บทความวิจัย (Research Article)

การตรวจสอบอุปกรณ์ที่ติดบอตเน็ตโดยวิเคราะห์ข้อมูลจราจรดีเอ็นเอส

DeBiM: Detecting Bot-infected Machines by DNS Traffic Analysis

กรวิชญ์ ชัยกังวาฬ*, ณัฐพัชร์ วัฒนชาญสิทธิ์, สุรศักดิ์ สงวนพงษ์, พีรวัฒน์ วัฒนพงศ์ และ พิรุณ ศรีสว่าง

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์
* ผู้รับผิดชอบบทความ
Korrawit.cha@ku.th

Received: 15 Jan 2021
Revised: 25 May 2021
Accepted: 28 May 2021

บทคัดย่อ

บอตเน็ตเป็นหนึ่งในภัยคุกคามในอินเทอร์เน็ตที่ผู้ไม่ประสงค์ดีสร้างขึ้นเพื่อใช้ก่ออาชญากรรมไซเบอร์ เช่น การทำให้ระบบปฏิเสธบริการ (DoS) การขโมยข้อมูลสำคัญและการกระจายสแปม เป็นต้น เนื่องจากบอตเน็ตมีพัฒนาการอย่างต่อเนื่องและตรวจจับได้ยาก จากเทคนิคการซ่อนเชิร์ฟเวอร์สั่งการและควบคุม โดยบอตต้องอาศัยดีเอ็นเอสในการค้นหาไอพีแอดเดรสของเชิร์ฟเวอร์สั่งการและควบคุม งานวิจัยฉบับนี้นำเสนอวิธีการค้นหาอุปกรณ์ในระบบที่ติดบอตเน็ตโดยใช้ข้อมูลการติดต่อสื่อสารระหว่างอุปกรณ์กับระบบดีเอ็นเอส เพื่อจำแนกประเภทของชื่อโดเมนที่สร้างจากขั้นตอนวิธีการสร้างโดเมนแบบสุ่ม (DGAs) ที่ใช้ซ่อนเชิร์ฟเวอร์สั่งการและควบคุมกับชื่อโดเมนที่มนุษย์เป็นผู้สร้างโดยอาศัยเทคนิคบัญชีขาว (Whitelist) ร่วมกับหลักประมวลผลภาษาธรรมชาติ

คำสำคัญ: การสร้างโดเมนแบบสุ่ม การประมวลผลภาษาธรรมชาติ การตรวจจับบอตเน็ต การตรวจจับความผิดปกติ

Abstract

The Botnet is one of the major threats in Internet which attackers use it to make cybercrimes such as DoS attacks, stealing sensitive data, or spam spreading. The Botnet constantly evolves itself thus making it more difficult to detect. Command-and-Control (C&C) servers are basic machines which contain bot script and are placed in Internet and waiting to be connected by bot-infected machine via domain name system (DNS) query. Infected machines typically make a likely random but systematic DNS query to connect to C&C server. This Domain Generation Algorithms (DGAs) technique makes a difficulty for monitoring system to detect the C&C server. In this paper, we present a methodology for detecting bot-infected machine using DNS traffic log. Our technique can differentiate legal domains from DGAs domain from DNS log by applying the combination of whitelist domain and Natural Language Processing (NLP) technique.

Keywords: Domain Generation Algorithm (DGAs), Natural Language Processing (NLP), Botnet Detection, Anomaly Detection

1. บทนำ

บอตเน็ตคือชื่อของกลุ่มคอมพิวเตอร์ที่มีมัลแวร์ ฝังตัวอยู่แต่ละคอมพิวเตอร์จะถูกเรียกว่าบอต (Bot) หรือซอมบี้ (Zombie) มัลแวร์ดังกล่าวเป็นซอฟต์แวร์ที่ใช้ทรัพยากรคอมพิวเตอร์เป้าหมายเพื่อให้ผู้ควบคุมมัลแวร์หรือบอตมาสเตอร์ (Bot Master) ใช้ก่ออาชญากรรมคอมพิวเตอร์ เช่น การทำให้ระบบปฏิเสธบริการ (DoS), การขโมยข้อมูล, หรือการกระจายสแปม เป็นต้น ซอฟต์แวร์ประเภทนี้จะคอยรับคำสั่งจากผู้สั่งการผ่านการควบคุมระยะไกลโดยการวางชุดคำสั่งไว้ในเครื่องและรอขั้นตอนการต่อเชื่อมเพื่อรับคำสั่งปฏิบัติงาน

การรับส่งคำสั่งระยะไกลสามารถทำได้ด้วยการสร้างเซิร์ฟเวอร์ขึ้นเพื่อให้บอตสร้างการเชื่อมต่อเพื่อร้องขอคำสั่งผ่านเครือข่าย ในยุคแรกเริ่มบอตเน็ตมักฝังค่าไอพีแอดเดรสของเซิร์ฟเวอร์สั่งการและควบคุม (C2 Server) ลงไปแบบตายตัว (Hard Code) แต่วิธีนี้บอตเน็ตจะถูกตรวจสอบและปิดกั้นได้ง่ายและไม่สามารถรับคำสั่งใหม่ได้ บอตเน็ตสมัยใหม่จึงพัฒนาวิธีแบบไดนามิกขึ้นโดยใช้ชื่อโดเมนเข้ามาช่วย การใช้ชื่อโดเมนทำให้สามารถเปลี่ยนไอพีแอดเดรสของเครื่องได้ แต่ทว่าการใช้ชื่อโดเมนก็ตรวจสอบและถูกปิดกั้นได้เช่นเดียวกัน บอตเน็ตจึงใช้วิธีสร้างโดเมนแบบสุ่ม [1] ที่ไม่ตายตัว เพื่อลดความสามารถในการตรวจจับชื่อโดเมน

ชื่อโดเมนที่สุ่มจากขั้นตอนการสร้างชื่อโดเมนแบบสุ่มหรือ Algorithmically Generated Domains (AGDs) จะเป็นชื่อโดเมนที่ผสมอักขระที่ไม่มีความหมายเชิงภาษา และไม่ตรงกับชื่อโดเมนปกติที่ใช้ในอินเทอร์เน็ต [2] ดังตัวอย่างในตารางที่ 1 ขั้นตอนวิธีสร้างชื่อโดเมนแบบสุ่มทำให้สร้าง AGDs ได้เป็นจำนวนมาก ทั้งบอตและเซิร์ฟเวอร์สั่งการและควบคุมมีหลักการสุ่มกลุ่มชื่อโดเมนเดียวกันเพื่อใช้จับคู่ไอพีแอดเดรสของบอตมาสเตอร์ นอกจากนี้บอตมาสเตอร์สามารถเปลี่ยนคำสั่งใหม่ โดยตั้งเซิร์ฟเวอร์สั่งการและควบคุมแล้วเชื่อมชื่อโดเมนหนึ่งในเซตกับไอพีแอดเดรสของเซิร์ฟเวอร์ชั่วขณะหนึ่งจนกระทั่งมั่นใจว่าบอตได้รับคำสั่งครบถ้วนแล้วก็จะตัดการเชื่อมต่อระหว่างชื่อโดเมนกับไอพีแอดเดรสนั้นเพื่อหลบซ่อนที่ตั้งของเซิร์ฟเวอร์สั่งการและควบคุม วิธีดังกล่าวจะทำให้การปิดกั้นโดยใช้บัญชีดำ (Blacklist) ไม่ว่าจะเป็นบัญชีดำสำหรับไอพีแอดเดรสหรือ บัญชีดำสำหรับชื่อโดเมนยากที่จะประสบผลสำเร็จ

เพื่อแก้ปัญหาข้างต้น จึงมีงานวิจัยที่นำเสนอแนวคิดการตรวจจับชื่อโดเมนแบบสุ่ม เพื่อตรวจหาอุปกรณ์ที่ติดบอตเน็ตแบบอัตโนมัติถึงแม้ว่าชื่อโดเมนแบบสุ่มกับชื่อโดเมนที่มนุษย์เป็นผู้สร้างจะสามารถจำแนกได้อย่างง่ายดายโดยใช้หลักการอ่าน แต่การจำแนกประเภทชื่อโดเมนเหล่านี้แบบอัตโนมัติโดยเทียบกับฐานข้อมูลชื่อโดเมนซึ่งมีอยู่จำนวนมากโดยตรงก็เป็นเรื่องท้าทายและใช้ทรัพยากรสิ้นเปลืองอย่างมาก งานวิจัยนี้จึงได้พัฒนาระบบอัตโนมัติสำหรับตรวจหาอุปกรณ์ที่ติดบอตเน็ตจากข้อมูลจราจรดีเอ็นเอสเพื่อแก้ไขปัญหาดังกล่าว

ตารางที่ 1 ตัวอย่างชื่อโดเมนจากการสร้างชื่อโดเมนแบบสุ่ม

ลำดับที่	ตระกูล	ชื่อโดเมน
1	nymaim	eeoahxj.info
2	locky	xvcgythoxtdjxp.pm
3	conficker	irmgrdclr.org
4	qadars	0darchu3gd2z.net
5	necurs	clruhgxy.cc
6	cryptolocker	jmyhfaxfsyeg.info

2. งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่วิเคราะห์ข้อมูลจราจรดีเอ็นเอสเพื่อค้นหาอุปกรณ์ที่ติดบอตเน็ตมักใช้วิธีตรวจจับจากบันทึกเครือข่าย (Network Log) ระบบวิเคราะห์บอตเน็ตจะใช้ข้อมูลจราจรเครือข่ายขณะนั้นหรือบันทึกข้อมูลก่อนหน้านั้นมาประมวลผล การตรวจจับมีทั้งวิธีจำแนกประเภทโดยใช้ฐานความรู้ (Knowledge Base), การเรียนรู้ของเครื่อง (Machine Learning), และการผสมระหว่างฐานความรู้และการเรียนรู้ของเครื่อง เป็นต้น

แนวคิดการใช้ฐานความรู้เพื่อสร้างตัวจำแนกประเภทจะใช้เฉพาะข้อมูลส่วนที่รู้นำมาดัดแปลงให้สามารถใช้งานได้ เช่น งานวิจัยของ Choi และคณะ [3] ใช้ข้อมูลแบบป้ายกำกับเพื่อทำบัญชีขาวและบัญชีดำโดยตรง ในงานวิจัยของ Alieyan และคณะ [4] นำข้อมูลมาวิเคราะห์องค์ประกอบหลัก และ ใช้อัตราส่วนการรับข้อมูล (Information Gain Ratio) มาหาค่าร่วมกัน การวิเคราะห์ AGDs มีอยู่ในงานวิจัยอย่างหลากหลาย เช่น Yadav และคณะ [5] นำส่วนหนึ่งของชื่อโดเมนมาคำนวนหาความน่าจะเป็นว่าใกล้เคียงกับ AGDs เพียงใด งานของ Wang และคณะ [6]

ตรวจสอบรูปแบบของการร้องขอ AGDs กับฐานข้อมูลที่เก็บรูปแบบไว้แล้ว และ Grill และคณะ [7] ตรวจหาบอตโดยประเมินจากอัตราส่วนจำนวนครั้งในการร้องขอชื่อโดเมนของแต่ละเครื่องเทียบกับจำนวนไอพีแอดเดรสทั้งหมดที่เครื่องนั้นร้องขอ

เทคนิคการเรียนรู้ของเพื่อสกัดลักษณะเฉพาะ (Features) เป็นอีกแนวทางหนึ่งในการค้นหาบอต เช่น การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ตัวอย่างเช่น การใช้ข้อมูลค่าเวลา TTL (Time-To-Live), อัตราส่วนของการสืบค้นข้อมูลดีเอ็นเอส (DNS Querying Ratio) และข้อมูลอื่น ๆ มาจัดกลุ่มความคล้ายคลึงกันของพฤติกรรม [8] การนำบันทึกสืบค้นข้อมูลชื่อโดเมนที่ไม่มีอยู่จริง หรือ Non-existent Domain Name (NXDOMAIN) มาจัดกลุ่มข้อมูลที่มีความคล้ายคลึงกัน [9] เป็นต้น

ส่วนเทคนิคการเรียนรู้แบบมีผู้สอน (Supervised Learning) มีตัวอย่างงานวิจัยเช่น วิธีการนำลักษณะเฉพาะของการร้องขอดีเอ็นเอสมาสร้างตัวจำแนกประเภทด้วยโมเดลต้นไม้ตัดสินใจ (Decision Tree) [10-11] วิธีแรนดอมฟอเรสต์ (Random Forest) กับข้อมูลชื่อโดเมน [12] การใช้มัลติเลเยอร์เพอร์เซปตรอน (Multilayer Perceptron) ควบคู่กับการทำไบแกรม [13] รวมทั้งการเรียนรู้แบบไฮบริดที่ใช้ฐานความรู้ส่วนของไอพีแอดเดรสที่อุปกรณ์ที่ติดบอตติดต่อไปมาผนวกกับการเรียนรู้ของเครื่องเพื่อหาความน่าจะเป็นที่จะเป็นบอต [14-15]

บทความนี้เสนอแนวคิดการตรวจจับในอีกรูปแบบหนึ่งซึ่งเน้นการเสนอขั้นตอนวิธีที่ออกแบบสำหรับนำไปใช้ในสภาพแวดล้อมจริงได้ โดยใช้หลักประมวลผลภาษาธรรมชาติเพื่อช่วยตรวจสอบว่า โดเมนเป้าหมายมีความน่าจะเป็นที่อยู่ในข่ายบอตเน็ตเพียงใด และใช้หลักเอ็นแกรม (N-Gram) นับอักขระที่ N=4 ซึ่งให้ผลลัพธ์ที่มีความแม่นยำสูง สามารถประมวลผลได้อย่างรวดเร็วจึงเหมาะกับการประยุกต์ใช้แบบเรียลไทม์ รูปแบบการทำงานจะอาศัยข้อมูลการจราจรดีเอ็นเอสจากดีเอ็นเอสเซิร์ฟเวอร์กลาง เพื่อนำบันทึกการร้องขอชื่อโดเมนมาประมวลผลโดยอัตโนมัติ

3. การออกแบบระบบ

ภาพรวมของระบบการตรวจสอบอุปกรณ์ที่ติดบอตเน็ตจากข้อมูลจราจรดีเอ็นเอสแบ่งออกเป็นสามโมดูลหลัก ได้แก่ โมดูลการรวบรวมข้อมูล, โมดูลการประมวลผล และ โมดูลการแสดงผล ดังรูปที่ 2 โดยแต่ละโมดูลมีแนวคิดและรายละเอียดการทำงานดังนี้

3.1 โมดูลการรวบรวมข้อมูล

ส่วนนี้ทำหน้าที่รวบรวมข้อมูลจราจรดีเอ็นเอสโดยไม่จำเป็นต้องเข้าเก็บข้อมูลถึงอุปกรณ์รายเครื่อง แต่ใช้ข้อมูลดีเอ็นเอสแบบพาสซีฟเพื่อนำบันทึกส่งผ่านเครือข่ายเพื่อประมวลผลต่อไป

บันทึกที่จะนำเข้าสู่โมดูลรวบรวมข้อมูลจะประกอบไปด้วย (1) วัน, (2) เวลา, (3) ชนิดข้อมูล, (4) ไอพีแอดเดรสต้นทาง, (5) พอร์ตต้นทาง, (6) ทิศทางข้อมูล, (7) ไอพีแอดเดรสปลายทาง, (8) พอร์ตปลายทาง, (9) ชนิดโปรโตคอล (10) ขนาดข้อความ (11) ชื่อโดเมน (12) คลาสข้อมูล และ(13) ชนิดระเบียนดีเอนเอส ดังตัวอย่างในรูปที่ 1

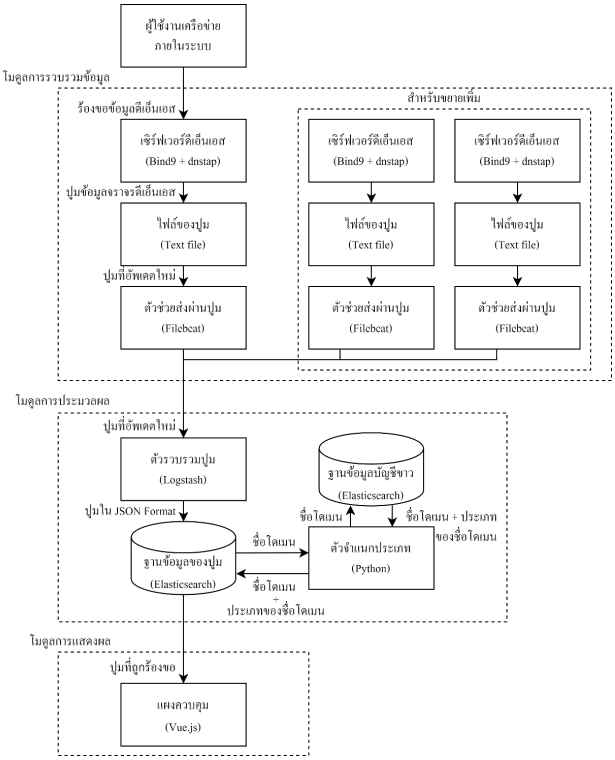
3.2 โมดูลการประมวลผล

ส่วนนี้ทำหน้าที่นำข้อมูลจากส่วนการรวบรวมข้อมูลมาประมวลผล โดยทำงานแบบใกล้เวลาจริง (Near Real-Time) การวิเคราะห์ในขั้นตอนแรกจะคัดกรองชื่อโดเมนปกติด้วยกฎการเทียบชื่อกับ Alexa Top Sites [16] หลังจากนั้นจะนำข้อมูลที่ยังไม่สามารถจำแนกประเภทได้มาคำนวณหาความน่าจะเป็นที่จะเป็นภาษามนุษย์ด้วยวิธีเอ็นแกรม โดยใช้ข้อมูลจาก Alexa Top Sites จำนวน 747,535 ข้อมูล ตามตัวอย่างในตารางที่ 2 มาใช้สร้างโมเดลเอ็นแกรมเมื่อ N=2, 3, ..., 6 และนำแต่ละแกรมมาหาคะแนนเอฟวัน (F1-Score) สูงที่สุดโดยทดสอบด้วยข้อมูลที่ไม่

```
 (1)         (2)          (3) (4):(5)          (6) (7):(8)     (9)(10) (11)/(12)/(13)
20-Feb-2020 11:06:38.203 RQ 192.168.4.5:57462 -> 8.8.8.8:53 UDP 53b www.qncxhawpotnsd.com/IN/A
20-Feb-2020 11:06:38.795 RR 192.168.4.5:57462 <- 8.8.8.8:53 UDP 41b www.qncxhawpotnsd.com/IN/A
20-Feb-2020 11:06:38.795 RQ 192.168.4.5:47512 -> 158.108.0.2:53 UDP 53b www.ku.ac.th/IN/A
20-Feb-2020 11:06:39.152 RR 192.168.4.5:47512 <- 158.108.0.2:53 UDP 586b www.ku.ac.th/IN/A
20-Feb-2020 11:06:39.152 RQ 192.168.4.5:58123 -> 8.8.8.8:53 UDP 58b www.asdqwjr.com/IN/A
20-Feb-2020 11:06:38.715 RQ 192.168.4.5:56123 -> 8.8.8.8:53 UDP 58b www.asdqfqwed.com/IN/A
```

รูปที่ 1 ตัวอย่างบันทึกจากดีเอ็นเอส

ทราบคำตอบ ต่อจากนั้นนำเกณฑ์ที่ได้มาทดสอบด้วยข้อมูลทดสอบเพื่อหาคะแนนเอฟวันที่สูงที่สุด เมื่อชื่อโดเมนถูกแปลงเป็นความน่าจะเป็นแล้วจะนำมาจำแนกประเภทของชื่อโดเมนด้วยเกณฑ์ และขั้นตอนสุดท้ายของการประมวลผลจะติดป้ายกำกับ (Label) ประเภทของชื่อโดเมนของข้อมูลแต่ละข้อมูลที่อยู่ในฐานข้อมูลเพื่อนำไปใช้ในส่วนของการแสดงผลต่อไป



รูปที่ 2 โครงสร้างระบบการตรวจสอบอุปกรณ์ที่ติดบอตเน็ตจากข้อมูลจราจรดีเอ็นเอสแบบ DeBiM

3.3 โมดูลการแสดงผล

ส่วนนี้ทำหน้าที่แสดงผลข้อมูล โดยประกอบด้วยแผนภูมิต่าง ๆ เพื่ออำนวยความสะดวกให้ผู้ดูแลระบบสามารถตรวจสอบและวิเคราะห์ระบบ

โมดูลการรวบรวมข้อมูลถูกพัฒนาขึ้นในเครื่องที่มีหน่วยประมวลผลกลาง (CPU) 1 แกนหน่วยความจำหลักขนาด 2 กิกะไบต์ทำงานภายใต้ระบบปฏิบัติการอูบุนตูเซิร์ฟเวอร์ 18.04 (Ubuntu server 18.04) ทำงานด้วย Bind9 ผ่านโปรแกรมดีเอนเอสแท็บ (dnstap) และใช้ไฟล์บีท (filebeat) เป็นตัวส่งข้อมูลไปยังตัวรวบรวมข้อมูลดีเอ็นเอสของโมดูลการประมวลผล

โมดูลการประมวลผลถูกพัฒนาขึ้นในเครื่องที่มีหน่วยประมวลผลกลาง 4 แกน หน่วยความจำหลัก ขนาด 4 กิกะไบต์ ภายใต้ระบบปฏิบัติการอูบุนตูเซิร์ฟเวอร์ 18.04 และใช้ล็อกสแตช (logstash) ทำหน้าที่รวมรวมและแปลงข้อมูลดีเอ็นเอสเพื่อจัดเก็บด้วยอีลาสติกเสิร์ช (elasticsearch) ตัวจำแนกประเภทของ DeBiM ที่พัฒนาขึ้นด้วยไพธอน 3.6

โมดูลการแสดงผลพัฒนาขึ้นโดยใช้ภาษาจาวาสคริปต์ (JavaScript) และเฟรมเวิร์ควิวเจเอส (VueJs)

ตารางที่ 2 ตัวอย่างข้อมูล 10 อันดับแรกจาก Alexa Top Sites

ลำดับที่	ชื่อโดเมน
1	google.com
2	youtube.com
3	tmall.com
4	qq.com
5	baidu.com
6	sohu.com
7	facebook.com
8	360.cn
9	taobao.com
10	jd.com

ตารางที่ 3 ผลการทดสอบข้อมูลที่มีอัตราส่วนโดเมนที่สร้างจากขั้นตอนวิธีสร้างชื่อโดเมนแบบสุ่มในสัดส่วน 30%

โมเดล	Accuracy	Precision	Recall	F1-Score
ไบแกรม	0.882	0.794	0.916	0.850
ไตรแกรม	0.903	0.845	0.898	0.872
โฟร์แกรม	0.918	0.853	0.934	0.892
ไฟว์แกรม	0.907	0.972	0.764	0.856
ซิกแกรม	0.858	0.984	0.617	0.758

4. การทดสอบระบบ

การทดสอบระบบจะแบ่งเป็นสองส่วน โดยส่วนแรกจะเป็นการวัดผลในรูปแบบของความแม่นยำซึ่งจะวัดโดยหลักของ Binary Classification [17] จากค่าดังนี้

- True Positive (TP) คือจำนวนสิ่งที่ทำนายว่าเป็นส่วนที่สนใจในข้อมูลที่เป็นส่วนที่สนใจจริง

- False Positive (FP) คือจำนวนสิ่งที่ทำนายว่าเป็นส่วนที่สนใจในข้อมูลเป็นส่วนที่ไม่ได้สนใจ
- True Negative (TN) คือจำนวนสิ่งที่ทำนายว่าเป็นส่วนที่ไม่ได้สนใจในข้อมูลเป็นส่วนที่สนใจ
- False Negative (FN) คือจำนวนสิ่งที่ทำนายว่าเป็นส่วนที่ไม่ได้สนใจในข้อมูลเป็นส่วนที่ไม่ได้สนใจจริง

ความแม่นยำที่สนใจในส่วนที่ทำนาย (Precision) ตามสมการที่ (1) เป็นค่าวัดความสามารถว่าโมเดลมีสัดส่วนของข้อมูลที่ทำนายว่าเป็นส่วนที่สนใจจริงกับข้อมูลทำนายว่าเป็นส่วนที่สนใจทั้งหมดเป็นเท่าใด

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (1)$$

ความแม่นยำที่สนใจในส่วนของความเป็นจริง (Recall) ตามสมการที่ (2) เป็นค่าวัดความสามารถว่าโมเดลมีสัดส่วนของข้อมูลที่ทำนายว่าเป็นส่วนที่สนใจจริงกับข้อมูลที่เป็นส่วนที่สนใจจริงเป็นเท่าใด

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (2)$$

ความแม่นยำโดยรวม (Accuracy) ตามสมการที่ (3) เป็นค่าวัดว่าโมเดลสามารถทำนายได้ถูกต้องมากเพียงใดเมื่อเทียบกับสิ่งที่ทำนายมาทั้งหมด ซึ่งจะใช้งานได้ดีในกรณีที่ข้อมูลสมดุลหรือเกือบสมดุล

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \qquad (3)$$

ค่าคะแนนเอฟวัน (F1-Score) ตามสมการที่ (4) แสดงค่าเฉลี่ยแบบฮาร์โมนิกระหว่าง Precision และ Recall ซึ่งค่าที่ได้แล้วจะมีประโยชน์คล้ายกับตัวความแม่นยำโดยรวม แต่จะดีกว่าในเมื่อข้อมูลไม่สมดุลกัน

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (4)$$

ในส่วนที่สองจะเป็นการวัดประสิทธิภาพโดยการวัดเวลาที่กลุ่มข้อมูล (Packets) ใช้ในการอยู่ในระบบ ด้วยวิธีการระบุเวลาในแต่ละหน่วยย่อยของระบบเพื่อนำมาคำนวณเวลาที่ใช้ทั้งหมดในระบบ ได้แก่ เวลาที่โมดูลการรวบรวมข้อมูลและบันทึกข้อมูลลงในฐานข้อมูล เวลาที่ใช้ในโมดูลการประมวลผล และเวลาในการดึงข้อมูลมาแสดงผลที่โมดูลการแสดงผล

ตารางที่ 4 ผลการทดสอบข้อมูลที่มีอัตราส่วนโดเมนที่สร้างจากขั้นตอนวิธีสร้างชื่อโดเมนแบบสุ่มในสัดส่วน 40%

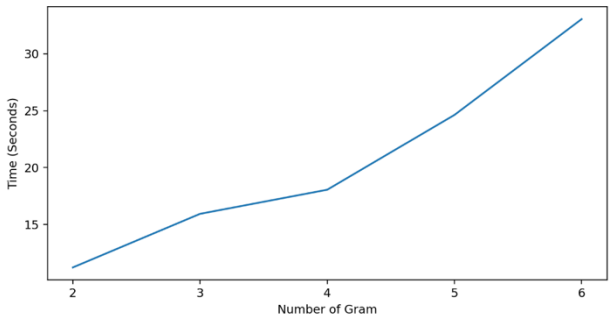
โมเดล	Accuracy	Precision	Recall	F1-Score
ไบแกรม	0.884	0.795	0.917	0.852
ไตรแกรม	0.904	0.847	0.897	0.871
โฟร์แกรม	0.926	0.865	0.946	0.904
ไฟวฟ์แกรม	0.915	0.974	0.791	0.873
ซิกแกรม	0.865	0.989	0.644	0.779

ตารางที่ 5 ผลการทดสอบข้อมูลที่มีอัตราส่วนโดเมนที่สร้างจากขั้นตอนวิธีสร้างชื่อโดเมนแบบสุ่มในสัดส่วน 50%

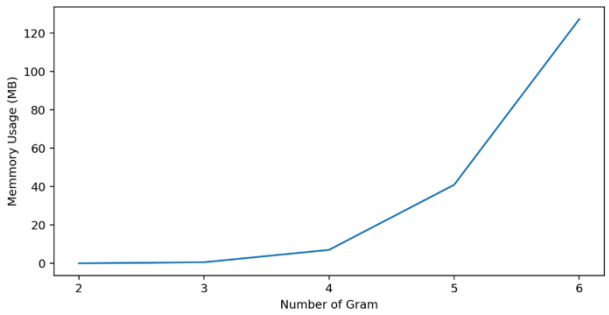
โมเดล	Accuracy	Precision	Recall	F1-Score
ไบแกรม	0.886	0.797	0.917	0.853
ไตรแกรม	0.904	0.848	0.896	0.871
โฟร์แกรม	0.924	0.865	0.942	0.902
ไฟวฟ์แกรม	0.916	0.978	0.792	0.875
ซิกแกรม	0.858	0.988	0.624	0.765

ตารางที่ 6 จำนวนแกรมที่ใช้กับเวลาที่ใช้ในจำแนกประเภทชื่อโดเมนต่อข้อมูลจำนวน 100,000 ข้อมูล

โมเดล	เวลาที่ใช้ (วินาที)
ไบแกรม	6.557
ไตรแกรม	6.616
โฟร์แกรม	3.099
ไฟวฟ์แกรม	3.984
ซิกแกรม	3.642



รูปที่ 3 จำนวนแกรมและเวลาที่ใช้ในการเรียนรู้



รูปที่ 4 จำนวนแกรมและขนาดของโมเดล

5. ผลการทดสอบระบบ

ผลของการพัฒนางานวิจัยฉบับนี้จะแบ่งออกเป็นสองส่วนหลักคือ ผลของการออกแบบโมดูลการประมวลผล และผลของระบบ

5.1 ผลของการออกแบบโมดูลการประมวลผล

การออกแบบโมดูลประมวลผลมีการทดสอบสองรูปแบบคือ ทดสอบความแม่นยำ และทดสอบความสามารถในการประมวลผล โดยมีรายละเอียดดังนี้

5.1.1. ผลการทดสอบระบบในรูปแบบของความแม่นยำ

การวัดประสิทธิภาพในรูปแบบของความแม่นยำจะทำการทดสอบกับกลุ่มข้อมูล 1000 ตัวอย่างโดยมีอัตราส่วนโดเมนที่สร้างจากขั้นตอนวิธีสร้างชื่อโดเมนแบบสุ่มในสัดส่วน 30%, 40%, 50% ตามลำดับ โดยมีผลลัพธ์แสดงดังตารางที่ 3, 4 และ 5

5.1.2. ผลการทดสอบระบบในรูปแบบของประสิทธิภาพ

เวลาที่ใช้สร้างโมเดลจะเพิ่มขึ้นแบบเส้นตรงตามจำนวนแกรมที่ใช้ ดังรูปที่ 3 เนื่องจากโมเดลเก็บอยู่ในรูปดิกชันนารี ซึ่งดิกชันนารีในภาษาไพธอนมีความซับซ้อนทางเวลา (Time Complexity) อยู่ที่ O(n) ส่วนขนาดของโมเดลที่ใช้จะมีขนาดเพิ่มขึ้นแบบเอกซ์โพเนนเชียลตามรูปที่ 4 เนื่องจากขนาดแกรมที่เพิ่มขึ้น 1 แกรมจะต้องเก็บลำดับของตัวอักษรยาวขึ้นกว่าเดิม 1 อักขระ ข้อมูลจึงเพิ่มขึ้นเป็นจำนวนเท่าของจำนวนแกรมที่ต่ำกว่า เวลาที่ใช้ในการจำแนกประเภทของชื่อจะพบว่าในจำนวนแกรมที่น้อยกว่าจะใช้เวลามากกว่าจำนวนแกรมที่มากกว่าเนื่องจากว่าการทำในแต่ละครั้งต้องแบ่งชื่อโดเมนเพื่อประมวลผลออกเป็นส่วนๆ แกรมที่น้อยกว่าจะต้องประมวลผลมากกว่าแกรมที่มากกว่าอยู่ 1 รอบเสมอ ดังแสดงในตารางที่ 6 จากข้อมูลนี้จะพบว่าโมเดลมีค่าความถูกต้องและค่าคะแนนเอฟวันสูงที่สุดเมื่อใช้จำนวนแกรมเท่ากับ 4 ดังนั้นโมเดลที่นำมาใช้ในโมดูลการประมวลผลจึงเลือกใช้จำนวนแกรมเท่ากับ 4 เนื่องจากระบบให้ความสำคัญกับค่าความถูกต้องมากกว่าเวลาที่ใช้ในการเรียนรู้และพื้นที่ที่ใช้ในการเก็บโมเดล

5.2 ผลของระบบ

ผลลัพธ์ของระบบจะแบ่งเป็นประสิทธิภาพของระบบ และหน้าเว็บไซต์ที่ใช้ในการแสดงผล โดยมีรายละเอียดดังนี้

5.2.1. ประสิทธิภาพของระบบ

ผลการทดสอบของระบบแสดงให้เห็นถึงเวลาที่ข้อมูลหนึ่งๆ ใช้เมื่ออยู่ในแต่ละโมดูลในระบบ เวลาที่โมดูลรวบรวมข้อมูลแล้วบันทึกข้อมูลลงในฐานใช้เวลา 20.3 วินาทีต่อข้อมูล 10,000 ข้อมูล หรือ 2 มิลลิวินาที ต่อ 1 ข้อมูล และโมดูลการประมวลผลจะใช้เวลาในการทำงานประมาณ 0.16 วินาทีต่อ 1 ข้อมูล โดยมีรายละเอียด ดังตารางที่ 7 และโมดูลการแสดงผลจะใช้เวลาในการดึงข้อมูลจากฐานข้อมูลประมาณ 0.17 วินาทีต่อ 1 ข้อมูล จากข้อมูลดังกล่าวข้างต้นจะสามารถสรุปได้ว่าข้อมูล 1 ข้อมูลจะใช้เวลาในการทำงานทั้งหมดในระบบประมาณ 0.5 วินาที

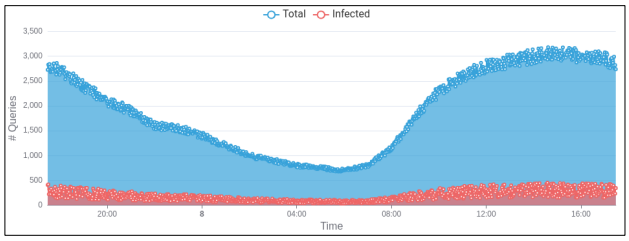
ตารางที่ 7 เวลาที่ใช้ในการทำงานตามขั้นตอนต่าง ๆ ของโมดูลการประมวลผลต่อข้อมูล 1 ข้อมูล

ขั้นตอนภายในโมดูลการประมวลผล	เวลาที่ใช้ (มิลลิวินาที)
ดึงข้อมูลดีเอ็นเอสจากอีลาสติกเสิร์ชเข้าสู่โปรแกรมจำแนกประเภท	8.334
ตรวจสอบชื่อโดเมนกับฐานข้อมูลบัญชีขาว	7.831
คำนวณหาความน่าจะเป็น	7.554×10^{-3}
อัพเดตประเภทของข้อมูลลงในฐานข้อมูล	1.05×10^{2}

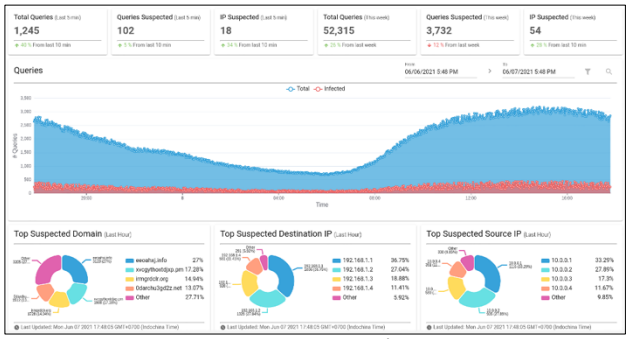
5.2.2. โมดูลการแสดงผล

เว็บไซต์สำหรับแสดงข้อมูลจะแบ่งออกเป็นสามส่วนหลักประกอบไปด้วยแผนภูมิเส้นสำหรับแสดงจำนวนข้อมูล แผนภูมิโดนัทสำหรับแสดงสัดส่วนของข้อมูล และประวัติการใช้งานแผนภูมิเส้นในรูปที่ 6 เป็นแดชบอร์ดแสดงภาพรวมระบบ และ รูปที่ 5 แสดงจำนวนข้อมูลที่เข้ามาในระบบตามเวลาโดยแสดงข้อมูลการใช้ดีเอ็นเอสทั้งหมด และการใช้ดีเอ็นเอสเฉพาะที่คาดว่าเป็นโดเมนสุ่ม

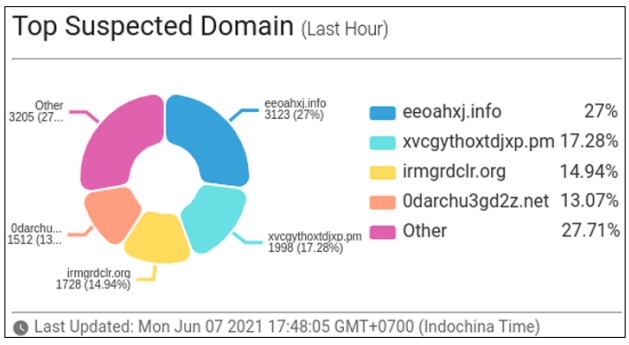
แผนภูมิโดนัทในรูปที่ 7 แสดง 5 อันดับแรกของโดเมนที่ถูกสุ่ม (DGAs) ส่วนรูปที่ 8 แสดงตารางประวัติการใช้งานเซิร์ฟเวอร์ดีเอ็นเอสย้อนหลังเพื่อที่จะทำให้ผู้ดูแลระบบสามารถตรวจสอบและระบุผู้ใช้งานในไอพีแอดเดรสเดียวกันในเวลาที่แตกต่างกันได้โดยสะดวก



รูปที่ 5 ตัวอย่างแผนภูมิเส้น



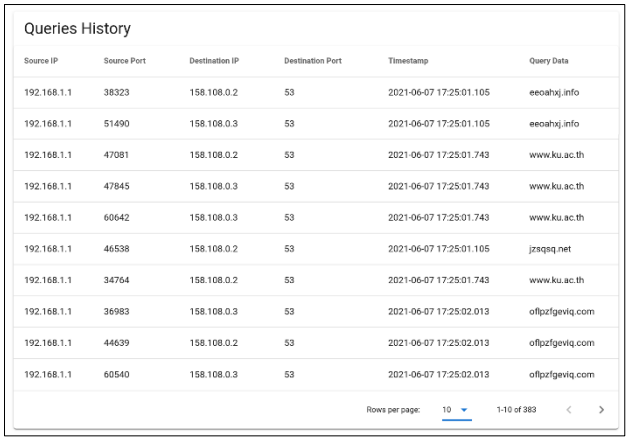
รูปที่ 6 ตัวอย่างแดชบอร์ดสำหรับแสดงข้อมูล



รูปที่ 7 ตัวอย่างแผนภูมิโดนัท

6. สรุปผล

ระบบตรวจสอบอุปกรณ์ที่ติดบอตเน็ตจากข้อมูลจราจรดีเอ็นเอสพัฒนาขึ้นโดยใช้ข้อมูลวิเคราะห์ข้อมูลดีเอ็นเอสเทียบกับ AGDs ด้วยวิธีเอ็นแกรม โดยใช้ค่า N = 4 และแบ่งประเภทด้วยเกณฑ์ซึ่งค่าทั้งหมดที่นำมาใช้นั้นได้ผ่านการทดสอบเพื่อให้ได้ค่าที่เหมาะสมกับการแบ่งประเภทของชื่อโดเมน



รูปที่ 8 ตัวอย่างตารางแสดงประวัติการร้องขอดีเอ็นเอส

7. เอกสารอ้างอิง

[1] A. K. Sood and S. Zeadally, "A Taxonomy of Domain-Generation Algorithms," IEEE Security & Privacy, vol. 14, no. 4, pp. 46-53, 2016.

[2] Network Security Research Lab at 360. (8 November 2020). Netlab DGA Project. [Online] Available: https://data.netlab.360.com/

[3] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic," in 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 16-19 Oct. 2007, pp. 715-720.

[4] K. Alieyan, M. Anbar, A. Almomani, R. Abdullah, and M. Alauthman, "Botnets Detecting Attack Based on DNS Features," in International Arab Conference on Information Technology , 28-30 Nov. 2018, pp. 1-4.

[5] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, Melbourne, Australia, 2010: Association for Computing Machinery.

[6] T. Wang, X. Hu, J. Jang, S. Ji, M. Stoecklin, and T. Taylor, "BotMeter: Charting DGA-Botnet Landscapes in Large Networks," in IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 27-30 June 2016, pp. 334-343.

[7] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow," in IFIP/IEEE International Symposium on Integrated Network Management, 11-15 May 2015, pp. 1304-1309.

[8] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," in Proceedings of the 4th International ICST Conference on COMmunication System softWAre and middlewaRE, Dublin, Ireland, 2009: Association for Computing Machinery.

[9] T.-S. Wang, H.-T. Lin, W.-T. Cheng, and C.-Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," Computers & Security, vol. 64, pp. 1-15, 1 Jan. 2017.

[10] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in Ndss, 2011, pp. 1-17.

[11] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains," ACM Trans. Inf. Syst. Secur., vol. 16, no. 4, 2014.

[12] X. D. Hoang and Q. C. Nguyen, "Botnet Detection Based On Machine Learning Techniques Using DNS Query Data," Future Internet, vol. 10, no. 5, p. 43, 2018.

[13] J. Mao, J. Zhang, Z. Tang, and Z. Gu, "DNS anti-attack machine learning model for DGA domain name detection," Physical Communication, vol. 40, 1 Jun. 2020.

[14] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks," in 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 22-25 June 2015, pp. 403-414.

[15] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Efficient and Accurate Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks," ACM Trans. Priv. Secur., vol. 19, no. 2, p. Article 4, 2016.

[16] Alexa. Alexa's Top Sites [Online] Available: https://s3.amazonaws.com/alexa-static/top-1m.csv.zip

[17] M. Kuhn and K. Johnson, Applied Predictive Modeling, New York: Springer, 2013.