื่องจากการแบ่งกลุ่มข้อมูลออกเป็นกลุ่มๆแบบลำดับชั้นแบบซีเอฟทรีนั่นหมายความว่าโหนดที่เป็นโหนดพ่อจะเป็นโหนดที่รวบรวมข้อมูลสรุปของโหนดลูกๆเอาไว้ อย่างไรก็ตามในการ

สรุปผลเราจำเป็นจะต้องเลือกกลุ่มข้อมูลที่น่าจะมีนัยสำคัญมากที่สุดในการแสดงผลและสรุปเป็นรายงานของระบบ

การเลือกกลุ่มข้อมูลที่มีนัยสำคัญนั้นขึ้นอยู่กับข้อมูลและจำนวนข้อมูลในแต่ละกลุ่มว่าสามารถบ่งบอกอะไรกับเราได้อย่างเช่นการกระจายของข้อมูลภายในกลุ่มหรือขนาดของกลุ่มข้อมูลที่ใหญ่จะทำให้เราสามารถสังเกตุได้ว่านั่นอาจจะเป็นนัยสำคัญอย่างหนึ่งเราจึงเลือกการสรุปผลของการแบ่งกลุ่มจากขนาดของกลุ่มข้อมูลที่ใหญ่จนผิดปกติซึ่งอาจบ่งบอกถึงการโจมตีในรูปแบบต่างๆได้ แต่อย่างไรก็ต้องมีเกณฑ์ซึ่งจะบอกได้ว่ากลุ่มข้อมูลใดใหญ่เกินไปเกณฑ์ที่ใช้คือค่า Tr ซึ่งเป็นเทรชโฮล์ของขนาดกลุ่มข้อมูลหาได้จาก

$$T_r = r\tau \text{ ซึ่ง } r = [0,1] \text{ และ } \tau \text{ แทนจำนวนเรคคอร์ดทั้งหมด}$$

ค่า r ซึ่งเป็นเทรชโฮล์ของการแบ่งกลุ่มนั้นจะถูกกำหนดให้เป็น 0 ในช่วงเริ่งต้นซึ่งจะทำให้ทุกชุดข้อมูลที่เข้าผ่านกระบวนการวิเคราะห์ถูกจัดให้อยู่ในคนละกลุ่มซึ่งจะทำให้ต้องมีการแบ่งกลุ่มทุกๆครั้งที่มีเรคคอร์ดใหม่เขข้ามา จนกระทั้งจำนวนหน่วยความจำที่ใช้จะเกินค่า M ซึ่งเป็นค่าหน่วยความจำที่มากที่สุดที่ระบบทำงานได้ ค่า r จะถูกทำให้เพิ่มขึ้นซึ่งมีผลทำให้จำนวนกลุ่มข้อมูลลดลงและใช้หน่วยความจำลดลงด้วย

เมื่อได้กลุ่มข้อมูลที่มีขนาดข้อมูลมากพอที่จะนำมาวิเคราะห์นัยสำคัญ ระบบจะนำกลุ่มข้อมูลที่ได้มานั้นมาทำการหาคาบการสื่อสารของแต่ละการเชื่อมต่อ ขึ้นอยู่กับว่าผู้ใช้ตั้งค่าระบบให้มีค่ากำหนดต่างๆเป็นเท่าใดอย่างเช่น มีการเชื่อมต่อทุกๆ 60 นาทีเป็นจำนวนมากว่า 10 ครั้ง ข้อมูลเหล่านี้จะถูกนำไปพิจารณาเปรียบเทียบกับบอทที่มีข้อมูลอยู่อีกครั้งหนึ่ง หากการเปรียบเทียบนั้นมีระยะห่างของข้อมูลอยู่ในเกณฑ์ที่เหมาะสมข้อมูลของการเชื่อต่อนั้นจะถูกกำหนดว่าเป็นข้อมูลที่ต้องสงสัย

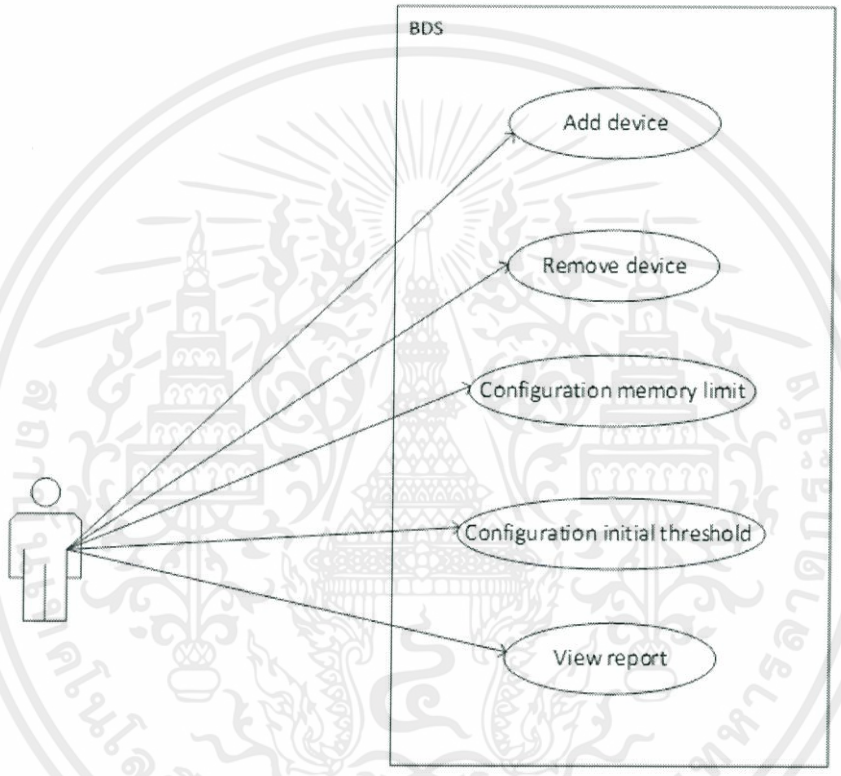
การกำหนดการวิเคราะห์ข้อมูลขึ้นอยู่กับผู้ใช้ว่าต้องการให้ระบบทำการวิเคราะห์ข้อมูลเป็นช่วงเวลาเท่าใดโดยสามารถกำหนดได้เช่น เป็นช่วงวัน สัปดาห์ หรือแม้แต่ช่วงเดือน ซึ่งระบบจะสร้างรายงานเป็นกราฟต้นไม้และบทสรุปของรายงานนั้นเป็นไฟล์ TXT โดยที่สามารถเป็นอ่านได้จากระบบ

3.4 ส่วนของผู้ใช้งานและยูสเคสไดอะแกรม

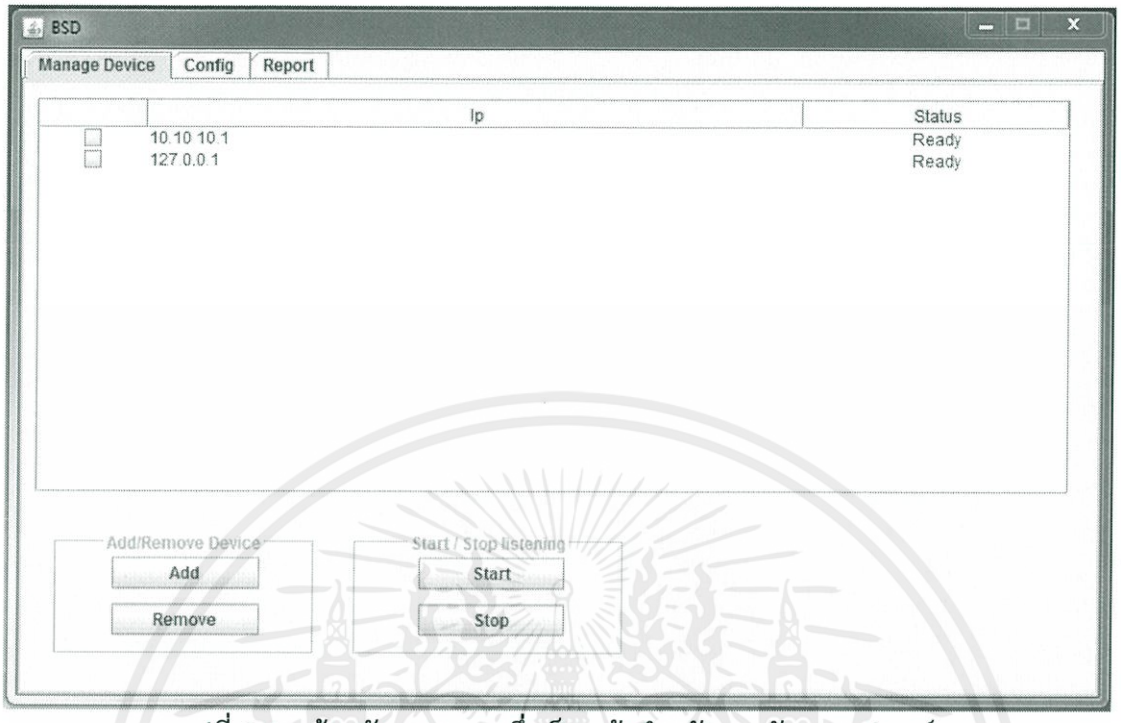
ส่วนของผู้ใช้นั้นสามารถตั้งค่าสำหรับการเพิ่มหมายเลขไอพีของเราเตอร์ สวิทช์ หรืออุปกรณ์ที่สามารถสร้างแพ็คเก็ตเน็ตโฟลว์ออกมาได้โดยสามารถเพิ่มและลบได้หลายอุปกรณ์โดยระบบจะรับข้อมูลจากอุปกรณ์เครือข่ายที่ได้ตั้งค่าไว้และเก็บลงฐานข้อมูลอัตโนมัติ ระบบจะทำการสร้างและจับคู่

อุปกรณ์กับฐานข้อมูลให้โดยอัตโนมัติจึงทำให้ผู้ใช้ไม่จำเป็นต้องกังวลเกี่ยวกับการจัดการเกี่ยวกับฐานข้อมูล นอกจากนั้นผู้ใช้ยังสามารถเริ่มและหยุดการรับข้อมูลจากอุปกรณ์เครือข่ายได้ตามต้องการ

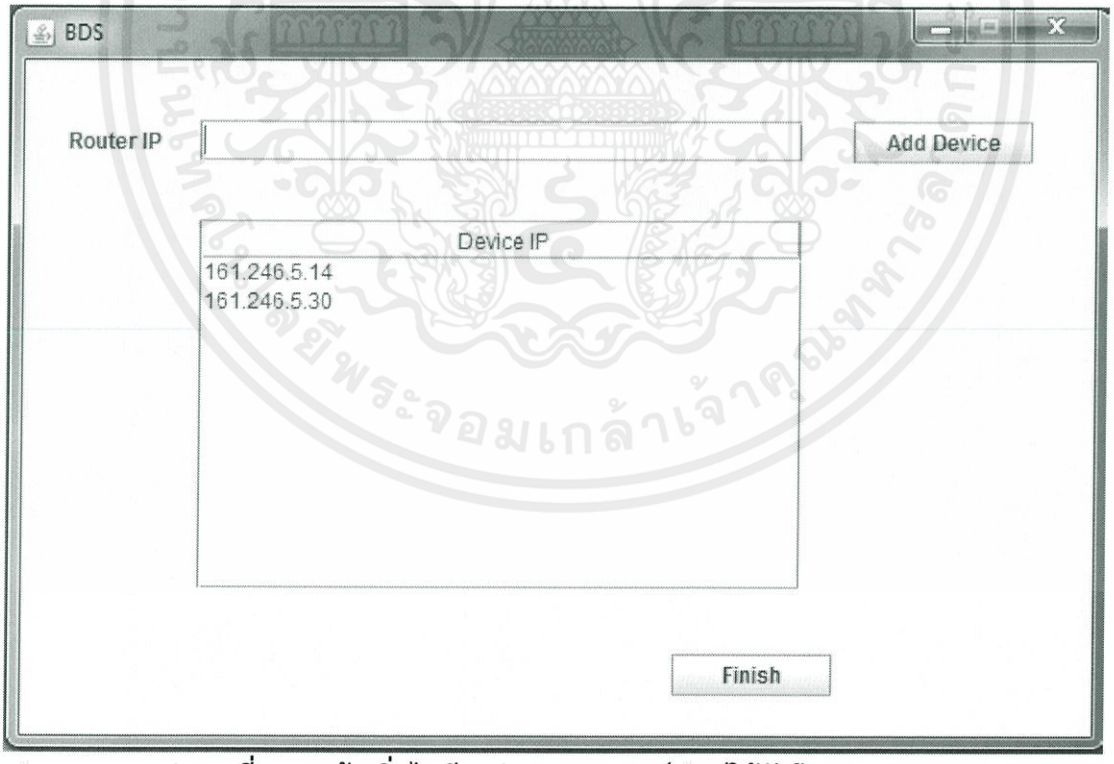
ผู้ใช้จำเป็นต้องมีการกำหนดค่าเริ่มต้นบางอย่างให้กับการรับข้อมูลและจัดการเกี่ยวกับซีเอฟทรีในเบื้องต้นก่อนคือการตั้งค่าหน่วยความจำที่สามารถใช้ได้หน่วยเป็นเมกะไบต์และต้องตั้งค่าเทรชโฮล์สำหรับรัศมีของแต่ละกลุ่มข้อมูลก่อนโดยค่าปกติจะถูกตั้งเป็น 0



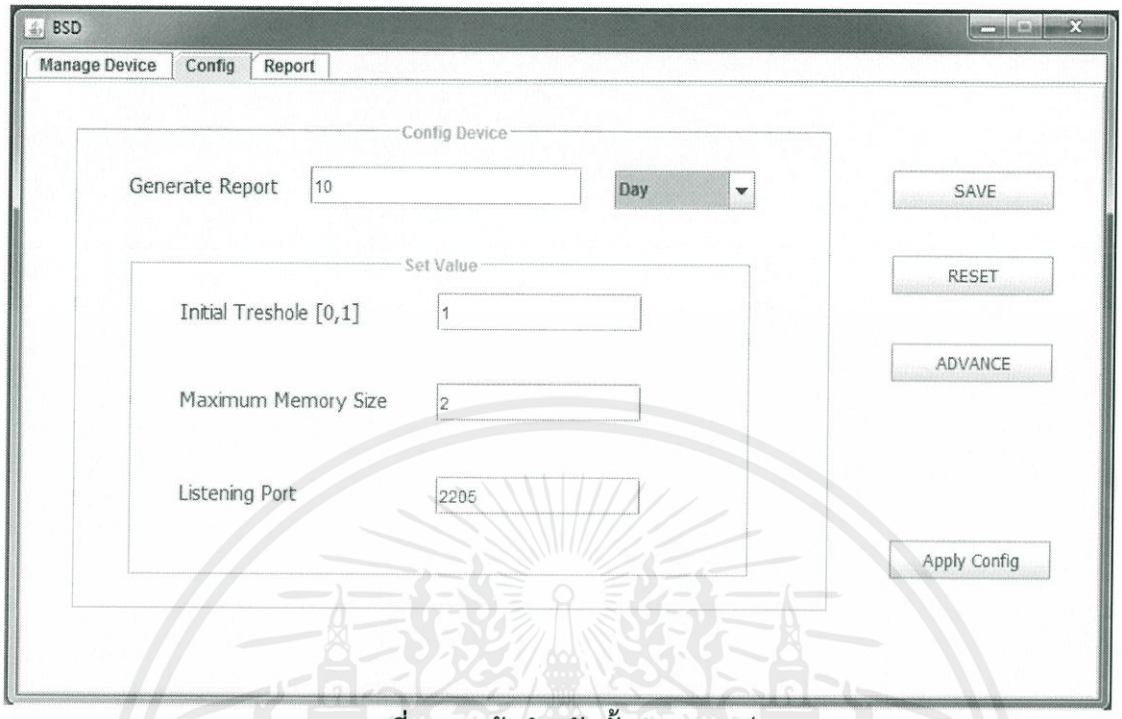
รูปที่ 3.6 ยูสเคสไดอะแกรม



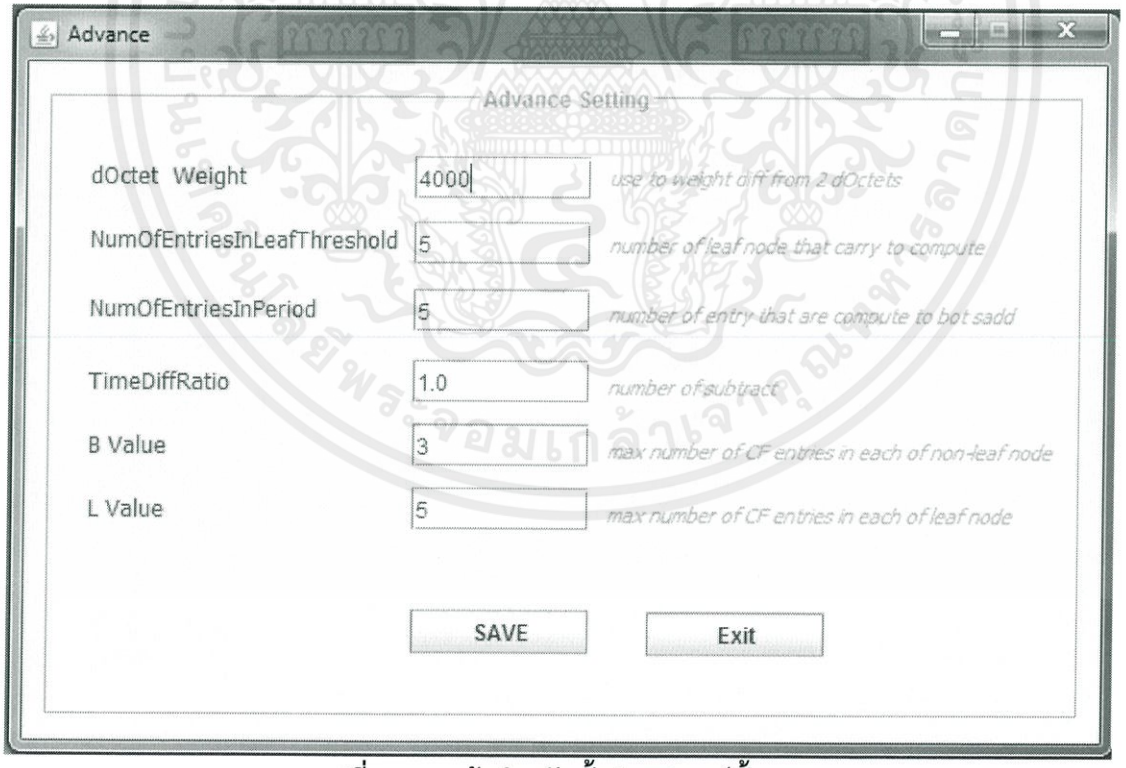
รูปที่ 3.7 หน้าหลักของระบบซึ่งเป็นหน้าสำหรับการจัดการอุปกรณ์



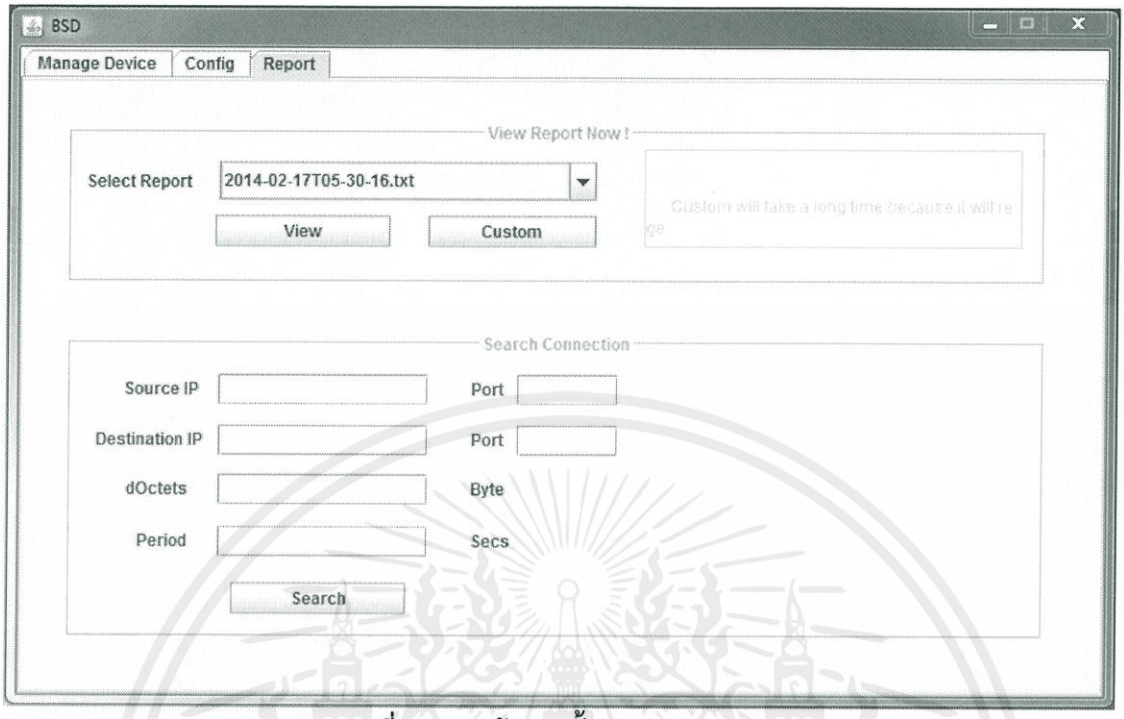
รูปที่ 3.8 หน้าเพิ่มไอพีแอดเดรสของเราท์เตอร์ให้กับโปรแกรม



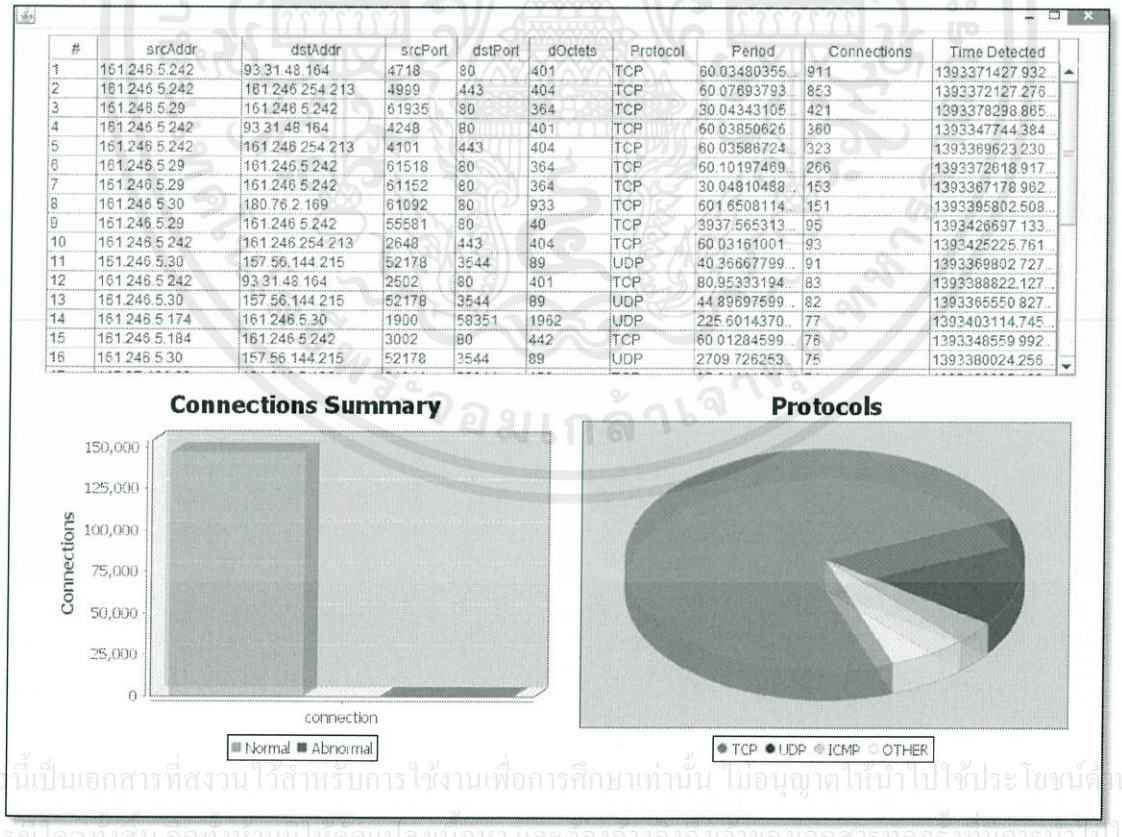
รูปที่ 3.9 หน้าสำหรับตั้งค่าอุปกรณ์



รูปที่ 3.10 หน้าสำหรับตั้งค่าอุปกรณ์ขั้นสูง



รูปที่ 3.11 หน้าการตั้งค่าการแสดงผล



#	srcAddr	dstAddr	srcPort	dstPort	dOctets	Protocol	Period	Connections	Time Detected
1	161.246.5.242	93.31.48.164	4718	80	401	TCP	60.03480355...	911	1393371427.932...
2	161.246.5.242	161.246.254.213	4999	443	404	TCP	60.07693793...	853	1393372127.276...
3	161.246.5.29	161.246.5.242	61935	80	364	TCP	30.04343105...	421	1393378298.865...
4	161.246.5.242	93.31.48.164	4248	80	401	TCP	60.03850626...	360	1393347744.384...
5	161.246.5.242	161.246.254.213	4101	443	404	TCP	60.03586724...	323	1393369623.230...
6	161.246.5.29	161.246.5.242	61518	80	364	TCP	60.10197469...	266	1393372618.917...
7	161.246.5.29	161.246.5.242	61152	80	364	TCP	30.04810488...	153	1393367178.962...
8	161.246.5.30	180.76.2.169	61092	80	933	TCP	601.6508114...	151	1393395802.508...
9	161.246.5.29	161.246.5.242	55581	80	40	TCP	3937.565313...	95	1393426697.133...
10	161.246.5.242	161.246.254.213	2648	443	404	TCP	60.03161001...	93	1393425225.761...
11	161.246.5.30	157.56.144.215	52178	3544	89	UDP	40.36667799...	91	1393369802.727...
12	161.246.5.242	93.31.48.164	2502	80	401	TCP	80.95333194...	83	1393388822.127...
13	161.246.5.30	157.56.144.215	52178	3544	89	UDP	44.89697599...	82	1393365550.827...
14	161.246.5.174	161.246.5.30	1900	58351	1962	UDP	225.6014370...	77	1393403114.745...
15	161.246.5.184	161.246.5.242	3002	80	442	TCP	60.01284599...	76	1393348559.992...
16	161.246.5.30	157.56.144.215	52178	3544	89	UDP	2709.726253...	75	1393380024.256...

รูปที่ 3.12 หน้ารายงานจำนวนการเชื่อมต่อและกราฟแสดงผลจำนวนบอทเน็ตที่มีในเครือข่าย

บทที่ 4
การทดลองและผลการทดลอง

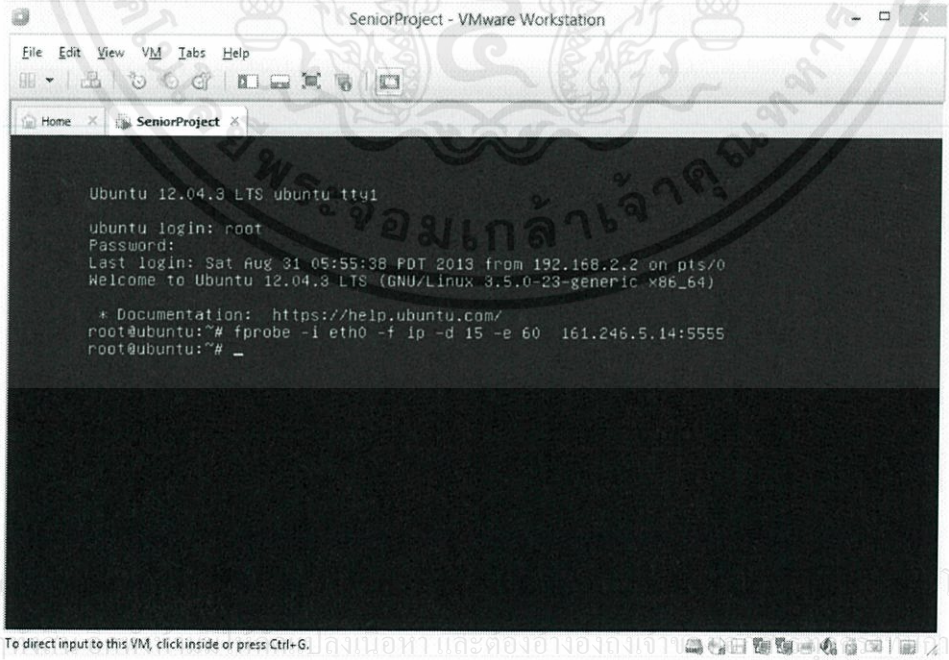
4.1 การทดลองโปรแกรมเอฟโพรบ (Fprobe)

ในการมอนิเตอร์เครือข่ายโดยการใช้แพ็คเก็ต เน็ตโฟลว์ ที่ซิสโก้ ได้พัฒนาขึ้นมานั้นต้องใช้อุปกรณ์เครือข่ายที่รองรับการใช้งานของ เน็ตโฟลว์ แต่เนื่องจากสาขาวิชาวิศวกรรมคอมพิวเตอร์ยังไม่มีอุปกรณ์ที่สามารถรองรับการทำงานของโพรโตคอล เน็ตโฟลว์ ได้ ผู้ดำเนินโครงงานจึงจำเป็นต้องหาวิธีที่จะมอนิเตอร์เครือข่ายได้อย่างมีประสิทธิภาพ

เอฟโพรบ เป็นซอฟต์แวร์หนึ่งที่มีความสามารถในการสร้างแพ็คเก็ต เน็ตโฟลว์ จากการแพ็คเก็ตต่างๆที่วิ่งผ่านเครื่องที่ลงซอฟต์แวร์นี้ โดยที่ เอฟโพรบ รองรับการทำงานของ เน็ตโฟลว์ เวอร์ชั่นที่ 5 ซึ่งมีรายละเอียดและข้อมูลต่างๆที่ผู้ดำเนินโครงงานต้องการ

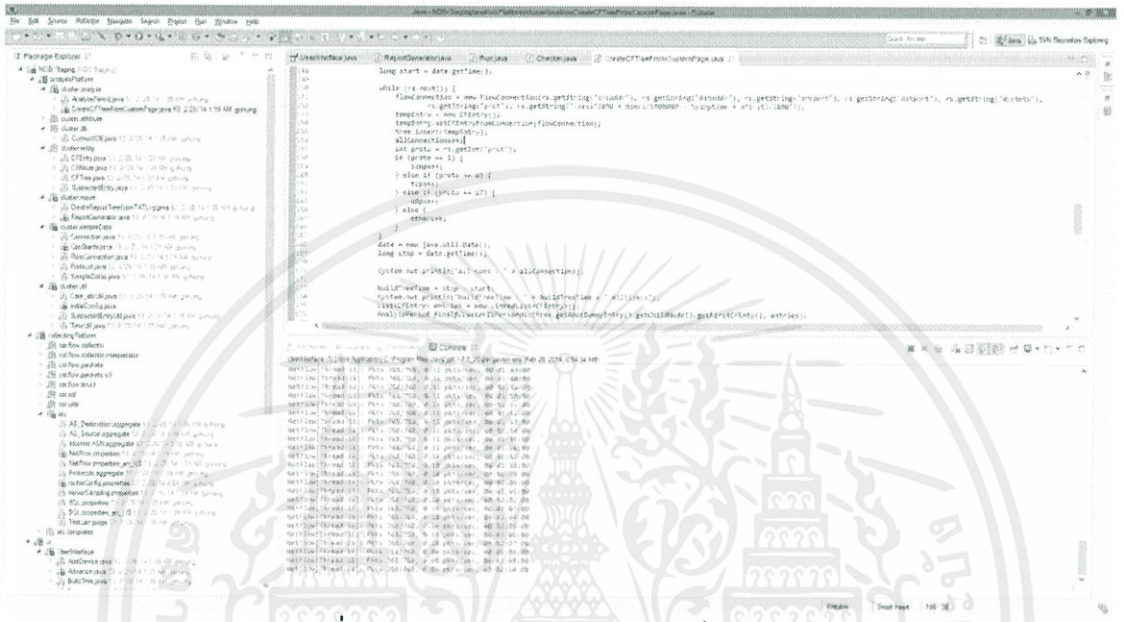
4.1.1 การติดตั้งและตั้งค่าเอฟโพรบ (Fprobe) และการเก็บข้อมูล NetFlow

การทำงานของ เอฟโพรบ นั้นต้องทำงานบนระบบปฏิบัติการ Linux เท่านั้นซึ่งผู้ทดลองได้ทำการติดตั้งระบบปฏิบัติการ Ubuntu Server เวอร์ชั่น 12.04.3 LTS บนเวอร์ชวลแมชชีน VMware เวอร์ชั่น 9



รูปที่ 4.1 แสดงการรันคำสั่ง fprobe –I eth0 ip –d 15 –e 60 161.246.5.14:5555

เพื่อสั่งให้เอฟโพรบทำการเก็บข้อมูลและสร้างเป็นแพ็คเก็ตเน็ตโฟลว์ส่งไปที่เครื่องคอมพิวเตอร์ของผู้วิจัย ผู้วิจัยได้ทำการพัฒนาโปรแกรมส่วนของการรวบรวมข้อมูล NetFlow ที่ได้รับมาจากเครื่องเซิร์ฟเวอร์ เพื่อกรองเฉพาะข้อมูลที่จำเป็นเพื่อเก็บข้อมูลลงฐานข้อมูล



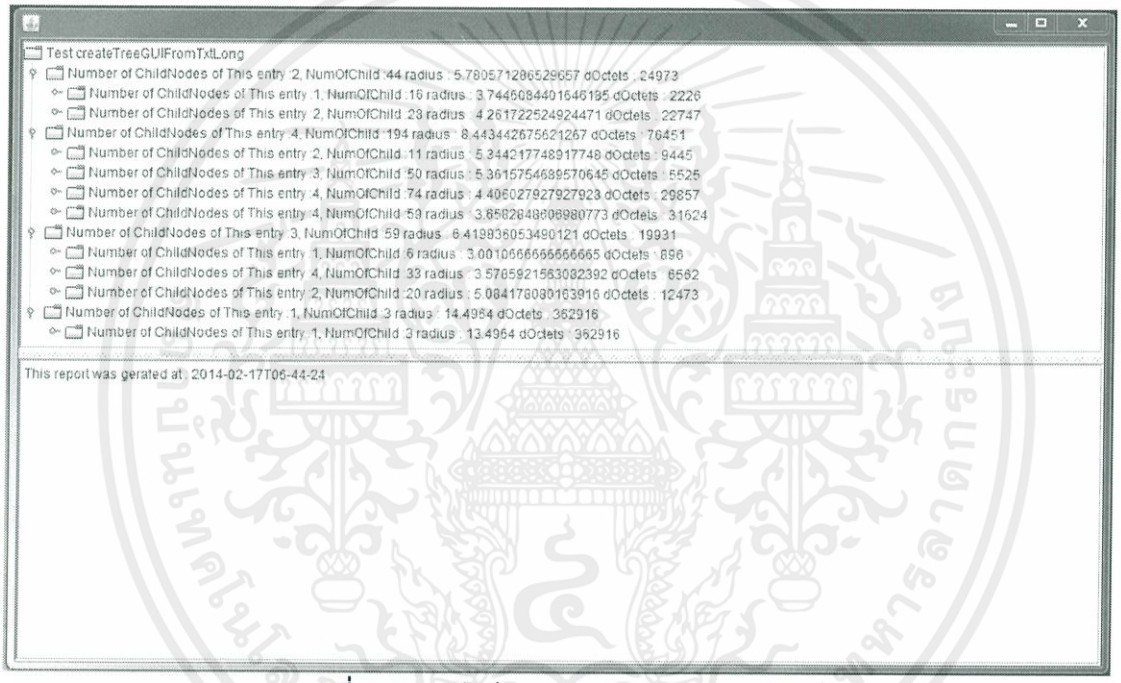
รูปที่ 4.2 รูปภาพแสดงการทดลองเก็บข้อมูล NetFlow

Router IP	SysUptime	Secs	Nsecs	Flow_Sequence	Engine_Type	Engine_ID	SrcAddr	DstAddr	NextHop	Input	Output	dPkts	dOctets	aFirst	aLast	SrcPort	DstPort
10.0.0.1	1048580995	1393340059	1562000	663108	0	0	161.246.5.184	161.246.5.242	0.0.0.0	0	0	6	442	1048558127	1048558147	1151	80
10.0.0.1	1048580995	1393340059	1562000	663108	0	0	161.246.5.242	161.246.5.184	0.0.0.0	0	0	4	1762	1048558151	1048558164	80	1152
10.0.0.1	1048580995	1393340059	1562000	663108	0	0	161.246.5.184	161.246.5.242	0.0.0.0	0	0	5	402	1048558150	1048558165	1152	80
10.0.0.1	1048580995	1393340059	1562000	663108	0	0	161.246.5.225	161.246.5.30	0.0.0.0	0	0	13	19955	1048560840	1048560855	5357	52750
10.0.0.1	1048580995	1393340059	1562000	663108	0	0	161.246.5.225	224.0.0.22	0.0.0.0	0	0	6	248	1048559828	1048559951	0	0
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.30	161.246.5.225	0.0.0.0	0	0	13	1491	1048556044	1048556066	52751	5357
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.184	161.246.5.242	0.0.0.0	0	0	6	442	1048558161	1048558171	1153	80
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.254.213	161.246.5.242	0.0.0.0	0	0	4	301	1048558297	1048558321	443	1413
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.225	161.246.5.30	0.0.0.0	0	0	2	2544	1048556043	1048556156	3702	61122
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.242	161.246.5.184	0.0.0.0	0	0	4	1762	1048558175	1048558185	80	1154
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.254.213	161.246.5.242	0.0.0.0	0	0	4	301	1048558266	1048558287	443	1411
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.242	161.246.254.213	0.0.0.0	0	0	6	404	1048558265	1048558286	1411	443
10.0.0.1	1048576000	1393340059	4903000	663136	0	0	161.246.5.242	93.31.48.164	0.0.0.0	0	0	5	401	1048558259	1048558264	1410	80
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.184	161.246.5.242	0.0.0.0	0	0	6	442	1048558174	1048558186	1154	80
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	93.31.48.164	161.246.5.242	0.0.0.0	0	0	5	268	1048558259	1048558265	80	1410
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.242	93.31.48.164	0.0.0.0	0	0	5	401	1048558287	1048558294	1412	80
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.225	161.246.5.30	0.0.0.0	0	0	12	19955	1048556046	1048556066	5357	52751
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.242	161.246.254.213	0.0.0.0	0	0	6	404	1048558296	1048558321	1413	443
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.30	94.245.121.251	0.0.0.0	0	0	1	89	1048558491	1048558491	63316	3544
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.242	161.246.5.184	0.0.0.0	0	0	4	1762	1048558161	1048558170	80	1153
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	93.31.48.164	161.246.5.242	0.0.0.0	0	0	5	268	1048558289	1048558296	80	1412
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	161.246.5.225	161.246.5.30	0.0.0.0	0	0	2	2544	1048556849	1048556014	3702	49564
10.0.0.1	1048580997	1393340064	1867000	663166	0	0	94.245.121.251	161.246.5.30	0.0.0.0	0	0	1	137	1048558770	1048558770	3544	63316
10.0.0.1	1048576000	1393340064	7545000	663194	0	0	161.246.5.242	93.31.48.164	0.0.0.0	0	0	5	401	1048558354	1048558359	1416	80
10.0.0.1	1048576000	1393340064	7545000	663194	0	0	161.246.5.242	161.246.254.213	0.0.0.0	0	0	6	404	1048558326	1048558354	1415	443
10.0.0.1	1048576000	1393340064	7545000	663194	0	0	93.31.48.164	161.246.5.242	0.0.0.0	0	0	5	268	1048558321	1048558325	80	1414

รูปที่ 4.3 รูปภาพแสดงฐานข้อมูล NetFlow

4.2 การทดลองการทำงานของอัลกอริทึมเบิร์ช

การทดลองการทำงานของอัลกอริทึมเบิร์ช ผู้วิจัยได้พัฒนาส่วนของการแบ่งกลุ่มข้อมูลโดยยึดหลักการตามเอกสารอ้างอิง ในขั้นแรกของการแบ่งกลุ่มข้อมูลจะต้องมีการเชื่อมต่อกับฐานข้อมูลเพื่อร้องขอข้อมูลดิบที่จะนำมาวิเคราะห์ จากนั้นจึงทำการแปลงข้อมูลดิบเป็นรูปแบบสำหรับการวิเคราะห์แล้วจึงแบ่งกลุ่มข้อมูล ซึ่งหลังจากการทดลองแบ่งกลุ่มข้อมูลเครือข่ายด้วยอัลกอริทึมเบิร์ชแล้วพบว่าเวลาที่ใช้ในการร้องขอข้อมูลจากฐานข้อมูลนั้นมักจะมีอัตราส่วนมากกว่าส่วนคำณวนเพื่อแบ่งกลุ่มข้อมูล



รูปที่ 4.4 ผลลัพธ์การแบ่งกลุ่มตามอัลกอริทึม

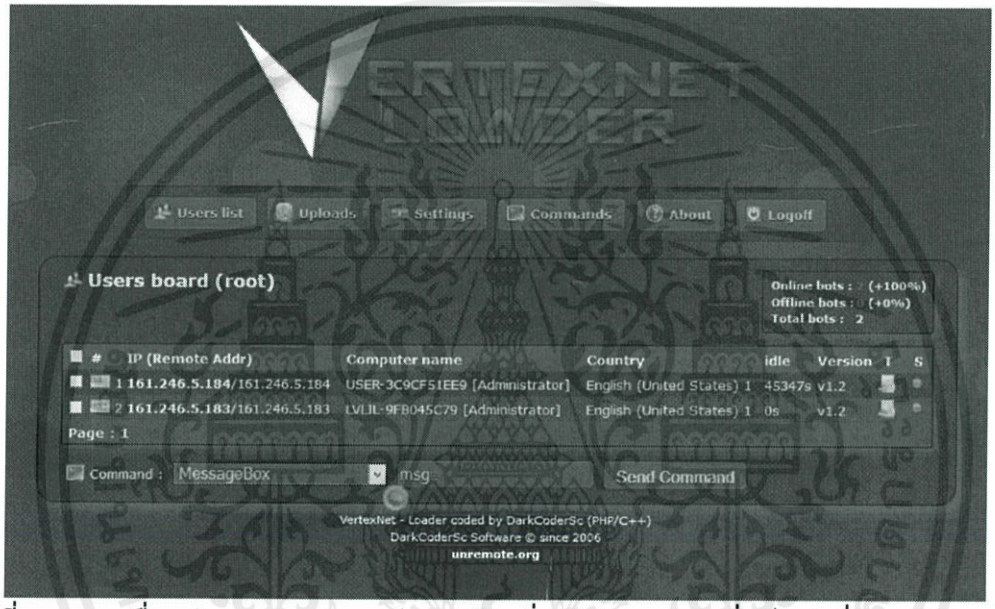
4.3 การทดลองการทำงานของ Botnet (VertexNet รุ่น 1.2.1)

4.3.1 ทดลองการติดตั้งบอท

ในการทดลองการทำงานของ Botnet นั้นได้เลือกใช้ Botnet ชื่อ VertexNet รุ่น 1.2.1 ซึ่งเป็น Botnet ที่สามารถหาใช้งานได้ง่ายจากการดาวน์โหลดทางอินเทอร์เน็ต VertexNet นั้นประกอบไปด้วยสองส่วนคือส่วนที่เป็น Command and Control Server (C&C) ซึ่งเป็นส่วนที่ทำงานเป็นตัวสั่งงานบอทและ Loader ซึ่งทำหน้าที่เป็นส่วนแพร่กระจายบอท

การทดลองการทำงานของบอทนั้นได้ถูกจัดการให้มีการทำงานอยู่ในสิ่งแวดล้อมเสมือน โดยติดตั้ง C&C บนเวอร์ชวลแมชีนโดยใช้ระบบปฏิบัติการ XP มีหน่วยความจำหลัก 512 เมกะไบต์และ

หน่วยความจำรอง 30 กิกะไบต์และได้จำกัดการเชื่อมต่อสู่เครือข่ายภายนอกของเวอร์ชวลแมชีนเพื่อเป็นการควบคุมไม่ให้ส่งผลกระทบต่อเครือข่ายภายนอก C&C จะทำงานเป็น HTTP เซิร์ฟเวอร์สำหรับสั่งงานและรับการเชื่อมต่อจากบอท โดย C&C ถูกติดตั้งโดยกำหนดหมายเลขไอพี 161.246.5.182 /24 และได้ทำการติดตั้งบอทจำนวน 2 เครื่องคือหมายเลขไอพี 161.246.5.183/24 และ 161.246.5.184/24 ซึ่งผลของการทดลองการเชื่อมต่อระหว่างบอทและ C&C คือตัว C&C เซิร์ฟเวอร์สามารถติดต่อกับบอทได้นอกจากนั้นยังสามารถรับรู้ได้ว่าบอทนั้นยังอยู่บนเครือข่ายหรือไม่ดังรูป



รูปที่ 4.5 การเชื่อมต่อระหว่าง C&C กับบอทโดยที่จะมีค่า “idle” ซึ่งเป็นค่าที่บอกว่าบอทนั้นมีการเคลื่อนไหวหรือไม่ ถ้าหากมีการเคลื่อนไหวของบอทเช่นผู้ใช้มีการขยับเมาส์บอทก็จะส่งแพ็คเก็ตเพื่อรีเซ็ตค่า “idle” เป็น 0

4.3.2 ทดลองการสั่งงานบอทและการสังเกตผล

ในการสั่งงานบอทของ C&C เซิร์ฟเวอร์นั่นจะมีคำสั่งที่ได้ถูกกำหนดไว้ให้แล้วว่าจะต้องใช้คำสั่งแบบใดโดยคำสั่งที่ให้ใช้นั้นมีอยู่ทั้งหมด 14 คำสั่งซึ่งประกอบไปด้วย

msg : การส่งข้อความ

exec : การรันคำสั่ง (Command Prompt) ของเครื่องที่เป็นบอท

close : ปิดการทำงานของ loader ของเครื่องที่เป็นบอท

urldl : ดาวน์โหลดไฟล์จากเครื่องที่เป็นบอท

getproc : คำสั่งเพื่อดูโพรเซสที่เปิดทำงานอยู่ในเครื่องที่เป็นบอท

setkeylogger : เปิดการทำงานของ key logger

getlogs : อ่านไฟล์บันทึกของ key logger

readfile : อ่านไฟล์ที่เครื่องที่เป็นบอท

uninstall : ถอนการติดตั้ง loader ออกจากเครื่องที่เป็นบอท

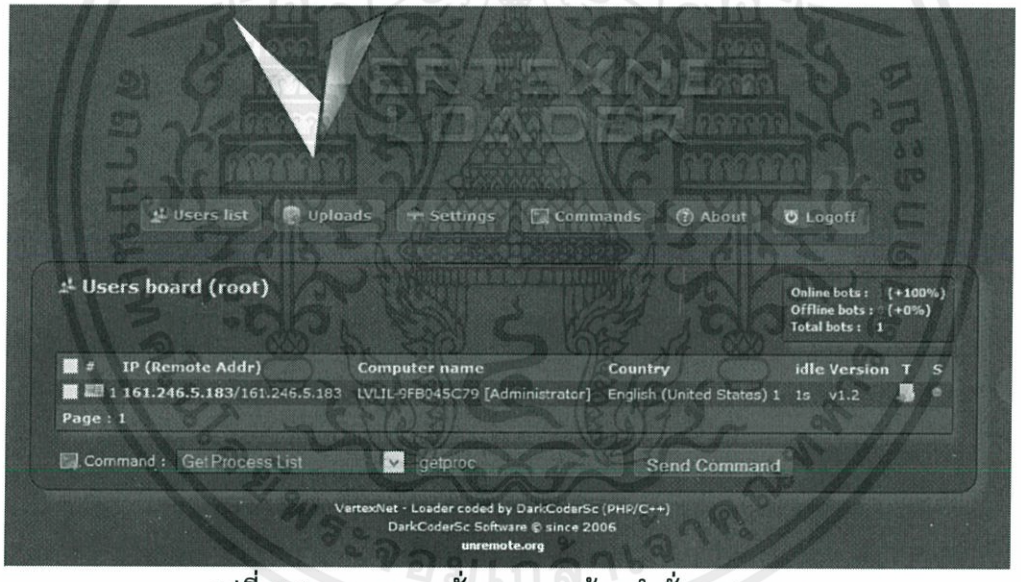
httpflood : สั่งให้บอททำการโจมตีแบบ httpflood ไปยังเครื่องเป้าหมาย

remoteshell : ทำการรีโมทใช้งานเชลซึ่งสามารถใช้งานบนเครื่องบอทที่ใช้งานลินุกซ์

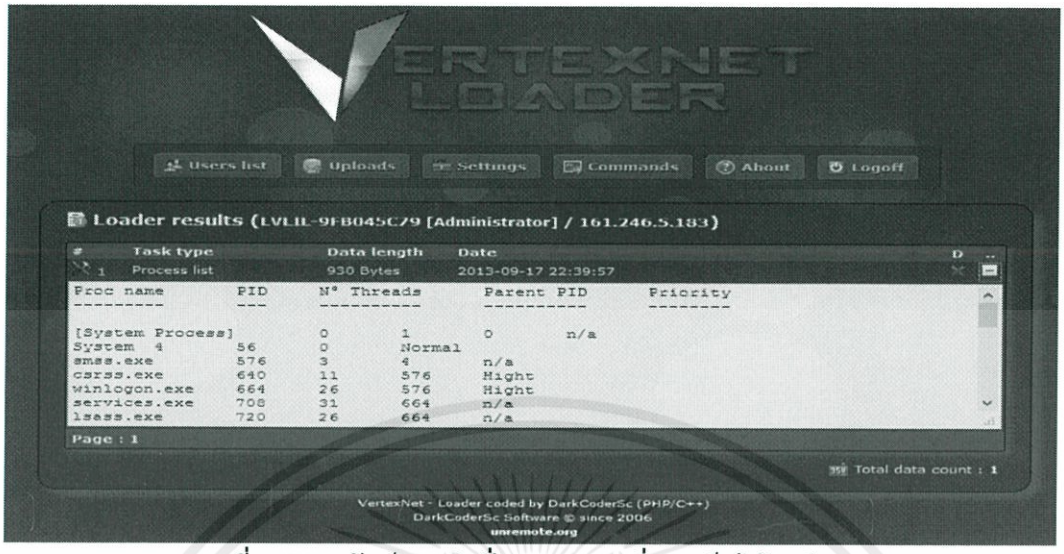
visitpage : สั่งงานให้เครื่องที่เป็นบอททำการเปิดเว็บเพจที่ต้องการช่วงขณะหนึ่ง

update : คำสั่งอัพเดท loader ให้เป็นเวอร์ชั่นล่าสุด

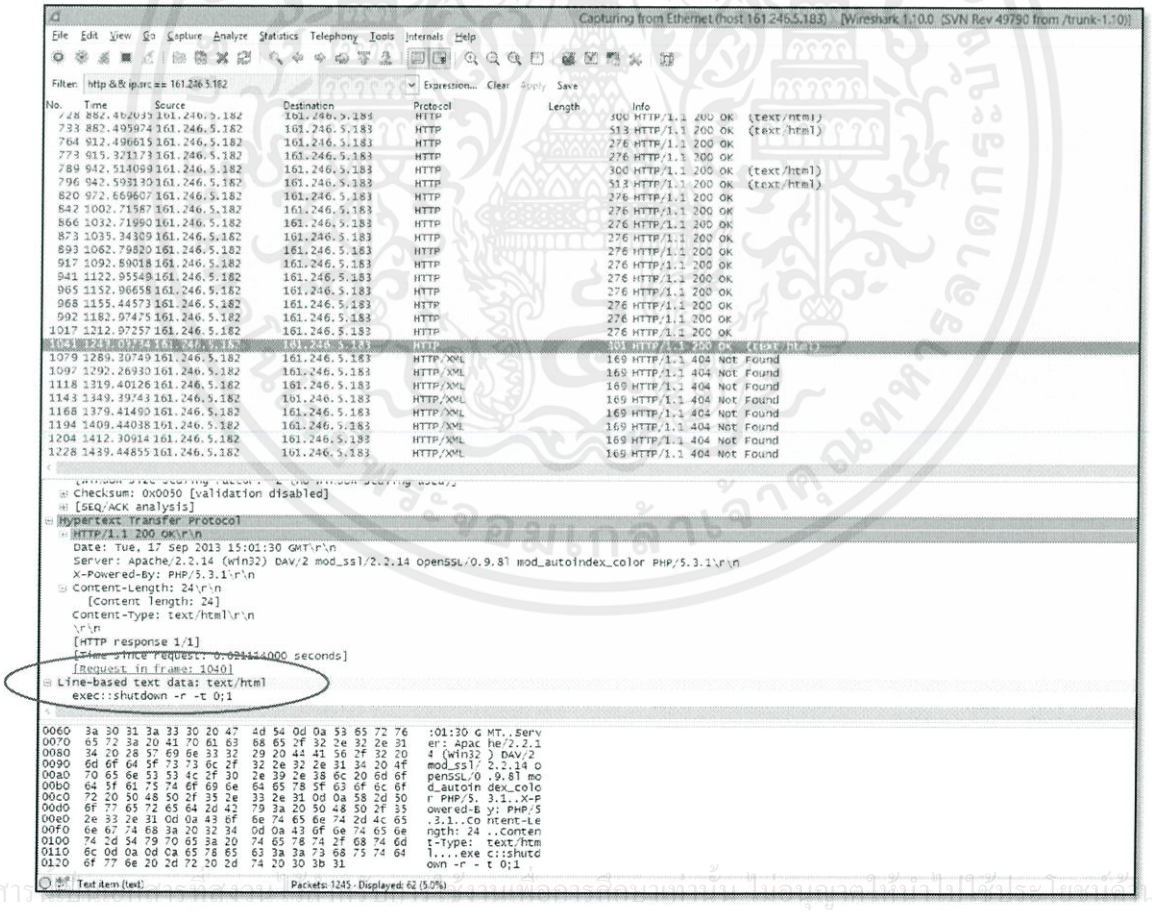
การทดลองการสั่งงานของ C&C ได้ทดลองใช้งานคำสั่ง “getproc” และคำสั่ง “exec : shutdown –r –t 0” ซึ่งคำสั่ง getproc นั้นมีจะมีผลทำให้ได้รับการรีเทิร์นโพรเซสที่ทำงานอยู่บนเครื่องที่เป็นบอท และคำสั่ง shutdown เป็นคำสั่งซึ่งสั่งให้เครื่องบอททำการเริ่มการทำงานใหม่ ซึ่งทั้งสองคำสั่งมีผลในทันทีที่ C&C สามารถติดต่อกับบอทได้ใช้เวลาประมาณ 3 -5 วินาที



รูปที่ 4.6 การทดลองสั่งงานบอทด้วยคำสั่ง getproc



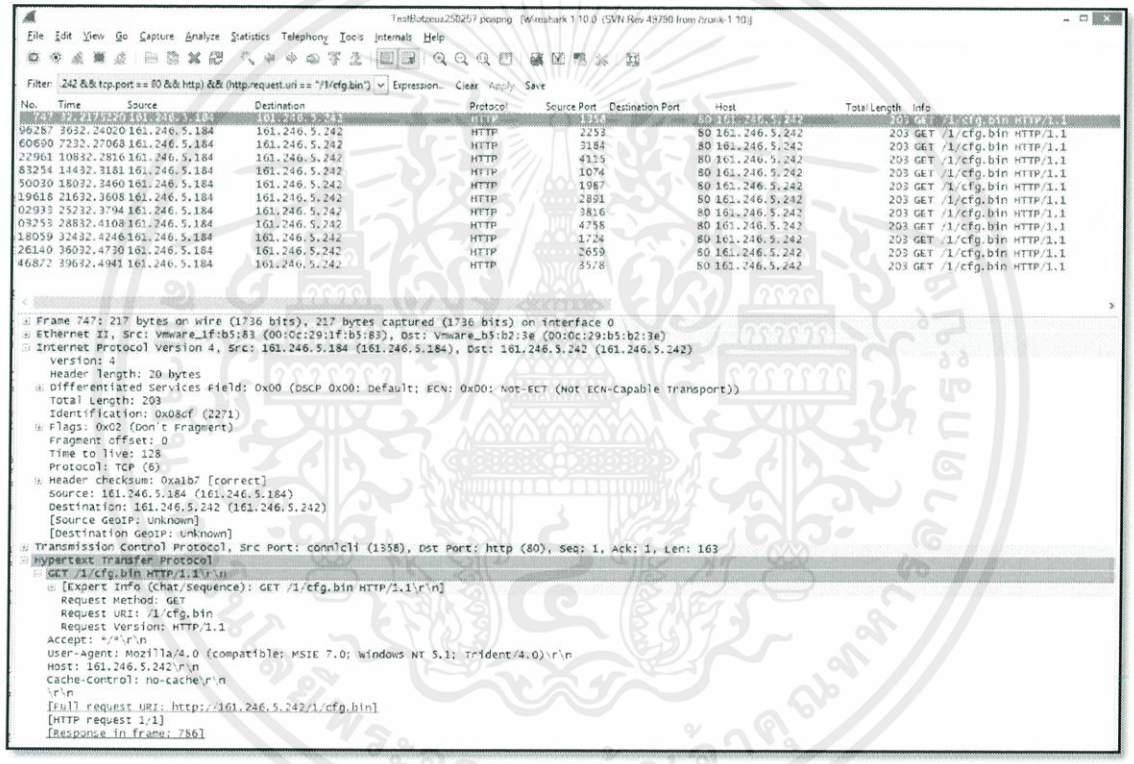
รูปที่ 4.7 ผลลัพธ์ของคำสั่ง getproc ซึ่งจะเห็นได้ว่ามีการรีเทิร์นโพรเซสที่ทำงานอยู่บนเครื่องที่เป็นบอท



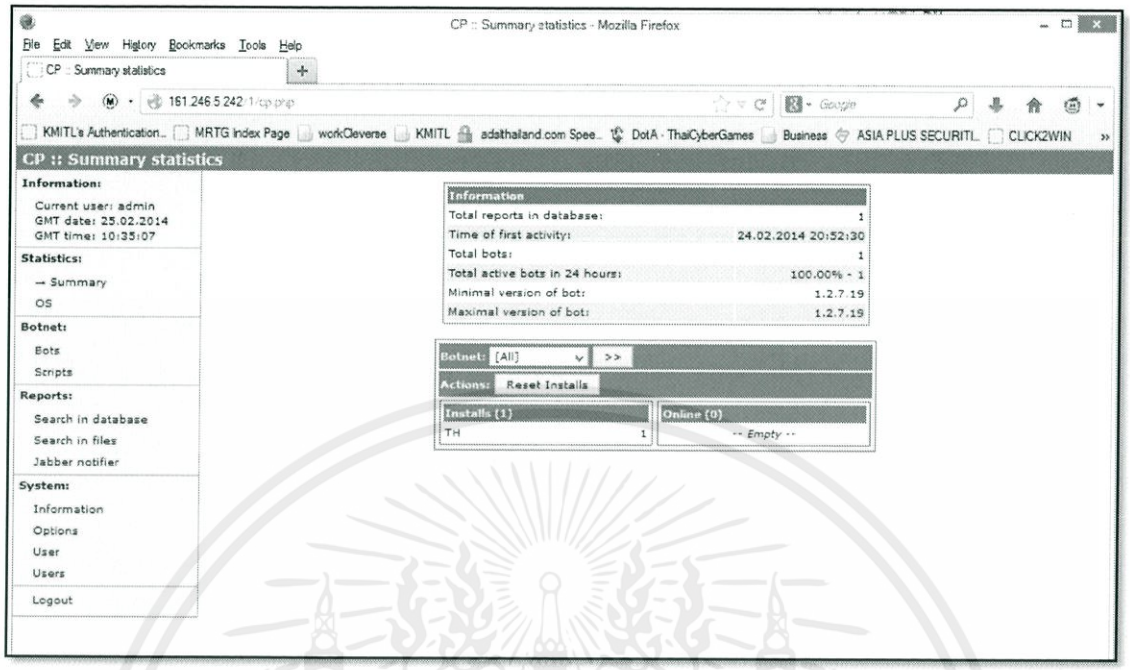
รูปที่ 4.8 การทดลองการดักจับแพ็คเก็ตเพื่อสังเกตการสั่งงานของ C&C โดยเป็นการดักจับแพ็คเก็ตขณะที่มีการสั่งคำสั่ง shutdown –r –t 0

4.4 การทดลองการทำงานของ Botnet (Zeus รุ่น 2.1.0.1)

การทดลองการใช้งานบอท Zeus รุ่น 2.1.0.1 นั้น ผู้วิจัยสามารถดักจับแพ็คเกจของ Zeus ได้ และจากผลการทดลองวิเคราห์เห็นว่า การทำงานของ Zeus มีลักษณะเป็นคาบการทำงานซ้ำๆตามทฤษฎีที่ได้อ้างถึงก่อนหน้า ทำให้ผู้วิจัยสามารถพัฒนาระบบเพื่อวิเคราะห์ตรวจหาคาบการทำงานได้ นอกจากนั้นยังเห็นว่าการทำงานของ Zeus นั้นจะมีการร้องขอไปยัง HTTP Server โดยมีพารามิเตอร์ซ้ำเดิมคือร้องขอหน้า page /cfg.bin เสมอซึ่งแสดงให้เห็นว่ามีความเป็นไปได้ที่จะสามารถพัฒนา IDS เพื่อตรวจจับลักษณะดังกล่าวได้



รูปที่ 4.9 ผลที่ได้จากการดักจับแพ็คเกจของ Wireshark ที่เป็นของบอท Zeus ที่ร้องขอไปที่ /cfg.bin



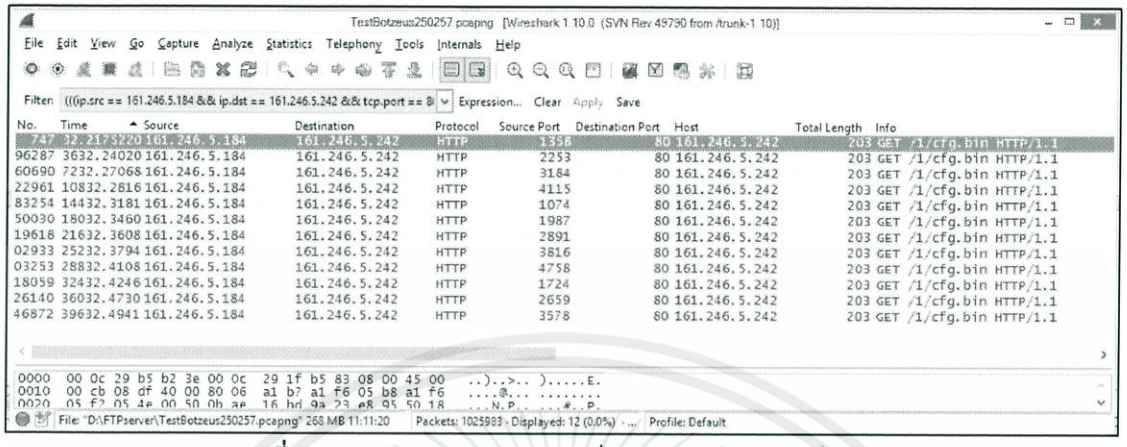
รูปที่ 4.10 หน้าเว็ปของบอทมาสเตอร์ที่คอยควบคุมและส่งคำสั่งให้กับบอทลูก

อย่างไรก็ตามการเชื่อมต่อของบอทกับเซิร์ฟเวอร์นั่นจะมีลักษณะซ้ำกันในคาบเวลาที่ซ้ำกันเสมอ แต่ผู้ใช้งานบอทก็สามารถที่จะตั้งค่าการทำงานของบอทให้มีคาบเวลาที่ต่างๆกันไปได้ซึ่งทำให้ยากต่อการตรวจจับ ดังนั้นการตรวจจับบอทโดยใช้พฤติกรรมทางเครือข่ายสามารถตรวจจับได้จากเพียงการค้นหาการทำงานเป็นคาบๆ ของการเชื่อมต่อใดๆเท่านั้น

4.4 การทดลองการวิเคราะห์การสื่อสารของบอทกับเซิร์ฟเวอร์

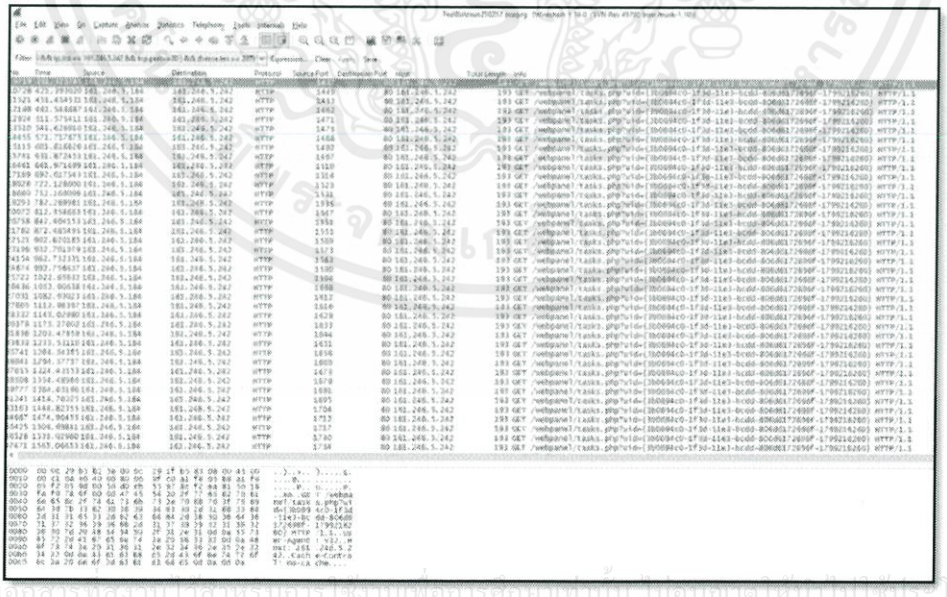
4.4.1 พฤติกรรมการเชื่อมต่อเป็นคาบของบอทเน็ต Zeus รุ่น 2.1.0.1 และ VertexNet รุ่น 1.2.1

ลักษณะการสร้างการเชื่อต่อเป็นช่วงๆ โดยที่มีการส่งข้อมูลที่มีขนาดเท่าๆกันไปยังหมาลเลขไอพีปลายทางซ้ำกันหลายๆครั้งทำให้เป็นที่น่าสังเกตว่าพฤติกรรมดังกล่าวอาจไม่ใช่การส่งข้อมูลปกติ โดยจากการทดลองการทำงานของบอทเน็ต Zeus โดยตั้งค่าให้มีการส่งข้อมูลเพื่อติดต่อกับเซิร์ฟเวอร์เป็นคาบมาตรฐานของบอทรุ่นนี้แล้วอาศัยเครื่องมือดักจับแพ็คเกจ Wireshark ดักจับแพ็คเกจ(เป็นเวลา 12 ชั่วโมงนับแต่เครื่องเหยื่อติดบอท) ของการส่งข้อมูลพบว่าบอท Zeus มีการส่งข้อมูลเป็นคาบอย่างมีนัยสำคัญคือการส่งข้อมูลติดต่อกับเครื่องเซิร์ฟเวอร์เป็นคาบเวลา 60 นาทีโดยมีขนาดข้อมูลที่เท่ากันเสมอคือ 207 ไบต์ ติดต่อกัน โดยที่เมื่อเปรียบเทียบกับการส่งข้อมูลปกติของผู้ใช้งานจะเห็นความแตกต่างอย่างเห็นได้ชัด



รูปที่ 4.11 รูปแสดงการกรองแพ็คเกจจากหมายเลขไอพี, หมายเลขพอร์ต, โพรโตคอลและปริมาณข้อมูล

จากการทดลองดักจับแพ็คเกจของเครื่องที่ถูกคุกคามโดย VertexNet รุ่น 1.2.1 พบว่าเครื่องที่ถูกคุกคามนั้นมีการสร้างการเชื่อมต่อไปยังเครื่องเซิร์ฟเวอร์เช่นเดียวกันกับ Zeus แต่แตกต่างกันที่เส้นทาง (Path) ที่ร้องขอการเชื่อมต่อเนื่องจากมีการตั้งค่าไม่เหมือนกัน นอกจากนั้นขนาดข้อมูลที่มีการสื่อสารกันและคาบเวลาที่ใช้ในการสื่อสารยังต่างกัน โดยที่ VertexNet มีการสร้างการเชื่อมต่อทุกๆ 30 วินาทีและมีขนาดข้อมูลคือ 193 ไบต์ดังรูป



รูปที่ 4.12 ผลที่ได้จากการกรองแพ็คเกจที่มีการส่งคำร้องขอไปที่ VertexNet เซิร์ฟเวอร์โดยมีการเชื่อมต่อส่งไปทุกๆ 30 วินาที

อย่างไรก็ตามจากการวิเคราะห์แพ็คเกจทั้งหมดทำให้สามารถสรุปได้ว่าการเชื่อมต่อของบอทจะมีการปะปนอยู่กับการเชื่อมต่อของเครื่องปกติแต่จะมีการเชื่อมต่อบางอย่างของบอทที่จะมีลักษณะที่จำเพาะและมีความซ้ำซ้อนที่สามารถตรวจสอบได้โดยการพัฒนาโปรแกรมเพื่อตรวจจับพฤติกรรมเหล่านี้

4.5 การทดลองการคำนวณตามเอกสารอ้างอิง [1]

ในการวิเคราะห์พฤติกรรมของเครือข่ายนั้นจำเป็นต้องมีการคำนวณพารามิเตอร์ที่ต่างชนิดกันซึ่งสามารถออกได้เป็น 3 ชนิดคือ ข้อมูลเชิงเลขคณิต เช่น จำนวนไบต์ ข้อมูลเชิงประเภท เช่น แฟลกของทีซีพีเฮดเดอร์ และข้อมูลเชิงลำดับชั้น เช่น หมายเลขไอพี เป็นต้น ในการคำนวณและเปรียบเทียบข้อมูลแต่ละชนิดจะมีกระบวนการที่แตกต่างกันซึ่งได้กล่าวไว้ในบทที่สาม

ในการทดลองการคำนวณตามอัลกอริทึมเบิร์ชนั้นได้กำหนดข้อมูล 3 แพ็คเก็ตซึ่งเป็นข้อมูลที่ได้มีการเก็บจากสถานการปกติ 1 แพ็คเก็ตและเก็บจากแพ็คเก็ตของ VertexNet อีก 2 แพ็คเก็ตซึ่งข้อมูลแต่ละชุดมีรูปแบบดังนี้

Source Address, Destination Address, Source Port, Destination Port, Bytes, Protocol

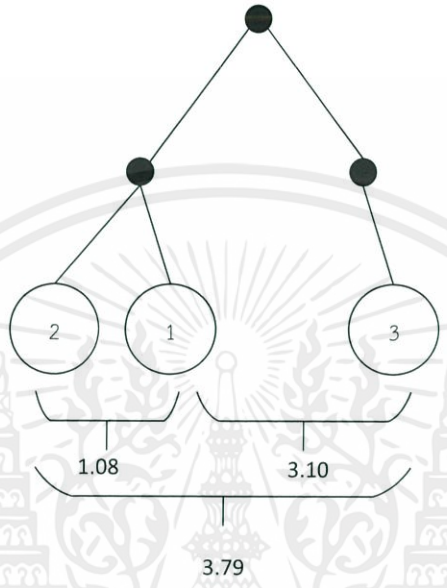
1. 161.246.5.183, 161.246.5.182, 1050, 80, 40 TCP
2. 161.246.5.184, 161.246.5.182, 1055, 80, 144 UDP
3. 161.246.5.86, 38.111.126.203, 4804, 23468, 144 TCP

กำหนดให้ เริ่มต้นทำการพิจารณาแพ็คเก็ตที่ 1 ทำการสร้างเป็น CF Tree ที่มี 1 โหนดจากนั้นทำการเพิ่มแพ็คเก็ตที่ 2 เข้ามาและสร้างเป็น CF Tree ที่มี 2 โหนด จากนั้นจึงพิจารณาโหนดที่ 3 กำหนด N = 2 หมายความว่า 1 โหนดจะมีสมาชิกได้ไม่เกิน 2 โหนดเท่านั้น กำหนดTCP={0,1}, UDP={1,0} เมื่อทำการคำนวณ CF Entry ของแต่ละชุดข้อมูลและจะสามารถเริ่มเปรียบเทียบแต่ละ Entry เพื่อสร้าง CF Tree ได้

N	Source Address		Destination Address		SrcPort		DstPort		Bytes	Protocol
	c	Radius	c	Radius	c	Range	c	Range		
1	161.246.5.183/32	161.246.5.183	161.246.5.182/32	161.246.5.182	1050/16	1050	80/16	80	40	{01}
1	161.246.5.183/32	161.246.5.183	161.246.5.184/32	161.246.5.184	1055/16	1055	80/16	80	144	{10}
1	38.111.126.203/32	38.111.126.203	161.246.5.86/32	161.246.5.86	4804/16	4804	23168/16	23468	144	{01}

รูปที่ 4.13 ผลการคำนวณ CF Entry ของข้อมูลแต่ละชุด

จากการคำนวณสามารถหาค่าความแตกต่างระหว่างชุดข้อมูลได้ซึ่งผลที่ได้คือ ข้อมูลชุดที่ 1 มีความใกล้เคียงกับชุดที่ 2 มากกว่าชุดที่ 3 และข้อมูลชุดที่ 3 มีความแตกต่างจากชุดที่ 2 มากกว่าชุดที่ 1 ดังนั้นทำให้สามารถจัดเป็น CF Tree ได้ดังนี้



รูปที่ 4.14 การจำลอง CF Tree ที่ได้จากการคำนวณ

บทที่ 5
บทสรุปและข้อเสนอแนะ

จากการศึกษาการแนวทางการทำงานต่างๆเพื่อหาวิธีในการตรวจจับบอทเน็ตอย่างมีประสิทธิภาพทำให้ทราบถึงกระบวนการต่างๆมากมายที่ถูกประยุกต์ใช้เพื่อจุดประสงค์ดังกล่าว สำหรับการตรวจจับบอทเน็ตโดยใช้กระบวนการสังเกตพฤติกรรมการใช้งานเครือข่ายของคอมพิวเตอร์ต่างๆภายในเครือข่ายนั้นมีปัญหาหลายอย่างซึ่งทำให้เกิดปัญหาของการศึกษาค้นคว้าเป็นต้นว่า การแบ่งกลุ่มข้อมูล การทำงานกับข้อมูลที่มีจำนวนมากหรือแม้แต่การจัดการความเร็วในการประมวลผล เป็นต้น

5.1 บทสรุป

โครงงานเรื่องระบบตรวจจับบอทเน็ตจากพฤติกรรมการใช้งานเครือข่ายได้ถูกออกแบบมาเพื่อใช้ในการตรวจจับบอทเน็ตจากการสังเกตการใช้งานเครือข่ายที่ใกล้เคียงกับกลุ่มข้อมูลตัวอย่างซึ่งแตกต่างจากเทคนิคการตรวจจับโดยใช้ลักษณะเฉพาะของบอทเน็ตเป็นสิ่งบ่งชี้ถึงความน่านะเป็นที่จะเป็นบอท ข้อดีของการใช้การสังเกตพฤติกรรมแทนการใช้ลักษณะบ่งชี้ดังกล่าวคือความสามารถในการตรวจหาบอทจากพฤติกรรมใกล้เคียงโดยจะสามารถตรวจจับบอทเน็ตได้ถึงแม้ว่าจะไม่มีการอัพเดทฐานข้อมูลของกลุ่มตัวอย่างเป็นประจำอย่างการใช้งานแอนตี้ไวรัสหรือระบบป้องกันเครือข่ายๆแบบอื่นๆ ซึ่งจะทำให้สามารถป้องกันความเสี่ยงที่อาจเกิดขึ้นได้อย่างกว้างขวางมากขึ้นกว่าเดิม

5.2 ปัญหา อุปสรรค และแนวทางการแก้ไข

1) เอกสารที่เกี่ยวกับการทำงานของอัลกอริทึมเบิร์ชยังมีจำนวนน้อยและส่วนมากเป็นชิ้นงานนำเสนอซึ่งคัดลอกกันมาเป็นทอดๆ ทำให้สามารถศึกษากรณีตัวอย่างและทำความเข้าใจได้ยากมากขึ้นเพราะมีจำนวณกรณีตัวอย่างที่แตกต่างกันน้อย
2) การศึกษาพฤติกรรมของการทำงานของเครือข่ายจะได้ข้อมูลซึ่งส่วนมากไม่ใช่ข้อมูลที่สามารถนำไปคำนวณทางคณิตศาสตร์ได้โดยตรง ผู้วิจัยจึงจำเป็นต้องมีการศึกษาการแปลงหรือแทนค่าและกำหนดกระบวนการทางคณิตศาสตร์ให้กับข้อมูลดังกล่าวอย่างเหมาะสมก่อนที่จะนำไปคำนวณ
3) เป้าหมายของโครงงานคือระบบที่สามารถตรวจจับบอทเน็ตได้อย่างมีประสิทธิภาพโดยใช้การสังเกตพฤติกรรมและการแบ่งกลุ่มข้อมูลของการใช้งานเครือข่ายดังนั้นจึงต้องมีการทำงานกับ

ข้อมูลจำนวนมากและใช้เวลานานในการเฝ้าสังเกตพฤติกรรมทำให้ระบบไม่เป็นแบบตามเวลาจริง

4) การวิเคราห์การใช้งานเครือข่ายของโครงงานนั้นอาศัยการทำงานของโพรโตคอล เน็ตโฟลว์ ที่พัฒนาโดยซิสโก้แต่เนื่องจากอุปกรณ์ของสาขาวิชาไม่รองรับการใช้งานดังกล่าว ผู้วิจัยจึงต้องมีการศึกษาการใช้ซอฟต์แวร์อื่นทดแทนที่สามารถสร้างแพ็คเก็ต เน็ตโฟลว์ เพื่อให้ข้อมูลที่ระบบต้องการได้
5) การบันทึกพฤติกรรมบอทเน็ตจำเป็นอย่างยิ่งที่ต้องมีการศึกษาพฤติกรรมในระบบปิดก่อนซึ่งต้องมีการติดตั้งและศึกษาจากเวอร์ชวลแมชีนหลายเครื่องซึ่งใช้ทรัพยากรระบบเป็นอย่างมาก
6) การศึกษาบอทเน็ตนั้นต้องใช้บอทเน็ตจริงหรือใกล้เคียงซึ่งหาได้ยากและง่ายต่อการถูกหลอกลวงให้ติดไวรัสหรือมัลแวร์

5.3 แนวทางในการพัฒนาต่อ

1) นำระบบที่พัฒนาได้ไปผนวกกับระบบเรียนรู้ของคอมพิวเตอร์ซึ่งจะทำให้ระบบสามารถตรวจจับบอทเน็ตหรือมัลแวร์อย่างอื่นได้อย่างมีประสิทธิภาพ
2) สามารถนำระบบไปพัฒนาเพื่อวิเคราะห์และแบ่งกลุ่มการใช้งานเครือข่ายเพื่อตรวจหาความพิดปกติของการใช้งานเครือข่ายได้

บรรณานุกรม

[1] Abdun Naser Mahmood, Christopher Leckie and Parampali Udaya. "**An Efficient Clustering Scheme to Exploit Hierarchical Data in Network Traffic Analysis**" IEEE Trans. On Knowle, vol. 20, no. 6, 2007

[2] T. Zhang, R. Ramakrishnan and M. Livny. "**Birch: An Efficient Data Clustering Method for Very Large Databases**" ACM, inc. 1996. pp103-114

[3] Meisam E. "**botAnalytics: Improving HTTP-Based Botnet Detection By Using Network Behavior Analysis System**" M. Thesis of Faculty of Computer Science and Information Technology of University of Malaya. 2010.

[4] Alexander B. "**Network Characterization For Botnet Detection Using Statistical-Behavioral Methods**" M. Thesis of Thayer School of Engineering Dartmouth College Hanover. 2009

[5] Yuanyuan Z. "**On Detection of Current and Next-Generation Botnets**" Ph.D Thesis of Michigan of University. 2012

[6] Jae-Seo, HyunCheol. "**The Activity of Malicious HTTP-based Botnets using Degree of Periodic Repeatability**" 83-86. IEEE Trans. 2008