บทที่ 1
บทนำ

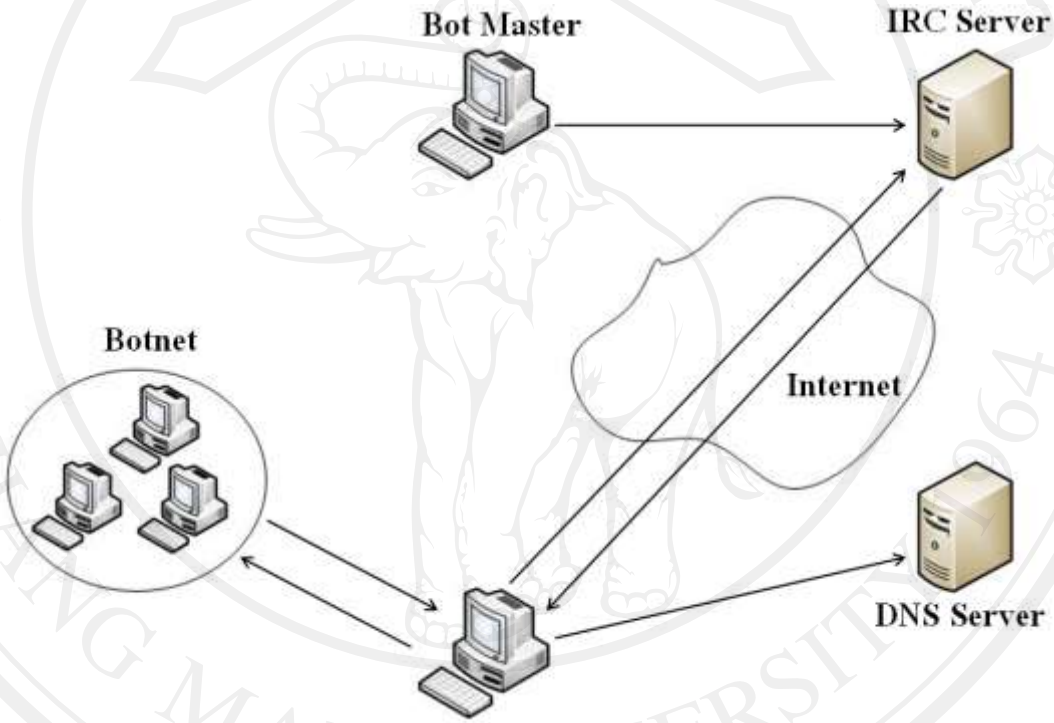
1.1 ที่มาและความสำคัญของปัญหา

บอตเน็ต (BOTNET) หรือ roBOT NETwork เป็นภัยคุกคามต่อผู้ใช้งานอินเทอร์เน็ตรูปแบบใหม่ ซึ่งผู้โจมตี (Hacker) เขียนโปรแกรมบอตเน็ตโดยใช้เทคนิคการโจมตีเครือข่ายอินเทอร์เน็ตด้วยโปรแกรมประสงค์ร้าย (Malware) ที่ซับซ้อนและมีรูปแบบที่หลากหลายกว่าไวรัสคอมพิวเตอร์หรือหนอนอินเทอร์เน็ตทั่วไป บอตเน็ตที่ถูกสร้างขึ้นนี้อาจเป็นเครื่องมือที่ใช้ส่งสแปมเมล์ (Spam Mail) และการโจมตีแบบฟิชชิง (Phishing) ซึ่งเป็นวิธีการสร้างความเสียหายให้กับระบบเครือข่ายอินเทอร์เน็ตได้ นอกจากนี้พบว่ามีการนำบอตเน็ตไปใช้เป็นเครื่องมือประกอบอาชญากรรมทางคอมพิวเตอร์อื่น ๆ อีกด้วย เช่น การขู่กรรโชกทรัพย์ เป็นต้น

บอต (Bot) ซึ่งเป็นหุ่นยนต์ทางซอฟต์แวร์ที่มักจะถูกติดตั้งบนเครื่องลูกข่าย (Client) ให้คอมพิวเตอร์สามารถทำงานตามคำสั่งที่กำหนดไว้ล่วงหน้าได้โดยอัตโนมัติ ตัวอย่างเช่นบอตของห้องสนทนาไออาร์ซี (Internet Relay Chat หรือ IRC) หรือบอตของเกมส์ออนไลน์ต่าง ๆ เป็นต้น ส่วนบอตที่ถูกใช้โดยโปรแกรมประสงค์ร้ายใน “บอตเน็ต” จะแตกต่างจากบอตธรรมดาตรงที่มันเป็นสมาชิกในเครือข่ายของบอตที่ถูกควบคุมจากระยะไกลโดยผู้โจมตี (Hacker) ที่ใช้เครื่องควบคุมบอตเน็ตหลัก (Bot master) เป็นตัวสั่งการ บอตเป็นโปรแกรมประเภทของไวรัสคอมพิวเตอร์ที่มีขนาดเล็ก จึงทำให้ผู้ใช้งานทั่วไปมักไม่สังเกตว่ามีบอตฝังตัวอยู่ในเครื่องคอมพิวเตอร์ที่ใช้อยู่และเครื่องเหล่านี้มักจะถูกควบคุมผ่านทางอินเทอร์เน็ต โดยช่องทางของการควบคุมเครื่องมักจะเป็นการส่งคำสั่งผ่านมาทางระบบห้องสนทนาไออาร์ซี (Internet Relay Chat) นอกจากนี้ไวรัสคอมพิวเตอร์สมัยใหม่บางชนิดจะทำการฝังบอตที่เครื่องของเหยื่อผู้เคราะห์ร้ายและคัดลอกตัวเองไปยังเครื่องคอมพิวเตอร์อื่นๆเพื่อที่จะขยายเครือข่ายของบอตเน็ตให้ใหญ่ขึ้นต่อไป เครื่องคอมพิวเตอร์ของเหยื่อผู้เคราะห์ร้ายจะเปรียบเสมือนเครื่องคอมพิวเตอร์ที่ตายแล้ว และถูกสั่งให้ทำงานโดยผู้อื่นที่ไม่ใช้เจ้าของเครื่อง ดังนั้นจึงถูกเรียกว่าเครื่องซอมบี้คอมพิวเตอร์ (Zombie Machines)

การทำงานของบอตเน็ต (Botnet) จะมีศูนย์กลางควบคุมและสั่งการโดยผู้โจมตี (Hacker) อยู่ที่ใดที่หนึ่งบนอินเทอร์เน็ต กลไกการทำงานของบอตเน็ตถูกออกแบบให้มีการแพร่กระจายตัวเพื่อหาเครื่องใหม่ให้เข้ามาอยู่ในกลุ่ม และมีความสามารถในการแก้ไขโปรแกรมของบอตที่ฝังตัวอยู่บนเครื่องซอมบี้คอมพิวเตอร์ (Zombie Machines) เพื่อเปลี่ยนแปลงรูปแบบการบุกรุก ลักลอบใช้

งานและสั่งการผ่านศูนย์ควบคุม ซึ่งองค์ประกอบหลักของบอตเน็ตได้แก่ เครื่องคอมพิวเตอร์สั่งการระยะไกลของผู้โจมตี เครื่องเซิร์ฟเวอร์ของห้องสนทนาไออาร์ซี (IRC Server) ที่เป็นจุดนัดพบระหว่างกลุ่มของบอตและผู้โจมตีเพื่อรอรับคำสั่ง กลุ่มของดีเอ็นเอสเซิร์ฟเวอร์ (DNS Server) ซึ่งเป็นทางผ่านเพื่อทำให้บอตสามารถหาเครื่องเซิร์ฟเวอร์ของห้องสนทนาไออาร์ซี (IRC Server) เจอได้ นอกจากนี้ยังมีกลุ่มของเครื่องคอมพิวเตอร์ต่าง ๆ บนเครือข่ายอินเทอร์เน็ตที่เป็นเป้าหมายของบอตเน็ตและกลุ่มที่ได้กลายเป็นส่วนหนึ่งของบอตเน็ตไปแล้ว

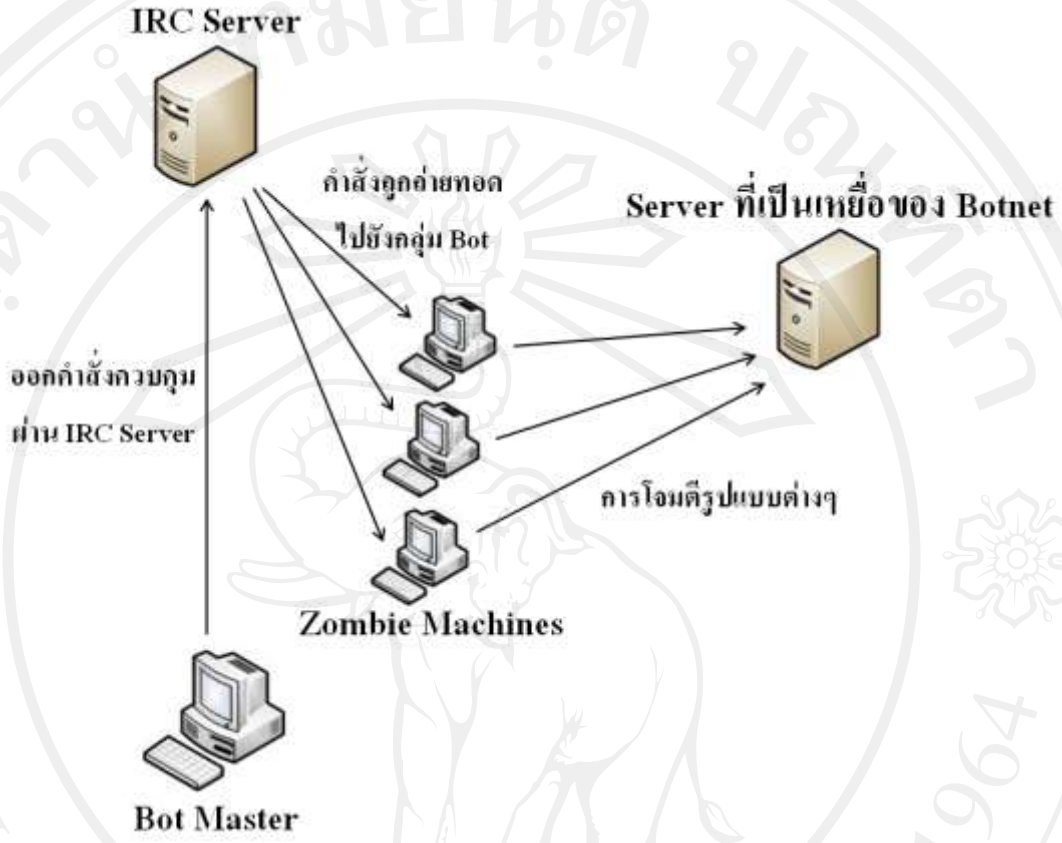


รูปที่ 1.1 กระบวนการทำงานของบอตเน็ต(Botnet) [1]

กระบวนการทำงานของบอตเน็ตมีขั้นตอนดังรูปที่ 1.1 เริ่มจากผู้โจมตี (Hacker) ที่เป็นเจ้าของบอตเน็ตจะสร้างบอตเน็ตด้วยการติดตั้งเครื่องเซิร์ฟเวอร์ของห้องสนทนาไออาร์ซี (IRC Server) เตรียมไว้ ณ ที่ใดที่หนึ่งบนเครือข่ายอินเทอร์เน็ต โดยอาจเป็นเซิร์ฟเวอร์ที่ถูกต้องตามกฎหมายหรือเป็นเครื่องที่ถูกบุกรุกเพื่อนำมาใช้เป็นเครื่องเซิร์ฟเวอร์ของห้องสนทนาไออาร์ซี (IRC Server) ก็ตาม หลังจากนั้นผู้โจมตี (Hacker) ก็จะทำการลงทะเบียนชื่อโดเมนและหมายเลขไอพีของเครื่องเซิร์ฟเวอร์ของห้องสนทนาไออาร์ซี (IRC Server) ไว้กับบริการดีเอ็นเอส (DNS) บนอินเทอร์เน็ต เพื่อให้บอตสามารถค้นหาเซิร์ฟเวอร์ของห้องสนทนาอื่น ๆ ที่ผู้โจมตีผู้นั้นได้ติดตั้ง

บอตไว้ก่อนแล้ว เมื่อโครงสร้างหลักข้างต้นพร้อมแล้ว ผู้โจมตี (Hacker) ก็จะเริ่มทำการโจมตีเครื่องคอมพิวเตอร์เครื่องอื่นๆ บนอินเทอร์เน็ตเพื่อค้นหาเหยื่อสำหรับติดตั้งบอตและทำให้เครื่องคอมพิวเตอร์เหล่านั้นกลายเป็นเครื่องซอมบี้คอมพิวเตอร์ (Zombie Machines)โปรแกรมโจมตีของบอตส่วนใหญ่ที่นิยมมักจะอยู่ในรูปแบบของหนอนอินเทอร์เน็ตหรือโปรแกรมโจมตีคอมพิวเตอร์ที่ฝังตัวอยู่ในโปรแกรมประยุกต์ต่างๆ เช่น การโจมตีโดยอาศัยข้อมูลทางเว็บไซต์ที่ผู้โจมตีอาจจะทำการฝังโปรแกรมประสงค์ร้ายต่างๆ ไว้บนเว็บเซิร์ฟเวอร์ก่อนอยู่แล้ว หรือการซ่อนโปรแกรมประสงค์ร้ายผ่านทางการแชร์ไฟล์ในแบบเพียร์ทูเพียร์ (Peer-to-Peer) เพื่อที่จะลักลอบเข้าไปติดตั้งโปรแกรมโทรจัน (Trojan) สำหรับบอตเน็ตในเครื่องคอมพิวเตอร์ของเหยื่อผู้เคราะห์ร้ายเป็นต้น เมื่อบอตสามารถทำงานบนเครื่องซอมบี้คอมพิวเตอร์ (Zombie Machines)ได้แล้ว บอตก็จะทำการติดต่อกับเครื่องเซิร์ฟเวอร์ดีเอ็นเอส (DNS Server) โดยอัตโนมัติเพื่อค้นหาเครื่องเซิร์ฟเวอร์ห้องสนทนาไออาร์ซี (IRC Server) ที่ผู้โจมตีติดตั้งรอไว้ เมื่อพบเซิร์ฟเวอร์แล้วบอตก็จะทำการล็อกอิน (Login)เข้าไปเพื่อรอรับคำสั่งจากผู้โจมตี (Hacker) เมื่อถึงเวลาที่ต้องการและมีจำนวนบอตมากเพียงพอผู้โจมตีจะล็อกอินผ่านเครื่องควบคุมบอตเน็ตหลัก (Bot master) เข้าสู่ระบบเครื่องเซิร์ฟเวอร์ไออาร์ซี (IRC Server) นั้นเพื่อทำการออกคำสั่งต่างๆ เช่น ให้ทำการโจมตีแบบดีดีโอเอส (Distributed Denial of Service หรือ DDoS) ไปยังเครื่องคอมพิวเตอร์เป้าหมายต่างๆ บนระบบเครือข่ายอินเทอร์เน็ต หรือทำการส่งสแปมเมล์ (Spam Mail) สร้างความรำคาญให้กับผู้ใช้อินเทอร์เน็ต

ผลกระทบของภัยคุกคามจากบอตเน็ต (Botnet) จากความซับซ้อนและรูปแบบของการโจมตีต่าง ๆ ของบอตเน็ตทำให้เกิดผลกระทบในวงกว้างต่อองค์กรและผู้ใช้อินเทอร์เน็ตทั่วไปได้ เนื่องจากปริมาณข้อมูลที่ถูกสร้างขึ้นโดยบอตเน็ตอาจมีปริมาณมหาศาลหากจำนวนบอตมีจำนวนมากนับแสนเครื่อง ผู้ที่ได้รับผลกระทบเป็นอันดับแรกคือ ผู้ให้บริการเครือข่ายอินเทอร์เน็ต (Internet Service Provider หรือ ISP) เนื่องจากปริมาณข้อมูลจำนวนมากอาจทำให้ระบบโครงสร้างหลักของอินเทอร์เน็ตไม่สามารถให้บริการต่อไปได้ เช่นบริการของเครื่องเซิร์ฟเวอร์ดีเอ็นเอส (DNS Server) อุปกรณ์เครือข่ายเราเตอร์ (Router) สวิตช์ (Switch) ต้องทำงานหนักจนเกินไป และสายส่งข้อมูลอาจขัดข้องได้เป็นต้น



รูปที่ 1.2 ผลกระทบของภัยคุกคามจากบอตเน็ต (Botnet)

สำหรับผู้ใช้งานคอมพิวเตอร์และอินเทอร์เน็ตทั่วไปซึ่งเป็นอาจตกเป็นเหยื่อ โดยถูกใช้เครื่องคอมพิวเตอร์ในการทำบอตเน็ตอาจถูกขโมยข้อมูลส่วนตัวที่สำคัญ เช่น รหัสผ่านและข้อมูลทางการเงินจำพวกบัตรเครดิตหรือหมายเลขบัญชีธนาคาร เป็นต้น เพื่อนำไปขายหรือหาประโยชน์ บอตเน็ตบางประเภทสามารถที่จะขโมยรหัสซีดี (CD keys) ของโปรแกรมต่างๆ ที่อยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลได้ นอกจากนี้บางครั้งเครื่องคอมพิวเตอร์อาจถูกใช้เป็นฐานในการโจมตีระบบเครือข่ายอื่นๆต่อไปอีกด้วย

จากปัญหาภัยคุกคามจากบอตเน็ตที่กล่าวมาข้างต้น ปัจจุบันได้มีงานวิจัยเกี่ยวกับการตรวจจับและการจำแนกบอตเน็ตบนระบบเครือข่ายอินเตอร์มากขึ้น เนื่องจากบอตเน็ตที่พบอยู่บนระบบเครือข่ายอินเทอร์เน็ตมีแนวโน้มทวีความรุนแรงขึ้น อีกทั้งยังมีความหลากหลายและพัฒนารูปแบบการโจมตี ให้สามารถเอาชนะระบบตรวจจับรูปแบบต่างๆได้

1.2 แนวทางการแก้ปัญหา

จากปัญหาการประมวลผลที่นานในกรณีที่ข้อมูลมีปริมาณมาก และยังทำให้ความถูกต้องในการจำแนกบอตเน็ตลดลงของการแบ่งกลุ่ม (Clustering Algorithm) โดยใช้อัลกอริทึมเคมีน (K-means Algorithm) ในโครงร่างวิทยานิพนธ์ฉบับนี้จะนำเสนอวิธีการแก้ปัญหาดังกล่าว โดยจะประยุกต์ใช้แนวคิดของงานวิจัย [8] ในส่วนของขั้นตอนการเลือกคุณลักษณะ (Feature Analysis) ใช้วิธีการ The Payload Signature-Based Classifier โดยพิจารณาจาก 256 ASCII Characters ใน Packet Payload [8,9,10,11] เพื่อศึกษาพฤติกรรมของบอตเน็ต และทำการพร็อตกราฟเพื่อดูความถี่เฉลี่ยของ Botnet Traffic และ Normal Traffic ในส่วนของขั้นตอนการแบ่งกลุ่ม (Clustering Algorithm) โดยจะใช้อัลกอริทึมเคมีนแบบขนานที่ทำการปรับปรุงใหม่ สำหรับประมวลผลในตัวประมวลผลหลายตัว ที่มีรูปแบบการทำงานเป็นการแบ่งงานกันประมวลไปพร้อมๆกัน มาทำการแบ่งกลุ่มข้อมูล Botnet Traffic ออกจากข้อมูล Normal Traffic การทดสอบประสิทธิภาพจะทำการเปรียบเทียบกับขั้นตอนวิธีดังนี้

1.2.1 ขั้นตอนวิธีเคมีนแบบเดิม

1.2.2 ขั้นตอนวิธีเคมีนแบบขนาน โดยใช้จุดศูนย์กลางร่วมกัน [22]

1.2.3 ขั้นตอนวิธีเคมีนแบบขนานที่งานวิจัยนี้นำเสนอ

และขั้นตอนสุดท้ายในการตัดสินใจว่าเป็นข้อมูล Botnet Traffic หรือข้อมูล Normal Traffic ใช้วิธี Standard Deviation Metric for Botnet Decision [8]

ในวิทยานิพนธ์นี้อาศัยการปรับปรุงในบางส่วนของขั้นตอนวิธี [8] คือส่วนของการแบ่งกลุ่ม (Clustering Algorithm) เนื่องจากขั้นตอนนี้เป็นขั้นตอนที่ใช้เวลาในการประมวลผลนานกว่าขั้นตอนอื่นๆ เพื่อปรับปรุงให้ใช้งานได้อย่างมีประสิทธิภาพสำหรับชุดข้อมูลที่มีขนาดใหญ่ โดยใช้เวลาในการประมวลผลที่เร็วกว่าและมีความถูกต้องในการจำแนกที่ดีกว่า เป้าหมายของงานวิจัยนี้คือ คือ

1. เปรียบผลที่ได้กับการจำแนกด้วยเทคนิคอื่นๆตามรูปแบบของชุดข้อมูล (Dataset) เดียวกัน โดยวัดประสิทธิภาพในด้านความความถูกต้องในการจำแนก Botnet Traffic กับ Normal Traffic และประสิทธิภาพทางด้านความเร็วในการประมวลผล
2. การวิเคราะห์ผลการทำลอง จะเปรียบเทียบขั้นตอนที่พัฒนากับขั้นตอนวิธีเคมีนแบบเดิม และขั้นตอนวิธีเคมีนแบบขนานโดยใช้จุดศูนย์กลางร่วม [22] ผลการ

วิเคราะห์พิจารณาจากประสิทธิภาพทางด้านความเร็วในการประมวลผล และความถูกต้องในการจำแนก Botnet Traffic จาก Normal Traffic

1.3 สรุปสาระสำคัญจากเอกสารที่เกี่ยวข้อง

ในงานวิจัย [2,3] ใช้เทคนิคการตรวจสอบบอตเน็ตโดยอาศัยวิธีการจำแนกทราฟฟิกของแอปพลิเคชันบนเครือข่าย (Traffic Application Classification) โดยจะสนใจกลุ่มของบอตเน็ตไอ อาร์ซี (IRC-based Botnets) และทำการจัดกลุ่มบอตเน็ตเป็นไออาร์ซี (IRC) และไม่ใช่ไออาร์ซี (Non-IRC) ออกจากกันโดยใช้คุณสมบัติสถิติการไหล(Statistical Flow Characteristics)ของข้อมูลมาพิจารณา ขั้นตอนสุดท้ายในการตรวจสอบบอตเน็ตใช้พฤติกรรมการสื่อสารระหว่างเครื่องควบคุมบอตเน็ตหลัก (Bot Master) กับเครื่องเซิร์ฟเวอร์ซีแอนด์ซี (C&C Server)

เทคนิคการตรวจสอบบอตเน็ตโดยวิเคราะห์จากสแปมเมล์ (Spam Mail) เช่น [4] เป็นเทคนิคที่วิเคราะห์รูปร่าง (Shapes) และโครงสร้างของสแปมเมล์ (Spam Mail) ที่ได้รับ, [5] ศึกษาวิธีการตรวจสอบบอตเน็ตจากหมายเลขไอพี (IP) สแปมเมล์ (Spam Mail) เป็นต้น

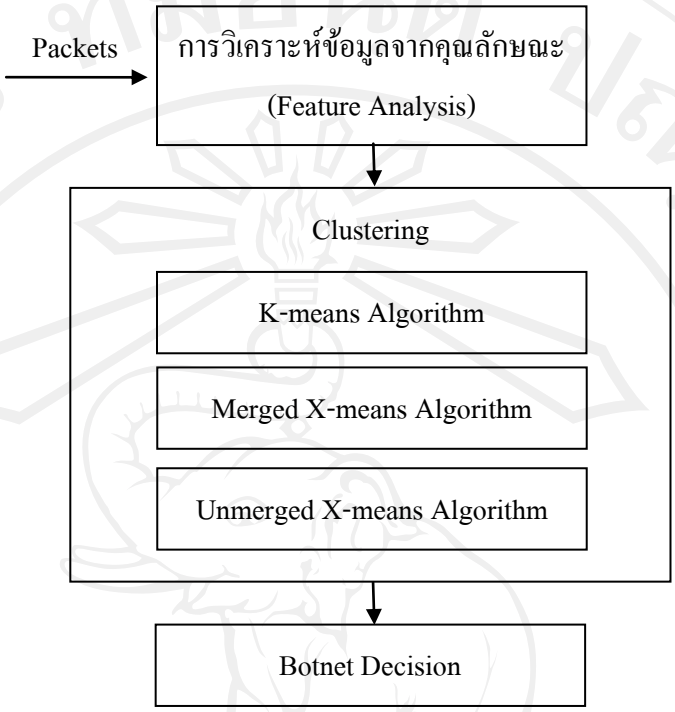
ในงานวิจัย [6] ใช้ทฤษฎีของเบส์ (Bayesian Method) วิเคราะห์ทราฟฟิกบนบริการดีเอนเอส (DNS) เพื่อหาเซตของบอตโฮสต์ (Bot Host) ของเครื่องเซิร์ฟเวอร์ซีแอนด์ซี (C&C Server) ว่ามีโฮสต์ที่มีชื่อเป็นบัญชีดำ (Blacklist) อยู่หรือไม่

งานวิจัยที่ใช้การระบุรูปแบบ (Pattern) [7] การสื่อสารที่แตกต่างกันระหว่างไออาร์ซีที่เป็นบอต (IRC Bot Client) และไออาร์ซีที่ปกติ (Normal IRC Client)

ในงานวิจัย [8] ได้มีการนำเสนอเทคนิคการจำแนกบอตเน็ต (Botnet) โดยมีขั้นตอนโครงสร้างการทำงานแบ่งออกเป็น 3 ขั้นตอนหลักดังนี้ คือ

1. ขั้นตอนการพิจารณาคุณลักษณะ (Feature Analysis) ใช้วิธีการ Payload Signature-Based Classifier เพื่อหาคุณลักษณะ (Feature) สำหรับใช้ต่อไปในขั้นตอนที่ 2
2. ขั้นตอนการแบ่งกลุ่ม (Clustering Algorithm) ใช้สามขั้นตอนวิธีในการเปรียบเทียบการจำแนกบอตเน็ตได้แก่ ขั้นตอนวิธีเคมีน (K-means Algorithm), ขั้นตอนวิธีเมิร์จเอ็กซ์มีน (Merged X-means Algorithm) และขั้นตอนวิธีอันเมิร์จเอ็กซ์มีน (Unmerged X-means Algorithm)
3. ขั้นตอนการตัดสินใจ (Botnet Decision) ใช้วิธี Standard Deviation Metric for Botnet Decision

โดยมีโครงสร้างการทำงานของระบบดังรูปที่ 1.3



รูปที่ 1.3 Detection Botnet Framework [8]

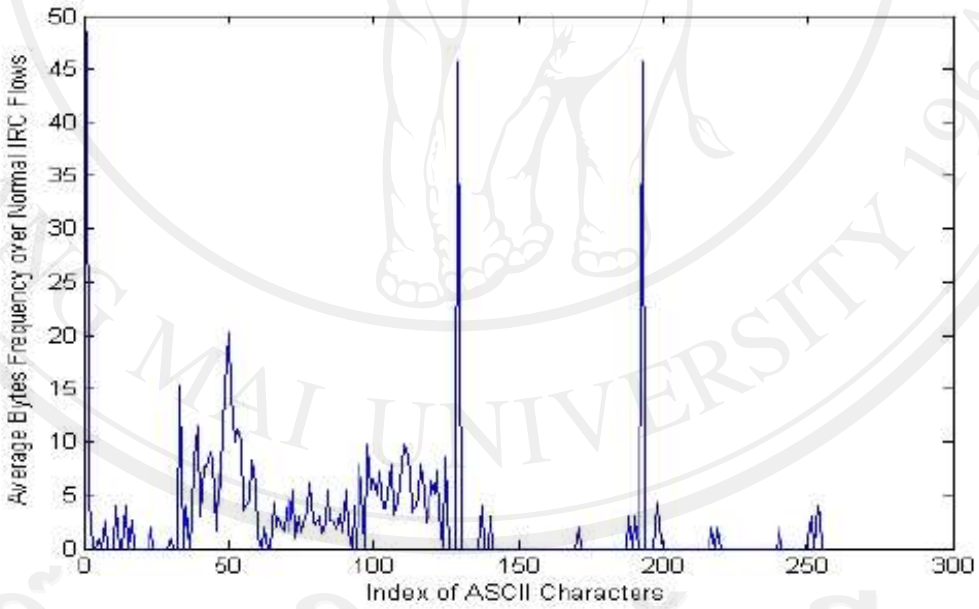
ขั้นตอนแรกของรูปที่ 3 ขั้นตอนการพิจารณาคุณลักษณะ (Feature Analysis) ใช้วิธีการ The Payload Signature-Based Classifier โดยพิจารณาจาก 256 ASCII Characters ใน Packet Payload ซึ่งเป็นวิธีที่นิยมใช้ในปัจจุบัน [8,9,10,11].

การนำ 256 ASCII Characters ใน Packet Payload แล้วจะทำการหา [9] 256-Dimensional n-gram (n=1) byte distribution เวกเตอร์ $< f_1^{t_i}, f_2^{t_i}, \dots, f_{256}^{t_i} >$ เมื่อ $f_{\mathrm{j}}^{t_i}$ คือ ความถี่ของจำนวน j^{th} ASCII ใน Flow Payload ในเวลา t_i โดย (j=1,2,…,256 ; i=1,2,…) หรือเรียกวิธีการนี้ว่า การหา Temporal-frequent Metric ของ Flow Payload. กำหนดให้เมตริกซ์มีขนาด $n \times 256$ เมตริกซ์ และ p^{app} คือแอปพลิเคชั่น แสดงได้ดังสมการ (1.1)

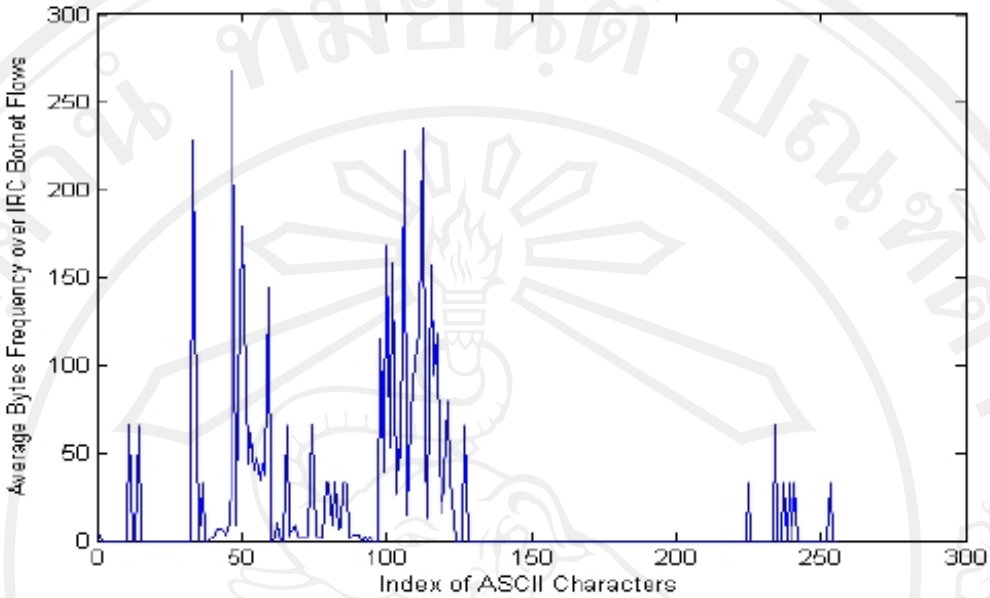
$$p_{n\times 256}^{app} = \begin{bmatrix} f_1^{t_1} & f_2^{t_1} & \cdots & f_{256}^{t_1} \\ f_1^{t_2} & f_2^{t_2} & \vdots & f_{256}^{t_2} \\ \vdots & \vdots & \vdots & \vdots \\ f_1^{t_n} & f_2^{t_n} & \cdots & f_{256}^{t_n} \end{bmatrix} \quad (1.1)$$

สำหรับ Unknown flow ที่ไม่สามารถระบุแอปพลิเคชั่นได้ จึงมีการนำเทคนิคต้นไม้ตัดสินใจ (Decision Tree) [12] มาใช้ในการจำแนกแอปพลิเคชั่น โดยจะนำข้อมูล Temporal-frequent Metric ของ Flow Payload ที่เราทราบ มาทำการ Training ก่อน เมื่อมี Unknown flow เข้ามาก็จะใช้ชุดการเรียนรู้นี้ในการ Train เพื่อให้ลาเบลกับข้อมูลว่าอยู่แอปพลิเคชั่นไหน

ในการหาแอปพลิเคชั่นของ Unknown flow ของขั้นตอนของการหา Feature Analysis ของ Detection Botnet Framework เหตุผลที่ใช้ Temporal-frequent Metric มาจำแนกบอตเน็ตนั้นได้มีการวิจัย [13] ซึ่งทำการศึกษาพฤติกรรมของบอตเน็ตและทำการพร็อตกราฟเพื่อดูความถี่ของทราฟฟิกที่ปกติ และทราฟฟิกที่เป็นบอตเน็ต ในผลการทดลองพบว่าค่าความถี่เฉลี่ยในแต่ละไบต์(Byte)ของตัวอักษรแอสกี (ASCII) ในทราฟฟิกที่เป็นบอตเน็ตมีความถี่ที่สูงกว่าทราฟฟิกที่ปกติอย่างเห็นได้ชัดเจน ทำให้สามารถสรุปได้ว่า ค่าความถี่เฉลี่ยในแต่ละไบต์(Byte)ของตัวอักษรแอสกี (ASCII) สามารถนำมาวิเคราะห์แยกทราฟฟิกที่เป็นบอตเน็ตและทราฟฟิกที่ปกติได้ แสดงในรูปที่ 1.4 และรูปที่ 1.5 ตามลำดับ



รูปที่ 1.4 Average byte frequency over 256 ASCII for Normal IRC flows [13]



รูปที่ 1.5 Average byte frequency over 256 ASCII for Botnet IRC flows [13]

ขั้นตอนที่สองของรูปที่ 3 แสดงขั้นตอนการแบ่งกลุ่ม (Clustering Algorithm) โดยใช้ขั้นตอนวิธีเคมีน (K-means Algorithm), ขั้นตอนวิธีเมิร์จเอ็กซ์มีน (Merged X-means Algorithm) และขั้นตอนวิธีอันเมิร์จเอ็กซ์มีน (Unmerged X-means Algorithm) เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) ออกจากทราฟฟิกที่ปกติ (Normal Traffic) และในขั้นตอนสุดท้ายของรูปที่ 3 คือขั้นตอนในการตัดสินใจว่าเป็น ทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) หรือทราฟฟิกที่ปกติ (Normal Traffic)ใช้วิธี Standard Deviation Metric for Botnet Decision [8] ซึ่งได้ผลการทดลองแสดงดังตารางที่ 1.1, 1.2 และ 1.3 ตามลำดับ

ตารางที่ 1.1 ผลการทดลองการแบ่งกลุ่มของข้อมูลทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) ออกจากทราฟฟิกที่ปกติ (Normal Traffic) โดยใช้ K-means Algorithm [8]

Dataset	Standard Deviation		BPR of Botnet (%)	NPR of Bonet (%)
Model-1	0.199	1.2557	96.72	98.02
Model-2	0.253	0.8946	100.0	85.71

ตารางที่ 1.2 ผลการทดลองการแบ่งกลุ่มของข้อมูลทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) ออกจากทราฟฟิกที่ปกติ (Normal Traffic) โดยใช้ Merged X-means Algorithm [8]

Dataset	Standard Deviation		BPR of Botnet (%)	NPR of Bonet (%)
Model-1	0.3055	0.3004	100.0	0.0
Model-2	0.1165	0.7488	95.0	10.71

ตารางที่ 1.3 ผลการทดลองการแบ่งกลุ่มของข้อมูลทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) ออกจากทราฟฟิกที่ปกติ (Normal Traffic) โดยใช้ Unmerged X-means Algorithm [8]

Dataset	Standard Deviation		BPR of Botnet (%)	NPR of Bonet (%)
Model-1	0.0983	0.1425	95.5	0.0
Model-2	0.019	0.259	81.25	10.71

งานวิจัยการแบ่งกลุ่มข้อมูล [14] พบว่าขั้นตอนวิธีในการจัดกลุ่มของวัตถุต่างๆ โดยใช้ขั้นตอนวิธีเคมีน (K-means Algorithm) และขั้นตอนวิธีเอ็กซ์มีน (X-means Algorithm) การนำไปประยุกต์ใช้มักประสบปัญหาเดียวกัน คือ ในเรื่องเวลา หน่วยความจำ และความถูกต้องในการจัดกลุ่มเนื่องจากข้อมูลมีขนาดใหญ่ขึ้น

งานวิจัยเกี่ยวกับขั้นตอนวิธีเคมีนคลัสเตอร์ริงแบบขนาน [15] ซึ่งแสดงถึงการวิเคราะห์ประสิทธิภาพของอัลกอริทึม พบว่าเวลาที่ใช้สำหรับการประมวลผลลดลงและความสามารถด้านความถูกต้องดีกว่าการจัดกลุ่มเมื่อเทียบกับขั้นตอนวิธีแบบเคมีนคลัสเตอร์ริง

ในงานวิจัย [8] พบว่าขั้นตอนการแบ่งกลุ่ม (Clustering Algorithm) โดยใช้ขั้นตอนวิธีเคมีน (K-means Algorithm), ขั้นตอนวิธีเมิร์จเอ็กซ์มีน (Merged X-means Algorithm) และขั้นตอนวิธีอันเมิร์จเอ็กซ์มีน (Unmerged X-means Algorithm) เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกทราฟฟิกที่เป็นบอตเน็ต (Botnet Traffic) ออกจากทราฟฟิกที่ปกติ (Normal Traffic) ขั้นตอนนี้เป็นขั้นตอนที่ใช้เวลาในการประมวลผลที่นานที่สุดของทั้งสามขั้นตอนที่กล่าวมาข้างต้น จากปัญหาการประมวลผลที่นานในกรณีที่ข้อมูลมีปริมาณมาก และยังทำให้ความถูกต้องในการจำแนกบอตเน็ต

ลดลง จะเป็นการดีกว่าถ้ามีการนำขั้นตอนวิธีที่มีประสิทธิภาพในการแบ่งกลุ่มเมื่อข้อมูลมีจำนวนมากใช้เวลาในการประมวลผลที่เร็วขึ้นและยังทำให้ความถูกต้องในการจำแนกสูงมาใช้

1.4 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและพัฒนาขั้นตอนวิธีใหม่ในการจำแนกบอตเน็ตบนทราฟฟิกของระบบเครือข่ายอินเทอร์เน็ต โดยใช้ขั้นตอนวิธีเคมีนแบบขนาน

1.5 ขอบเขตการทำวิจัย

1.5.1 ข้อมูลบอตเน็ตที่ใช้ในการทดสอบชื่อ q8bot ซึ่งเป็นบอตเน็ตประเภท IRC bot จาก [17]

1.5.2 ข้อมูลนำเข้าของการทดลองคือชุดข้อมูลที่ต้องการทำการจำแนกและข้อมูลนำออกของการทดลองคือข้อมูลที่แยกบอตเน็ตออกจากข้อมูลที่ปกติได้โดยใช้ขั้นตอนวิธีที่พัฒนาขึ้น

1.5.3 โดยเปรียบเทียบผลที่ได้ กับการจำแนกด้วยเทคนิคอื่นๆที่มีการจำแนกรูปแบบของชุดข้อมูล (Dataset) แบบเดียวกัน

1.5.4 การวิเคราะห์ผลการทำลอง จะเปรียบเทียบขั้นตอนที่พัฒนากับขั้นตอนวิธีที่ใช้ K-means Algorithm, ขั้นตอนวิธีเคมีนแบบขนานโดยใช้จุดศูนย์กลางร่วม และขั้นตอนวิธีเคมีนแบบขนานที่นำเสนอ ผลการวิเคราะห์พิจารณาจาก ความถูกต้องในการจำแนก Botnet Traffic จาก Normal Traffic และ ความเร็วในการประมวลผล

1.6 ประโยชน์ที่ได้รับจากการศึกษาเชิงทฤษฎีและเชิงประยุกต์

ได้ขั้นตอนวิธีใหม่ที่พัฒนาขึ้นมาโดยใช้อัลกอริทึมเคมีนแบบขนาน ในการจำแนกบอตเน็ตบนทราฟฟิกของระบบเครือข่ายอินเทอร์เน็ต

1.7 ระเบียบวิธีวิจัย

1.7.1 ศึกษาเอกสารข้อมูลและงานวิจัยที่เกี่ยวข้อง

1.7.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับโครงสร้างการทำงานของบอตเน็ต

1.7.3 ศึกษาทฤษฎีที่เกี่ยวข้องกับการระบุทราฟฟิกแบบ Payload Signature-Based Classifier

1.7.4 ศึกษาทฤษฎีที่เกี่ยวข้องกับการแบ่งกลุ่มข้อมูลโดยใช้อัลกอริทึมเคมีนแบบขนาน

1.7.5 ออกแบบและพัฒนาแบบจำลอง

1.7.6 ออกแบบและสร้างขั้นตอนวิธีการแบ่งกลุ่มข้อมูลโดยใช้ขั้นตอนวิธีเคมีนแบบขนาน

1.7.7 พัฒนาการแบ่งกลุ่มข้อมูลโดยใช้อัลกอริทึมเคมีนแบบขนานและทำการเปรียบเทียบกับวิธีการอื่นๆ

1.7.8 วิเคราะห์ผลที่ได้จากการทดลอง

1.7.9 จัดทำวิทยานิพนธ์ฉบับสมบูรณ์