ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing

นายวุฒิชัย อ่อนสำลี

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing

นายวุฒิชัย อ่อนสำลี

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing

โดย นายวุฒิชัย อ่อนสำลี

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ

คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

.....

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

.....
ประธานกรรมการ

(รองศาสตราจารย์ ดร.สุมิตรา นวลมีศรี)

.....
อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์)

.....
กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

ชื่อ : นายวุฒิชัย อ่อนสำลี
ชื่อการค้นคว้าอิสระ : ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing
สาขาวิชา : การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์
ปีการศึกษา : 2567

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบต้นแบบสำหรับ ตรวจจับและแจ้งเตือนภัยไซเบอร์ โดยอาศัยฐานข้อมูลจาก MISP Threat Sharing ร่วมกับเครื่องมือ Logstash และ OpenSearch เพื่อช่วยให้องค์กร โดยเฉพาะองค์กรขนาดเล็ก สามารถตอบสนองต่อภัยคุกคามทางไซเบอร์ได้อย่างรวดเร็วและแม่นยำ ระบบประกอบด้วย 4 ส่วนหลัก ได้แก่ (1) การจัดเก็บและเตรียมข้อมูล Blacklist IP Address จากฐานข้อมูล MISP Threat Sharing (2) การส่งข้อมูล Log Traffic จาก Firewall FortiGate 50E ไปยัง Logstash เพื่อประมวลผล (3) การวิเคราะห์ข้อมูล IP Address พร้อมการเพิ่ม Tag ระบุสถานะว่าเป็น Blacklist IP Address หรือ Non-Blacklist IP Address และ (4) การแสดงผลข้อมูลและแจ้งเตือนภัยผ่าน OpenSearch

ระบบที่พัฒนาขึ้นนี้สามารถช่วยให้องค์กรขนาดเล็กลดค่าใช้จ่ายในการจัดซื้ออุปกรณ์ด้านความปลอดภัยราคาแพง และเพิ่มความสามารถในการรับมือภัยคุกคามไซเบอร์ได้อย่างมีประสิทธิภาพ

(มีจำนวนทั้งสิ้น 42 หน้า)

คำสำคัญ : กระบวนการบริหารจัดการเหตุการณ์ผิดปกติ, วิทยาการวิเคราะห์ข้อมูล, การตรวจสอบรูปแบบอัตโนมัติ

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Wuttichai Onsamlee
Independent Study Title : The Cyber Threat Detection and Alert System Using MISP Threat Sharing Database
Major Field : Network and Information Security Management
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Assistant Professor Dr. Nawaporn Wisitpongphan
Academic Year : 2024

ABSTRACT

This research aims to develop a prototype system for detecting and alerting cyber threats using data from MISP Threat Sharing combined with Logstash and OpenSearch tools. The goal is to help organizations, especially small-sized ones, respond to cyber threats quickly and accurately. The system consists of four main components: (1) storing and preparing Blacklist IP Address data from the MISP Threat Sharing database, (2) forwarding Log Traffic data from the FortiGate 50E Firewall to Logstash for processing, (3) analyzing IP Address data with the addition of tags to identify whether they are Blacklist IP Addresses or Non-Blacklist IP Addresses, and (4) displaying data and issuing alerts via OpenSearch.

The developed system can assist small organizations in reducing the costs of purchasing expensive security equipment and enhance their ability to effectively handle cyber threats.

(Total 42 Pages)

Keywords: Incident Management, Data Analytic, Automated Audit

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงด้วยดี ด้วยความอนุเคราะห์ คำแนะนำ และการดูแลเอาใจใส่อย่างใกล้ชิดจาก ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์ อาจารย์ที่ปรึกษา ซึ่งได้ให้ความช่วยเหลือด้วยความเอาใจใส่และเมตตาอย่างยิ่ง นอกจากนี้ ข้าพเจ้าขอขอบพระคุณ คณาจารย์ภาควิชาการบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศ ทุกท่าน ที่ได้ถ่ายทอดองค์ความรู้ในแขนงต่าง ๆ ซึ่งมีส่วนสำคัญต่อการดำเนินงานวิจัยในครั้งนี้

ข้าพเจ้าขอแสดงความขอบคุณเป็นอย่างยิ่งต่อ บริษัท บีพีเอส แอร์ แอนด์ เซอร์วิส จำกัด ที่ให้การสนับสนุนข้อมูลและทรัพยากรที่จำเป็นต่อการดำเนินงานวิจัย ซึ่งมีส่วนสำคัญอย่างยิ่งในการพัฒนาระบบและการทดลองที่เกี่ยวข้องในงานวิจัยฉบับนี้

ท้ายที่สุด ข้าพเจ้าขอแสดงความขอบคุณจากใจต่อครอบครัว เพื่อน ๆ และพี่น้องที่คอยให้การสนับสนุนและเป็นกำลังใจตลอดมา จนทำให้การค้นคว้าอิสระฉบับนี้เสร็จสมบูรณ์

วุฒิชัย อ่อนสำลี

สารบัญ

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

สารบัญภาพ (ต่อ)

หน้า

สารบัญภาพ (ต่อ)

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน ภัยคุกคามทางไซเบอร์มีแนวโน้มทวีความรุนแรงและมีความหลากหลายมากขึ้น ทำให้องค์กรทุกขนาด ไม่ว่าจะเป็นองค์กรขนาดใหญ่หรือขนาดเล็ก ต่างเผชิญกับความเสี่ยงจากการถูกโจมตีทางไซเบอร์ ไฟร์วอลล์ทั่วไปสามารถช่วยป้องกันการเข้าถึงที่ไม่ได้รับอนุญาต แต่ไม่สามารถตรวจจับและป้องกันภัยคุกคามขั้นสูงได้อย่างมีประสิทธิภาพ หากต้องการระบบป้องกันที่มีความแข็งแกร่งขึ้น องค์กรจำเป็นต้องใช้อุปกรณ์ไฟร์วอลล์รุ่นใหม่ (Next-generation firewall) [1] ซึ่งมีต้นทุนสูง และอาจยังไม่สามารถแสดงรายละเอียดของ IP Address ที่เป็นอันตรายจากภายนอกได้อย่างชัดเจน หนึ่งในมาตรการที่นิยมใช้ในการลดผลกระทบจากภัยคุกคามดังกล่าว คือการบล็อก IP Address ที่ปรากฏในรายชื่อ Blacklist เพื่อป้องกันการเข้าถึงระบบโดยไม่ได้รับอนุญาต

การตรวจสอบสถานะ Blacklist ของ IP Address มักกระทำโดยการนำ IP Address ไปค้นหาผ่านเว็บไซต์ที่ให้บริการตรวจสอบ Blacklist หรือติดตามข่าวสารเกี่ยวกับภัยคุกคามทางไซเบอร์อย่างใกล้ชิด กระบวนการดังกล่าวต้องใช้ทั้งเวลาและทรัพยากรในการดำเนินการ ส่งผลให้การตอบสนองต่อภัยคุกคามเป็นไปอย่างล่าช้า และอาจก่อให้เกิดความเสียหายต่อองค์กรได้ เพื่อเพิ่มประสิทธิภาพในการป้องกันภัยคุกคามไซเบอร์ จึงมีการนำฐานข้อมูล IP Address ที่น่าสงสัยหรืออยู่ใน Blacklist มาใช้ในการบล็อกการเข้าถึงเครือข่ายล่วงหน้า วิธีการนี้ไม่เพียงแต่ช่วยลดความเสี่ยงจากการถูกโจมตีเท่านั้น แต่ยังช่วยเพิ่มความมั่นคงและเสถียรภาพของระบบเครือข่าย ทำให้องค์กรสามารถดำเนินงานได้อย่างต่อเนื่องและมั่นใจมากยิ่งขึ้น

MISP Threat Sharing [2] เป็นแพลตฟอร์มโอเพนซอร์สที่ออกแบบมาเพื่อให้หน่วยงานต่าง ๆ สามารถแบ่งปันข้อมูลเกี่ยวกับภัยคุกคามทางไซเบอร์ได้อย่างรวดเร็วและมีประสิทธิภาพ โดยเฉพาะข้อมูลที่เกี่ยวข้องกับมัลแวร์ อินดิเคเตอร์ของการถูกบุกรุก และภัยคุกคามอื่น ๆ ส่วน Logstash [3] เป็นเครื่องมือช่วยในการจัดเตรียมข้อมูลบันทึกให้พร้อมใช้งาน ด้วยเหตุนี้ การพัฒนาเครื่องมือที่สามารถตรวจจับและแจ้งเตือนภัยไซเบอร์โดยอาศัยการเทียบเคียงข้อมูล IP Address กับฐานข้อมูลของ MISP Threat Sharing และแสดงผลวิเคราะห์ผ่าน OpenSearch [4] จึงกลายเป็นสิ่งสำคัญ เครื่องมือนี้ช่วยในการตรวจสอบและแจ้งเตือนภัยคุกคาม ทำให้สามารถวิเคราะห์และจัดการปัญหาที่เกิดขึ้นในเครือข่ายได้อย่างมีประสิทธิภาพและรวดเร็วมากยิ่งขึ้น

การค้นคว้าอิสระนี้มีวัตถุประสงค์เพื่อพัฒนาเครื่องมือป้องกันภัยคุกคามที่มีความสามารถตรวจจับและแจ้งเตือนภัยไซเบอร์จาก IP Address ที่เป็นอันตรายให้กับองค์กรขนาดเล็กที่มีงบประมาณจำกัด เพื่อลดผลกระทบที่เกิดจากภัยคุกคามทางไซเบอร์

1.2 วัตถุประสงค์

เพื่อออกแบบและพัฒนาระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยเทียบเคียงข้อมูล IP Address กับฐานข้อมูล MISP Threat Sharing และแสดงผลวิเคราะห์ผ่าน OpenSearch ได้อย่างอัตโนมัติ

1.3 ขอบเขตการวิจัย

1.3.1 เครื่องมือที่ใช้ ได้แก่

1.3.1.1 Firewall FortiGate 50E

1.3.1.2 Lenovo M700 CPU i5 RAM24GB

1.3.1.3 Logstash

1.3.1.4 OpenSearch

1.3.2 ชุดข้อมูลที่ใช้ในการพัฒนาและทดสอบ จำนวน 3 ชุดข้อมูล ได้แก่

1.3.2.1 Blacklist IP Address IPsum

1.3.2.2 Blacklist IP Address Myip.ms

1.3.2.3 Blacklist IP Address blocklist.de

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เพิ่มประสิทธิภาพการตรวจจับและแจ้งเตือนภัยไซเบอร์ จาก IP Address ที่เป็นอันตราย สำหรับหน่วยงานหรือองค์กรที่ขาดแคลนผู้เชี่ยวชาญด้านไซเบอร์

1.4.2 ช่วยให้องค์กรตรวจจับการโจมตีทางไซเบอร์จากที่อยู่ IP Address ที่เป็นอันตรายได้อย่างรวดเร็วและแม่นยำ

1.4.3 ลดโอกาสในการสูญเสียข้อมูลสำคัญและลดความเสียหายที่อาจเกิดขึ้นในองค์กร

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing อาศัยหลักการและเทคนิคจากงานวิจัยและทฤษฎีที่เกี่ยวข้อง ซึ่งสามารถแบ่งออกเป็นหัวข้อต่าง ๆ ดังต่อไปนี้

2.1 MISP Default Feeds

2.2 Firewall Log

2.3 Logstash

2.4 OpenSearch Dashboards

2.5 งานวิจัยที่เกี่ยวข้อง

2.1 MISP Default Feeds

MISP (Malware Information Sharing Platform) เป็นแพลตฟอร์มโอเพนซอร์สที่ได้รับการพัฒนาเพื่อรองรับการแบ่งปันข้อมูลภัยคุกคามทางไซเบอร์ระหว่างองค์กรต่าง ๆ โดยช่วยให้สามารถรวบรวม จัดเก็บ และแจกจ่ายข้อมูลภัยคุกคามได้อย่างเป็นระบบและมีประสิทธิภาพ

MISP Default Feeds[8] คือ ชุดฟีดข้อมูลสาธารณะ (OSINT) ที่มีอยู่ใน MISP โดยค่าเริ่มต้น ฟีดเหล่านี้สามารถใช้เป็นแหล่งข้อมูลสำหรับตรวจสอบความสัมพันธ์ระหว่างเหตุการณ์และแอตทริบิวต์ต่าง ๆ โดยไม่จำเป็นต้องนำข้อมูลเข้าสู่ระบบโดยตรง ระบบฟีดของ MISP ช่วยให้สามารถวิเคราะห์ข้อมูลได้อย่างรวดเร็ว รวมถึงสามารถเปรียบเทียบฟีดจากแหล่งต่าง ๆ เพื่อเพิ่มประสิทธิภาพในการประเมินภัยคุกคามทางไซเบอร์ นอกจากนี้ MISP ยังรองรับฟีดข้อมูลในหลากหลายรูปแบบ เพื่อให้สามารถดึง วิเคราะห์ และเผยแพร่ข้อมูลภัยคุกคามได้อย่างมีประสิทธิภาพ รูปแบบหลักของฟีดใน MISP มีดังต่อไปนี้

2.1.1. MISP Standardized Format

ฟีดรูปแบบนี้ใช้โครงสร้าง MISP JSON Format ซึ่งออกแบบมาให้ทำงานร่วมกับแพลตฟอร์ม MISP ได้อย่างสมบูรณ์ รองรับการดึงและแจกจ่ายข้อมูลภัยคุกคาม พร้อมทั้งสามารถจัดการ Indicators of Compromise (IoC) ได้อย่างมีประสิทธิภาพ ฟีดประเภทนี้ช่วยให้สามารถตรวจจับและเชื่อมโยง IoC กับเหตุการณ์ที่มีอยู่ภายในระบบ รวมถึงรองรับการวิเคราะห์แนวโน้มของภัยคุกคามและการแบ่งปันข้อมูลกับชุมชนด้านความปลอดภัยทางไซเบอร์ ฟีดที่ใช้มาตรฐานของ MISP มักมาจากแหล่งข้อมูลที่เชื่อถือได้ เช่น CIRCL, Abuse.ch และ MalwareBazaar ซึ่งเป็นแพลตฟอร์มที่ให้บริการข้อมูล Threat Intelligence สำหรับติดตามและป้องกันภัยคุกคามทางไซเบอร์

2.1.2. CSV Format

ฟีดในรูปแบบ Comma-Separated Values (CSV) ช่วยให้สามารถนำเข้าข้อมูลภัยคุกคามที่อยู่ในรูปแบบของตารางข้อมูล โดยผู้ใช้สามารถกำหนดคอลัมน์ที่ต้องการนำเข้าได้ เช่น รายการ IP Address, Domain Name, URLs หรือ Hashes ซึ่งสามารถกำหนดค่าให้ MISP แปลงเป็น Attributes เพื่อใช้ในการวิเคราะห์ภัยคุกคามและเชื่อมโยงกับเหตุการณ์ที่มีอยู่ในระบบได้

2.1.3. Free Text Format

ฟีดในรูปแบบ Free Text เหมาะสำหรับข้อมูลที่ไม่มีโครงสร้างชัดเจน เช่น รายงานภัยคุกคาม (Threat Reports) หรือบันทึกเหตุการณ์โจมตีทางไซเบอร์ (Incident Logs) โดย MISP สามารถวิเคราะห์และสกัด IoC ออกจากเนื้อหาโดยอัตโนมัติ ทำให้สามารถดึงข้อมูลที่เกี่ยวข้อง เช่น IP Address, Email, หรือ Hashes จากข้อความที่ไม่มีโครงสร้างตายตัวได้ ช่วยเพิ่มความสามารถในการวิเคราะห์และรวบรวมข้อมูลภัยคุกคามจากแหล่งที่ไม่เป็นทางการ

2.1.4. Network (URL)

ฟีดประเภทนี้ใช้สำหรับดึงข้อมูลภัยคุกคามจากแหล่งภายนอกผ่าน URL ที่กำหนด เช่น https://example.com/feed.json ซึ่งช่วยให้ MISP สามารถรับข้อมูลจากผู้ให้บริการ Threat Intelligence ภายนอก และอัปเดตข้อมูลแบบเรียลไทม์ ฟีดในรูปแบบนี้เหมาะสำหรับการติดตามภัยคุกคามที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง และสามารถตั้งค่าให้มีการดึงข้อมูลเป็นระยะ ๆ เพื่อให้มั่นใจว่าองค์กรได้รับข้อมูลล่าสุดอยู่เสมอ

2.1.5. Local (file)

ฟีดในรูปแบบ Local Feeds ใช้สำหรับนำเข้าข้อมูลภัยคุกคามจากแหล่งข้อมูลภายในองค์กร เช่น ไฟล์ JSON หรือ CSV ที่จัดเก็บอยู่ในเซิร์ฟเวอร์ขององค์กร โดยสามารถกำหนดให้ MISP ดึงข้อมูลจากไฟล์ที่กำหนด เช่น /opt/misp/feeds/myfeed.json และอัปเดตข้อมูลตามความต้องการ ฟีดประเภทนี้เหมาะสำหรับองค์กรที่ต้องการควบคุมข้อมูลภายในเครือข่ายของตนเอง หรือมีข้อจำกัดด้านการเชื่อมต่ออินเทอร์เน็ต เช่น ในสภาพแวดล้อมที่ต้องใช้ Air-Gapped Systems หรือระบบที่ต้องการความปลอดภัยสูง

2.2 Firewall Logs

ข้อมูล log จาก FortiGate Firewall ถูกแบ่งออกเป็น สองส่วนหลัก ได้แก่ Header Fields และ Body Fields [9] โดยแต่ละส่วนมีหน้าที่และองค์ประกอบเฉพาะที่ช่วยให้สามารถวิเคราะห์ข้อมูลเครือข่ายและเหตุการณ์ด้านความปลอดภัยทางไซเบอร์ได้อย่างมีประสิทธิภาพ

2.2.1 Header Fields

Header Fields เป็นส่วนที่บันทึกข้อมูลพื้นฐานของเหตุการณ์ที่เกิดขึ้น ซึ่งปรากฏในทุกประเภทของ log บน FortiGate ประกอบด้วยองค์ประกอบสำคัญ ได้แก่

2.2.1.1 Date (date) วันที่ที่มีการบันทึก log

2.2.1.2 Time (time) เวลาที่เกิดเหตุการณ์

2.2.1.3 Log ID (logid) หมายเลขเฉพาะของ log ซึ่งใช้ระบุประเภทและเหตุการณ์ที่เกี่ยวข้อง

2.2.1.4 Type (type) ประเภทของ log เช่น traffic, event, security

2.2.1.5 Subtype (subtype) ประเภทย่อยของ log เช่น local, forward, multicast (สำหรับ traffic logs) หรือ IPS, Antivirus, Web Filter (สำหรับ security logs)

2.2.1.6 Virtual Domain (vd) ระบุชื่อของ Virtual Domain (VDOM) ที่ log ถูกบันทึก

2.2.1.7 Severity Level (pri) ระดับความรุนแรงของเหตุการณ์ เช่น notice, warning, error, critical

2.2.2. Body Fields

Body Fields เป็นส่วนที่บรรจุรายละเอียดของเหตุการณ์ที่ถูกบันทึก ซึ่งแตกต่างกันไปตามประเภทของ log โดยทั่วไปแล้ว สามารถแบ่งออกเป็น 3 ประเภทหลัก ได้แก่ Traffic Log, Security Log และ Event Log

2.2.2.1 Traffic Log (บันทึกข้อมูลการรับส่งเครือข่ายผ่าน FortiGate)

2.2.2.1.1 Source IP (srcip) IP ต้นทางของแพ็กเก็ต

2.2.2.1.2 Source Port (srcport) พอร์ตต้นทางของแพ็กเก็ต

2.2.2.1.3 Destination IP (dstip) IP ปลายทางของแพ็กเก็ต

2.2.2.1.4 Destination Port (dstport) พอร์ตปลายทางของแพ็กเก็ต

2.2.2.1.5 Protocol (proto) โปรโตคอลที่ใช้ เช่น TCP, UDP

2.2.2.1.6 Service (service) บริการหรือแอปพลิเคชันที่เกี่ยวข้อง เช่น HTTP, HTTPS

2.2.2.1.7 Session ID (sessionid) หมายเลขเซสชันที่กำหนดให้กับการเชื่อมต่อ

2.2.2.1.8 Status (status) สถานะของเซสชัน เช่น start, close, deny

2.2.2.1.9 Bytes Sent (sentbyte) จำนวนข้อมูลที่ถูกส่งจากต้นทาง

2.2.2.1.10 Bytes Received (rcvdbyte) จำนวนข้อมูลที่ถูกส่งจากปลายทาง

2.2.2.2 Security Log (บันทึกเหตุการณ์ด้านความปลอดภัย)

2.2.2.2.1 Action (action) การดำเนินการที่ใช้กับทราฟฟิก เช่น allow, block, reset

2.2.2.2.2 Attack Name (attack) ชื่อของการโจมตีที่ถูกตรวจพบ (เช่น SQL Injection, DDoS Attack)

2.2.2.2.3 Virus Name (virus) ชื่อไวรัสที่ถูกตรวจพบ

2.2.2.2.4 Application Name (app) ชื่อของแอปพลิเคชันที่ถูกบล็อกหรือได้รับอนุญาต

2.2.2.2.5 Category (catdesc) หมวดหมู่ของภัยคุกคาม เช่น Malware, Phishing

2.2.2.2.6 User (user) ชื่อบัญชีผู้ใช้ที่เกี่ยวข้องกับเหตุการณ์

2.2.2.2.7 Reputation Score (crscore) คะแนนความน่าเชื่อถือของไคลเอนต์

2.2.2.3 Event Log (บันทึกเหตุการณ์ระบบ)

2.2.2.3.1 System Event (msg) ข้อความที่อธิบายเหตุการณ์ที่เกิดขึ้น เช่น Firewall policy applied, Configuration change detected

2.2.2.3.2 Administrator (user) ชื่อผู้ดูแลระบบที่ดำเนินการเปลี่ยนแปลง

2.2.2.3.3 Configuration Change (config_id) รายละเอียดของการเปลี่ยนแปลงค่าการตั้งค่า

2.2.2.3.4 High Availability (ha_role) บทบาทของอุปกรณ์ในโหมด HA เช่น Master, Slave

ตัวอย่าง log แสดงดังภาพที่ 2-1

```
date=2025-02-05 time=23:40:42 devname="BPS-FW-01"
devid="FGT50E3U17012538" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root"
eventtime=1738773642498225300 tz="+0700" srcip=10.88.100.54
srcname="A47600" srcport=35669 srcintf="LAN-INTERNAL"
srcintfrole="lan" dstip=74.125.200.156 dstport=443 dstintf="wan1"
dstintfrole="wan" srcuuid="800eb708-3946-51ef-83f3-
c3daa97d8445" dstuuid="4e234d8c-3944-51ef-2d10-
0e6d285b0de5" sessionid=482079 proto=6 action="close"
policyid=1 policytype="policy" poluuid="c9dfa72a-3946-51ef-e3f0-
636c9d3b7115" service="HTTPS" dstcountry="Singapore"
srccountry="Reserved" trandisp="snat" transip=192.168.1.102
transport=35669 duration=242 sentbyte=4117 rcvdbyte=4254
sentpkt=18 rcvdpkt=19 appcat="unscanned" sentdelta=203
rcvddelta=309 osname="Windows" srcswversion="8.1"
mastersrcmac="b0:dc:ef:89:91:17" srcmac="b0:dc:ef:89:91:17"
srcserver=0
```

ภาพที่ 2-1 Example Log from Firewall FortiGate 50E

2.3 Logstash

Logstash [10] เป็นซอฟต์แวร์โอเพนซอร์สที่พัฒนาโดย Elastic ทำหน้าที่เป็นตัวกลางในการรวบรวม ประมวลผล และส่งต่อข้อมูลจากแหล่งต่าง ๆ ไปยังปลายทางที่กำหนด โดยเป็นส่วนหนึ่งของ Elastic Stack (ELK Stack)

โครงสร้างการทำงานของ Logstash ถูกออกแบบเป็น Pipeline ซึ่งแบ่งออกเป็นสามองค์ประกอบหลัก ได้แก่ Input, Filter และ Output โดยแต่ละขั้นตอนสามารถกำหนดค่าผ่าน Plugins ต่าง ๆ เพื่อให้สามารถปรับแต่งกระบวนการประมวลผลข้อมูลได้อย่างยืดหยุ่นและเหมาะสมกับลักษณะของข้อมูลที่นำเข้า

2.3.1 Input (การรับข้อมูล)

Inputs เป็นองค์ประกอบสำคัญของ Logstash ที่ทำหน้าที่รับข้อมูลจากแหล่งต่าง ๆ เพื่อนำเข้าสู่กระบวนการประมวลผล โดยรองรับการรับข้อมูลจากหลายช่องทาง โดย Input Plugin ที่ได้รับความนิยม ได้แก่

2.3.1.1 File ใช้สำหรับอ่านข้อมูลจากไฟล์ใน Filesystem โดยมีลักษณะการทำงานคล้ายกับคำสั่ง tail -F บนระบบปฏิบัติการ UNIX

2.3.1.2 Syslog รับฟังข้อความ Syslog ผ่านพอร์ตมาตรฐาน 514 และแปลงข้อมูลตามมาตรฐาน RFC3164

2.3.1.3 Redis ดึงข้อมูลจาก Redis Server ผ่าน Redis Channels และ Redis Lists โดย Redis มักถูกใช้เป็น Message Broker ในโครงสร้าง Centralized Logstash เพื่อทำหน้าที่จัดคิว (Queue) Logstash Events ที่รับมาจาก Remote Logstash Shippers

2.3.1.4 Beats รองรับและประมวลผล Events ที่ถูกส่งมาจาก Beats ซึ่งเป็นชุดเครื่องมือสำหรับรวบรวมและส่งข้อมูลแบบกระจายศูนย์

Input Plugin แต่ละประเภทได้รับการออกแบบให้สามารถดึงข้อมูลเข้าสู่ระบบได้อย่างมีประสิทธิภาพและเหมาะสมกับแหล่งข้อมูลที่แตกต่างกัน

2.3.2 Filters (การกรองและประมวลผลข้อมูล)

ข้อมูลที่ได้รับจาก Input จะถูกประมวลผลในขั้นตอน Filter เพื่อแปลงให้อยู่ในรูปแบบที่เหมาะสมต่อการวิเคราะห์และจัดเก็บ โดยสามารถใช้ร่วมกับ Conditionals เพื่อตรวจสอบเงื่อนไขและดำเนินการกับข้อมูลเฉพาะที่ตรงตามเกณฑ์ที่กำหนด ตัวอย่าง Filter Plugin ที่ได้รับความนิยม ได้แก่

2.3.2.1 Grok ใช้สำหรับแยกและแปลงข้อมูลที่ไม่มีโครงสร้างให้เป็นข้อมูลที่มีโครงสร้าง และสามารถสืบค้นได้

2.3.2.2 Mutate ใช้สำหรับเปลี่ยนแปลงฟิลด์ของข้อมูล เช่น เปลี่ยนชื่อ ลบ หรือแก้ไขค่า

2.3.2.3 Drop ใช้สำหรับกรองและลบข้อมูลที่ไม่จำเป็นออกจากระบบ เช่น ข้อมูล Debug

2.3.2.4 Clone ใช้ในการสร้างสำเนาของข้อมูล โดยสามารถเพิ่มหรือลบฟิลด์บางส่วนได้

2.3.2.5 GeoIP ใช้สำหรับเพิ่มข้อมูลเกี่ยวกับตำแหน่งทางภูมิศาสตร์ของที่อยู่ IP ซึ่งสามารถแสดงผลเป็นแผนที่ใน Kibana ได้

2.3.3 Output (การส่งต่อข้อมูล)

Output เป็นขั้นตอนสุดท้ายที่ Logstash ส่งข้อมูลที่ผ่านการประมวลผลไปยังปลายทางที่กำหนด โดย Output Plugin แต่ละประเภทรองรับการส่งข้อมูลไปยังระบบที่เหมาะสม ตัวอย่างปลายทางที่ได้รับความนิยม ได้แก่

2.3.3.1 Elasticsearch ใช้สำหรับจัดเก็บและวิเคราะห์ข้อมูลแบบเรียลไทม์

2.3.3.2 File Output บันทึกข้อมูลลงในไฟล์เพื่อใช้ในระบบอื่น

2.3.3.3 Graphite ส่งข้อมูลไปยังระบบมอนิเตอร์เพื่อวิเคราะห์เมตริก

2.3.3.4 StatsD ใช้สำหรับจัดเก็บสถิติและวิเคราะห์ข้อมูล

นอกจากนี้ Logstash ยังรองรับ Codec Plugin ซึ่งสามารถใช้ร่วมกับ Input หรือ Output เพื่อเข้ารหัสหรือถอดรหัสข้อมูลก่อนและหลังการประมวลผล ตัวอย่าง Codec Plugin ที่นิยมใช้ได้แก่ JSON และ Multiline

ด้วยโครงสร้างที่ยืดหยุ่นและ Plugin ที่หลากหลาย Logstash สามารถรองรับการจัดการข้อมูลที่ซับซ้อน และช่วยให้กระบวนการประมวลผลข้อมูลมีประสิทธิภาพมากขึ้น เหมาะสำหรับการนำไปใช้ในงาน Log Management, Security Analytics และ Data Processing

2.4 OpenSearch Dashboards

OpenSearch Dashboards [11] เป็นซอฟต์แวร์โอเพนซอร์สที่ใช้สำหรับการแสดงผลและวิเคราะห์ข้อมูลที่จัดเก็บใน OpenSearch โดยรองรับการสร้างแดชบอร์ด (Dashboards) การสำรวจข้อมูล (Data Exploration) และการสร้างภาพข้อมูล (Data Visualization) เพื่อช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลเชิงลึกได้อย่างมีประสิทธิภาพ

คุณสมบัติหลักของ OpenSearch Dashboards

2.4.1 Data Visualization

รองรับการสร้างแผนภูมิ กราฟ และแผนที่ เพื่อนำเสนอข้อมูลในรูปแบบที่เข้าใจง่าย ช่วยให้สามารถวิเคราะห์แนวโน้มและรูปแบบของข้อมูลได้อย่างชัดเจน

2.4.2 Data Exploration and Discovery

ผู้ใช้สามารถใช้ Discover ซึ่งเป็นเครื่องมือค้นหาและกรองข้อมูล เพื่อวิเคราะห์แนวโน้มและตรวจสอบข้อมูลในเชิงลึก

2.4.3 Dashboard Management

รองรับการสร้างและปรับแต่งแดชบอร์ดเพื่อแสดงข้อมูลที่สำคัญ สามารถกำหนดเลย์เอาต์และปรับแต่งองค์ประกอบต่าง ๆ ให้เหมาะสมกับการใช้งาน

2.4.4 Support for Multiple Query Languages

OpenSearch Dashboards รองรับภาษาค้นหาหลายรูปแบบ เช่น Dashboards Query Language (DQL), SQL, และ Piped Processing Language (PPL) เพื่อเพิ่มความยืดหยุ่นในการสืบค้นข้อมูล

2.4.5 Installation and Configuration

สามารถติดตั้งและกำหนดค่าได้หลายวิธี เช่น Docker, Tarball, RPM, Debian, Helm และ Windows เพื่อให้สามารถใช้งานได้ในสภาพแวดล้อมที่หลากหลาย

ด้วยคุณสมบัติที่ครอบคลุมเหล่านี้ OpenSearch Dashboards เป็นเครื่องมือสำคัญสำหรับการวิเคราะห์และแสดงผลข้อมูลจาก OpenSearch ซึ่งช่วยให้ผู้ใช้สามารถตรวจสอบแนวโน้มของข้อมูล ทำการวิเคราะห์เชิงลึก และนำเสนอข้อมูลได้อย่างมีประสิทธิภาพ

2.5 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาการพัฒนาโครงสร้างพื้นฐานสำหรับการสตรีมข้อมูล เพื่อรองรับการวิเคราะห์บันทึกข้อมูลการจราจรเครือข่าย (log file) Francesco Amori และคณะ [13] ใช้ Filebeat ในการรวบรวมและส่งข้อมูลจากไฟล์ log ใช้ Kafka เป็นตัวกลางจัดระเบียบข้อมูล เพื่อเตรียมความพร้อมก่อนนำไปใช้งาน และใช้ Logstash เพื่อกรองข้อมูล จากนั้นจึงนำข้อมูลเข้าสู่ OpenSearch เพื่อค้นหาและวิเคราะห์ข้อมูลขนาดใหญ่ สำหรับการปรับปรุงการติดตามความปลอดภัยของ CNAF ได้มีการผสานการใช้งาน OpenSearch เข้ากับ Wazuh เพื่อแสดงผลอย่างมีประสิทธิภาพ

ในส่วนของการวิเคราะห์บันทึกข้อมูลการจราจรเครือข่าย เพื่อระบุและตรวจจับกิจกรรมที่เป็นอันตราย Poongkuyil Muse และคณะ [14] ได้เสนอวิธีการใช้กรอบงาน ELK (Elasticsearch, Logstash, Kibana) เพื่อทำการตรวจจับความผิดปกติจากข้อมูลจริงในเวลาจริง โดยใช้ชุดข้อมูลบันทึกจากเซิร์ฟเวอร์ Apache HTTP และจำแนกประเภทความผิดปกติของผู้ใช้ที่ไม่เชื่อถือโดยใช้ฟิลเตอร์ "mutate" ใน Logstash ซึ่ง ฟิลเตอร์ "mutate" จะถูกใช้เพื่อเพิ่มฟิลด์ใหม่ที่ชื่อว่า "bot" ในบันทึก และแสดงผลการวิเคราะห์ข้อมูลบันทึกผ่าน Kibana

นอกจากนี้ Răzvan Stoleriu และคณะ [15] ยังนำเสนอระบบที่ใช้ ELK สำหรับการตรวจจับและวิเคราะห์การโจมตีทางไซเบอร์ โดยใช้ Packetbeat ในการบันทึกข้อมูลการจราจรเครือข่าย มุ่งเน้นไปที่การโจมตีที่เกี่ยวข้องกับการเชื่อมต่อ HTTP และ DNS ข้อมูลเหล่านี้จะถูกส่งไปยัง Logstash เพื่อกรอง IP Address และ DNS ที่ตรงกับ Indicators of Compromise

(IOCs) ที่ระบุไว้ใน MISP จากนั้นข้อมูลจะถูกส่งต่อไปยัง Elasticsearch เพื่อจัดเก็บ และแสดงผลด้วย Kibana

ในขณะเดียวกัน Mohsen Bin Mohamad Hata และคณะ [16] ได้ทำการวิเคราะห์บันทึกข้อมูลการจราจรเครือข่าย เพื่อระบุและตรวจจับกิจกรรมที่เป็นอันตรายในระบบ SIEM โดยใช้บันทึกจากอุปกรณ์เครือข่ายภายนอก (External Network Devices) ซึ่งถูกส่งผ่าน Filebeat และใช้ Logstash Filter ในการกรอง แยก และจัดรูปแบบข้อมูล เช่น การแยกฟิลด์จากข้อความบันทึก และการแปลงข้อมูลให้อยู่ในรูปแบบ JSON โดย Elasticsearch ทำหน้าที่เป็นฐานข้อมูลสำหรับจัดเก็บและค้นหาข้อมูลบันทึก และใช้งาน Splunk เพื่อตรวจสอบ IP reputation ของระบบปลายทางโดยอิงจากข้อมูล OSINT (Open Source Intelligence) และแสดงผลผ่าน Kibana

แพลตฟอร์มที่สามารถแบ่งปันข้อมูลเกี่ยวกับเหตุการณ์หรือภัยคุกคามมีหลายแพลตฟอร์ม Ricardo Fernandes และคณะ [17] เลือกใช้ Malware Information Sharing Platform (MISP) เนื่องจาก MISP ถูกออกแบบมาเพื่อส่งเสริมการเผยแพร่ข้อมูลเกี่ยวกับภัยคุกคามหรือเหตุการณ์ที่เกี่ยวข้อง เช่น Indicators of Compromise (IoC) ภายในชุมชน โดยมีจุดมุ่งหมายเพื่อช่วยให้หน่วยงานต่าง ๆ สามารถเชื่อมต่อและแบ่งปันข้อมูลเกี่ยวกับการโจมตีหรือภัยคุกคามได้อย่างมีประสิทธิภาพ Aleena Terese George และคณะ [18] ได้ใช้ API ของ MISP ในการตรวจสอบสถานะของ IP address ที่เข้าถึงระบบ โดยระบบจะส่งคำขอไปยัง MISP เพื่อตรวจสอบว่า IP address นั้นถูกบล็อกหรือไม่ หากพบว่า IP address ถูกบล็อก ระบบจะกำหนดคะแนนความเชื่อถือเป็น '0' ซึ่งหมายความว่าผู้ใช้หรืออุปกรณ์นั้นจะไม่ได้รับอนุญาตให้เข้าถึงทรัพยากร

นอกจากนี้ยังมีแพลตฟอร์มอื่นที่คล้ายคลึงกัน เช่น OSINT Framework และ STIX/TAXII โดย Renato Marinho และคณะ [19] ใช้ Open Source Intelligence (OSINT) เพื่อระบุและสร้างโปรไฟล์ภัยคุกคามทางไซเบอร์ที่เกิดขึ้นใหม่ โดยใช้ Logstash Twitter Plugin รวบรวมข้อมูลจาก Twitter ซึ่งอัปเดตอย่างรวดเร็วและหลากหลาย และใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) วิเคราะห์ข้อความและคำที่เกี่ยวข้อง พร้อมสร้างโปรไฟล์ของภัยคุกคามโดยเปรียบเทียบกับฐานข้อมูล MITRE ATT&CK เพื่อเข้าใจกลยุทธ์และเทคนิคการโจมตีได้ดีขึ้น และ Asif Siddiqui และคณะ [20] ใช้ STIX (Structured Threat Information Expression) และ TAXII (Trusted Automated eXchange of Indicator Information) สำหรับการจัดการและแลกเปลี่ยนข้อมูลภัยคุกคามทางไซเบอร์ โดย STIX ให้โครงสร้าง Indicators of Compromise (IoC) ที่ชัดเจน เช่น IP addresses, domains, URLs และ hashes ขณะที่ TAXII อำนวยความสะดวกในการแลกเปลี่ยนข้อมูลเหล่านี้ระหว่างระบบต่าง ๆ อย่างปลอดภัยและอัตโนมัติ ในงานวิจัยนี้ IPtables ใช้ข้อมูล IoC เพื่อสร้าง access control lists (ACLs) ที่สามารถปรับเปลี่ยนตามสถานการณ์ ซึ่งช่วยในการบล็อกการเข้าถึงจาก

แหล่งที่เป็นอันตรายได้อย่างรวดเร็ว และ Suricata ตรวจจับการบุกรุกและส่ง log ไปยัง Logstash สำหรับการเก็บรวบรวมและวิเคราะห์ข้อมูลอย่างมีประสิทธิภาพ

บทที่ 3
วิธีการดำเนินงานวิจัย

จากการศึกษาทฤษฎีที่สำคัญและงานวิจัยที่เกี่ยวข้อง ผู้วิจัยได้สังเคราะห์องค์ความรู้เพื่อนำมาประยุกต์ใช้ในการจัดทำระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing โดยสามารถกำหนดขั้นตอนในการดำเนินการวิจัยได้ดังต่อไปนี้

3.1 ศึกษาและวิเคราะห์ปัญหาที่เกี่ยวข้องกับงานวิจัย

3.2 การออกแบบระบบและอุปกรณ์

3.3 การพัฒนาระบบและอุปกรณ์

3.4 ทดสอบการใช้งานของระบบและอุปกรณ์

3.1 ศึกษาและวิเคราะห์ปัญหาที่เกี่ยวข้องกับงานวิจัย

งานวิจัยนี้ได้นำอุปกรณ์ Firewall FortiGate 50E มาใช้ในการส่ง Log Traffic ไปยัง Logstash ซึ่งทำหน้าที่คัดกรอง (Filter) ข้อมูล โดยระบบจะทำการกรอง Destination IP Address และตรวจสอบความสอดคล้องกับ Dataset Blacklist IP Address หากพบว่า Destination IP Address ตรงกับข้อมูลใน Blacklist Dataset ระบบจะกำหนด Tag เป็น "blacklist-ip-address" แต่หากไม่พบ จะกำหนด Tag เป็น "none-blacklist-ip-address" เมื่อการคัดกรองและติดแท็กเสร็จสิ้น Log Traffic ที่ผ่านการประมวลผลจะถูกส่งไปจัดเก็บใน OpenSearch เพื่อใช้ในการแสดงผลและวิเคราะห์ข้อมูล ทั้งนี้ Dataset Blacklist IP Address ได้มาจาก MISP Default Feeds โดยใช้ Python ทำการดึงข้อมูลจากแหล่งที่มาในทุก ๆ 1 ชั่วโมง ในการดำเนินการวิจัย ผู้วิจัยได้ศึกษาขั้นตอนการพัฒนาและดำเนินงานตามลำดับดังต่อไปนี้

3.1.1 ศึกษาการทำงานของอุปกรณ์ Firewall FortiGate 50E ในการส่ง Log Traffic รวมถึงโครงสร้างและเนื้อหาของ Log เพื่อใช้เป็นแนวทางในการกำหนดเงื่อนไขการคัดกรองข้อมูลบน Logstash

3.1.2 ศึกษาระบบโครงสร้างการทำงานของ Logstash รวมถึงกระบวนการติดตั้ง การกำหนดค่าเพื่อรับ Log การคัดกรอง (Filter Log) และการกำหนด Tag บน Logstash

3.1.3 ศึกษาระบบโครงสร้างการทำงานของ OpenSearch รวมถึงวิธีการแสดงผล Log และการตั้งค่า Dashboard เพื่อดึงข้อมูลที่ต้องการมาแสดง

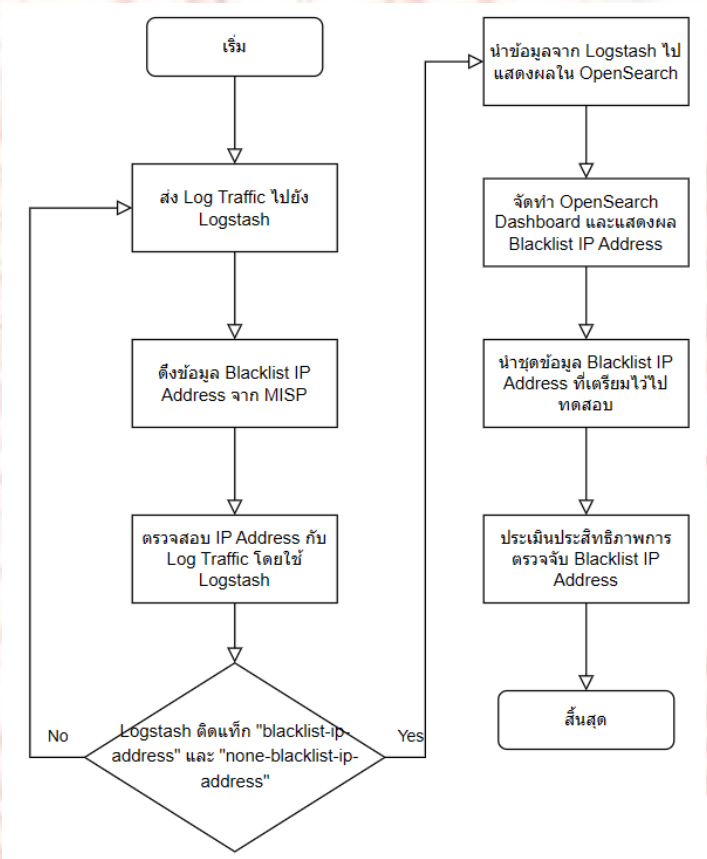
3.1.4 ศึกษาการนำข้อมูลจาก MISP Threat Sharing มาใช้ในการวิเคราะห์และตรวจจับภัยคุกคาม โดยการเทียบเคียงกับ Log Traffic ที่ได้รับจากระบบ

3.2 การออกแบบระบบและอุปกรณ์

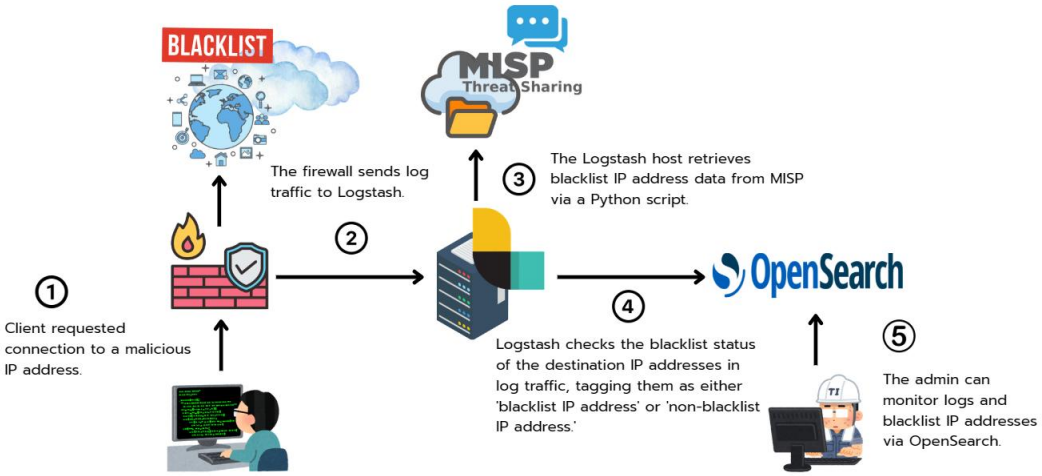
ผู้วิจัยกำหนดขอบเขตของการค้นคว้าอิสระโดยมุ่งเน้นการออกแบบและติดตั้งระบบตรวจจับและแจ้งเตือนภัยไซเบอร์ โดยใช้ฐานข้อมูล MISP Threat Sharing ระบบดังกล่าวดึงข้อมูล Blacklist IP Address จาก MISP Default Feeds เพื่อนำมาเปรียบเทียบกับ Destination

IP Address ที่บันทึกไว้ใน log ของอุปกรณ์ Firewall FortiGate 50E หากพบว่าที่อยู่ IP ตรงกับชุดข้อมูลที่ถูกขึ้นบัญชีดำ ระบบจะทำการแจ้งเตือนทันที

จากนั้น ผู้วิจัยทำการประเมินประสิทธิภาพของระบบโดยตรวจสอบว่าการเปรียบเทียบกับ Blacklist IP Address จากแหล่งข้อมูลทั้ง 3 แห่งให้ผลลัพธ์ตรงกันในสัดส่วนเท่าใด ผู้วิจัยจึงได้กำหนดกระบวนการวิจัยดังแสดงใน ภาพที่ 3-1 และโครงสร้างของแพลตฟอร์มตรวจสอบ Blacklist IP Address ดังภาพที่ 3-2 เพื่อให้มั่นใจว่าระบบสามารถตรวจจับ Blacklist IP Address ได้อย่างมีประสิทธิภาพและสอดคล้องกับขอบเขตและวัตถุประสงค์ของการวิจัย



ภาพที่ 3-1 แผนภาพแสดงกระบวนการวิจัย



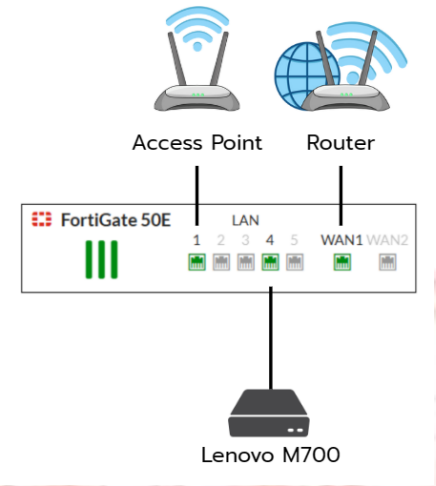
ภาพที่ 3-2 โครงสร้างของแพลตฟอร์มตรวจสอบ Blacklist IP Address

3.3 การพัฒนาระบบและอุปกรณ์

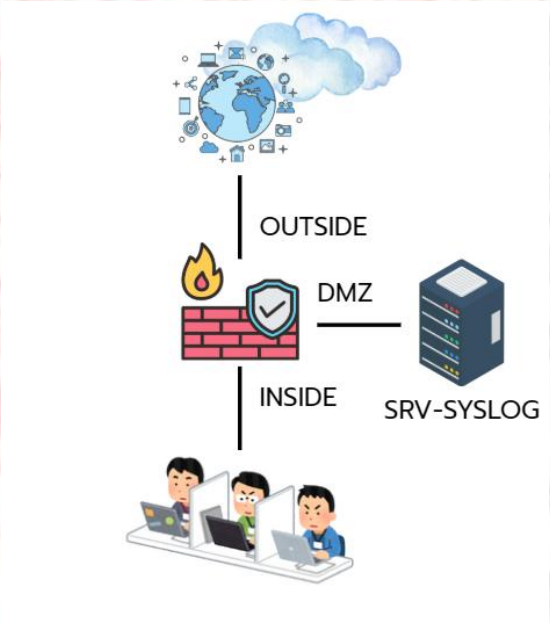
การพัฒนาระบบแบ่งออกเป็น 2 ส่วน ได้แก่ ฮาร์ดแวร์ และ ซอฟต์แวร์ โดยในส่วนของฮาร์ดแวร์ ประกอบด้วย Firewall FortiGate 50E ซึ่งทำหน้าที่เป็น Gateway สำหรับการเชื่อมต่ออินเทอร์เน็ตและบันทึก log traffic และ Lenovo M700 (CPU i5, RAM 24GB) ที่ใช้เป็นอุปกรณ์สำหรับติดตั้งระบบปฏิบัติการ (OS) และรองรับการทำงานของซอฟต์แวร์ที่เกี่ยวข้อง ในส่วนของซอฟต์แวร์ ทำหน้าที่รับ log traffic ดึงข้อมูลจาก MISP Threat Sharing เพื่อนำมาเปรียบเทียบและประมวลผลเพื่อแสดงผลลัพธ์ของการตรวจจับ Blacklist IP Address

3.3.1 การพัฒนาด้านฮาร์ดแวร์

ผู้วิจัยได้ทำการเชื่อมต่ออุปกรณ์ทางกายภาพตามที่แสดงใน ภาพที่ 3-3 เพื่อให้ได้โครงสร้างเครือข่ายเสมือนตาม ภาพที่ 3-4

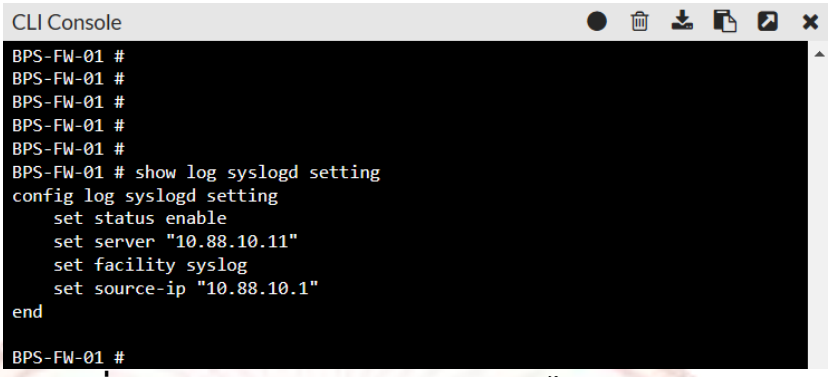


ภาพที่ 3-3 การเชื่อมต่ออุปกรณ์ทางกายภาพ



ภาพที่ 3-4 โครงสร้างเครือข่ายเสมือน

หลังจากดำเนินการเชื่อมต่ออุปกรณ์ตามภาพที่ 3-3 เพื่อให้ได้โครงสร้างเครือข่ายเสมือนตามภาพที่ 3-4 ผู้วิจัยได้ทำการกำหนดการตั้งค่า Firewall FortiGate 50E เพื่อส่ง log ไปยัง Syslog Server โดยใช้คำสั่ง Command Line Interface (CLI) ตามที่แสดงในภาพที่ 3-5

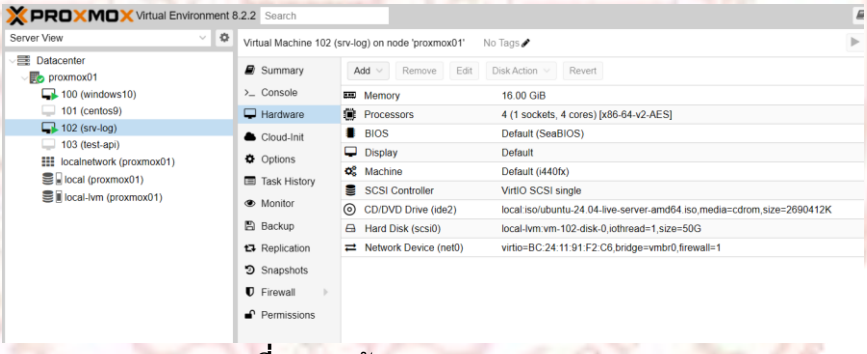


ภาพที่ 3-5 Command Line สำหรับส่ง Log ไปยัง Syslog Server

3.3.2 การพัฒนาด้านซอฟต์แวร์

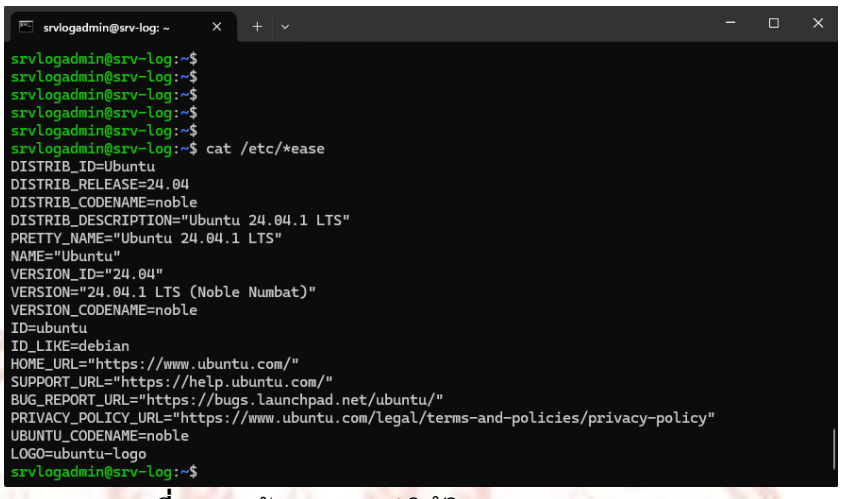
ในส่วนของซอฟต์แวร์ที่นำมาใช้งาน ประกอบด้วย

3.3.2.1 PROXMOX ทำหน้าที่เป็นแพลตฟอร์มสำหรับจัดการและรันเครื่องเสมือน (Virtual Machines) ซึ่งในงานวิจัยนี้ใช้สำหรับรัน SRV-SYSLOG ตามที่แสดงใน ภาพที่ 3-6 โดยทำหน้าที่เป็น Syslog Server สำหรับการจัดเก็บและวิเคราะห์บันทึกข้อมูล (Log) ในระบบ



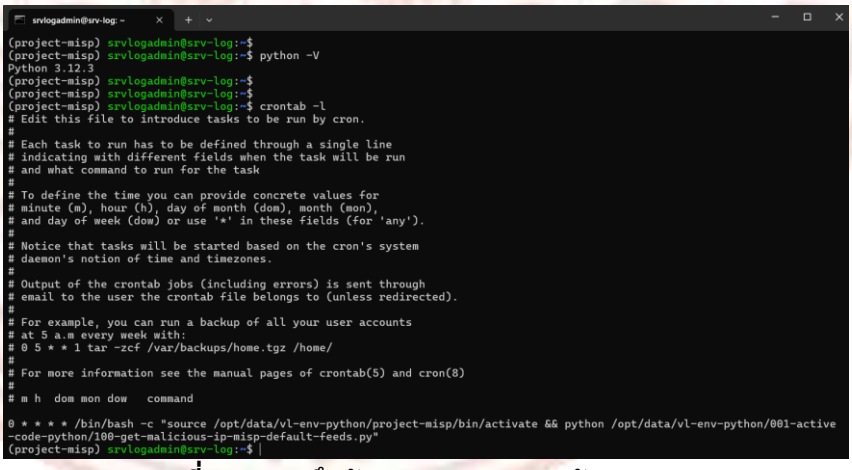
ภาพที่ 3-6 หน้าจอของ PROXMOX VM

3.3.2.2 SRV-SYSLOG เป็นระบบที่ใช้ Ubuntu 24.04.1 LTS เป็นระบบปฏิบัติการหลัก ตามที่แสดงในภาพที่ 3-7 โดยทำหน้าที่รัน Services ที่จำเป็นสำหรับการวิจัย เช่น Python, Logstash, OpenSearch และซอฟต์แวร์อื่น ๆ ที่เกี่ยวข้อง

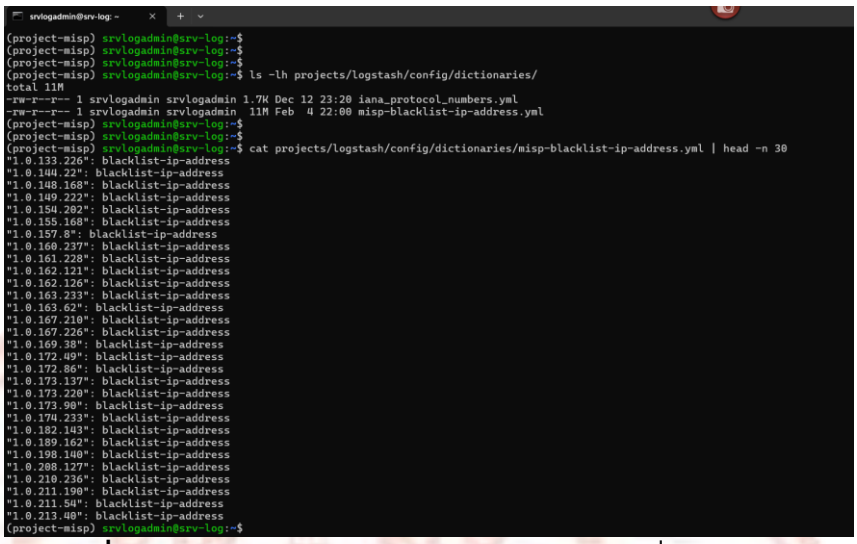


ภาพที่ 3-7 หน้าจอระบบปฏิบัติการของ SRV-SYSLOG

3.3.2.3 Python เป็นภาษาที่ใช้ในการพัฒนาโปรแกรมสำหรับดึงข้อมูล Blacklist IP Address จาก MISP Default Feeds โดยใช้ฟังก์ชัน crontab เพื่อทำการดึงข้อมูลเป็นระยะทุก ๆ 1 ชั่วโมง ตามที่แสดงใน ภาพที่ 3-8 ข้อมูลที่ได้รับจะถูกจัดเก็บและเตรียมไว้สำหรับใช้ในการตรวจสอบ Blacklist IP Address กับ Logstash โดยผลลัพธ์ของข้อมูลแสดงใน ภาพที่ 3-9 ซึ่งช่วยสนับสนุนกระบวนการวิเคราะห์และแจ้งเตือนภัยคุกคาม



ภาพที่ 3-8 การดึงข้อมูล Blacklist IP ด้วย Crontab



ภาพที่ 3-9 ผลลัพธ์ของข้อมูล Blacklist IP Address ที่ดึงจาก MISP

ทั้งนี้ โค้ดของ Python ที่ใช้สำหรับดึงข้อมูลจาก MISP Default Feeds แบ่งเป็น 3 ส่วน ได้แก่

3.3.2.3.1 การดึงรายการ Blacklist IP Address จาก MISP Default Feeds ส่วนนี้เป็นการดึงรายการ Blacklist IP Address ที่ MISP Default Feeds แบ่งปันไว้ โดยสามารถดูตัวอย่างได้ในภาพที่ 3-10 ซึ่งในตัวอย่างเป็นการสร้างฟังก์ชันที่ชื่อว่า fetch_greensnow() และกำหนดตัวแปร url ให้ชี้ไปที่ลิงก์ของ MISP Default Feeds ที่แบ่งปันรายการ Blacklist IP Address โดยในที่นี้ชี้ไปที่ https://blocklist.greensnow.co/greensnow.txt ต่อมาใช้ try ในการดึงข้อมูลด้วย requests.get(url, timeout=10) ซึ่งบอกว่าให้ทำการส่ง HTTP GET request ไปที่ URL ที่กำหนด โดยใช้ timeout = 10 วินาที เพื่อป้องกันการรอโหลดนานเกินไป หากเว็บไซต์ไม่ตอบสนองภายใน 10 วินาที ฟังก์ชันจะ หยุดทำงาน และเข้าสู่ except ส่วน response.raise_for_status() เป็นการใช้ตรวจสอบ รหัสสถานะของ HTTP Response หากการร้องขอสำเร็จ (HTTP Status Code 200 OK) ฟังก์ชันจะทำงานต่อ แต่ถ้าเกิดข้อผิดพลาด เช่น 404 Not Found ฟังก์ชันจะหยุดทำงานทันที และไปที่ except หลังจากนั้นจะใช้ data = response.text.strip().splitlines() ในการดึงข้อมูลที่เป็นข้อความ (string) จากไฟล์ greensnow.txt ซึ่งมีการลบช่องว่าง แยกข้อความแต่ละบรรทัดออกเป็นรายการ และให้คืนค่าข้อมูลโดยการ return data ทั้งนี้จะมีการดักจับข้อผิดผลาดด้วย except requests.exceptions.RequestException ถ้าเกิดข้อผิดพลาด ฟังก์ชันจะ พิมพ์ข้อความแจ้งเตือนเป็น Error fetching greensnow: {รายละเอียดข้อผิดพลาด} เช่น The request timed out.

```python
def fetch_greensnow():
    url = "https://blocklist.greensnow.co/greensnow.txt"
    try:
        response = requests.get(url, timeout=10)
        response.raise_for_status()
        data = response.text.strip().splitlines()
        return data
    except requests.exceptions.RequestException as e:
        print(f"Error fetching greensnow: {e}")
        return []
```

ภาพที่ 3-10 โค้ด Python ที่ใช้ดึงข้อมูลจาก MISP Default Feeds

3.3.2.3.2 การรวม Blacklist IP Address จาก MISP Default Feeds โค้ดส่วนนี้ทำหน้าที่รวม Blacklist IP Address จากทุกแหล่งข้อมูลที่ดึงมา โดยใช้ set() เพื่อเก็บค่าที่ไม่ซ้ำกัน ดังภาพที่ 3-11

```python
# Combine results from all sources
combined_results = set()
combined_results.update(fetch_blocklist_de())
combined_results.update(fetch_greensnow())
combined_results.update(fetch_emergingthreats())
combined_results.update(fetch_cinsscore())
combined_results.update(fetch_cybercure())
combined_results.update(fetch_mirai_security())
combined_results.update(fetch_ipsum())
combined_results.update(fetch_home_nuug())
combined_results.update(fetch_threatview())
combined_results.update(fetch_myms())
```

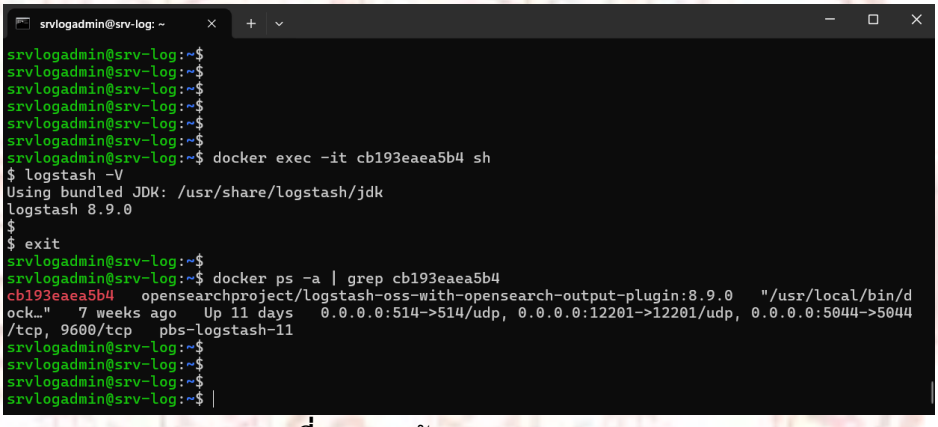
ภาพที่ 3-11 โค้ด Python ที่ใช้รวมข้อมูลจาก MISP Default Feeds

3.3.2.3.3 การจัดรูปแบบ Blacklist IP Address จาก MISP Default Feeds ที่ถูกรวมไว้ แล้วบันทึกลงไฟล์ YAML โดยโค้ดตามภาพที่ 3-12 จะดำเนินการจัดรูปแบบ Blacklist IP Address ให้พร้อมใช้งานกับ Logstash โดยใช้คำสั่ง formatted_results = [f"{ip}": blacklist-ip-address' for ip in sorted(combined_results)] เพื่อให้ผลลัพธ์อยู่ในรูปแบบ "IP": blacklist-ip-address และใช้ sorted(combined_results) เพื่อเรียงลำดับ IP Address จากน้อยไปมากก่อนที่จะนำมาแปลงเป็นข้อความ จากนั้นข้อมูลจะถูกเขียนลงไฟล์โดยกำหนดพาธไฟล์ output_file เป็น "/home/srvlogadmin/projects/logstash/config/dictionaries/misp-blacklist-ip-address.yml" เพื่อใช้ในการเทียบเคียงกับ Logstash ต่อไป

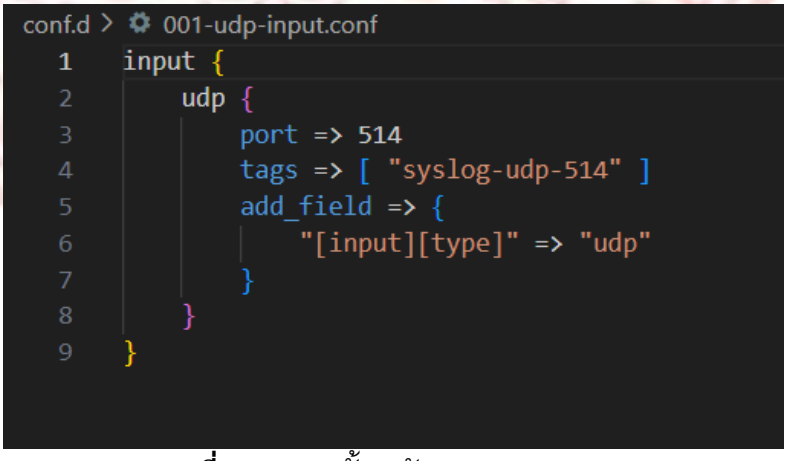
```python
# Format all IPs as "IP": backlist-ip-address
formatted_results = [f'"{ip}": blacklist-ip-address' for ip in sorted(combined_results)]

# Write to output file
output_file = "/home/srvlogadmin/projects/logstash/config/dictionaries/misp-blacklist-ip-address.yml"
try:
    with open(output_file, "w") as file:
        file.write("\n".join(formatted_results))
    print(f"Combined data has been saved to {output_file}")
except Exception as e:
    print(f"Error writing combined file: {e}")
```

ภาพที่ 3-12 โค้ด Python ที่ใช้จัดรูปแบบข้อมูลจาก MISP Default Feeds

3.3.2.4 Logstash ในงานวิจัยนี้ Logstash ทำหน้าที่รับ log จากอุปกรณ์ Firewall FortiGate 50E ผ่าน service port UDP 514 ตามที่แสดงในภาพที่ 3-13 โดยการตั้งค่าการรับ log ของ Logstash แสดงอยู่ในภาพที่ 3-14



ภาพที่ 3-13 หน้าจอ Logstash Version



ภาพที่ 3-14 การตั้งค่ารับ syslog UDP 514

จากนั้นทำการ filter ค่า destination IP address จาก log traffic โดยนำมาเปรียบเทียบกับไฟล์ misp-blacklist-ip-address.yml ซึ่งได้จากการดึงข้อมูลผ่าน Python ในตอนต้น หากพบว่า destination IP address อยู่ใน dataset จะติด tag "blacklist-ip-address" และนำข้อมูลนั้นไปอยู่ใน [threat][software][type] แต่หากไม่พบ จะติด tag "none-blacklist-ip-address" ตามที่แสดงใน ภาพที่ 3-15 จากนั้นส่งข้อมูลที่ผ่านการประมวลผลไปยัง OpenSearch เพื่อนำไปใช้ในการวิเคราะห์และแสดงผลต่อไป ตามที่แสดงใน ภาพที่ 3-16

conf.d > 016-fortigate-custom.conf

```
filter {
    if "fortigate" in [tags] {
        if [destination][ip] {
            translate {
                dictionary_path => [ "/usr/share/logstash/config/dictionaries/misp-blacklist-ip-address.yml" ]
                source => "[destination][ip]"
                target => "[threat][software][type]"
                fallback => "none-blacklist-ip-address"
            }
        }
    }
}
```

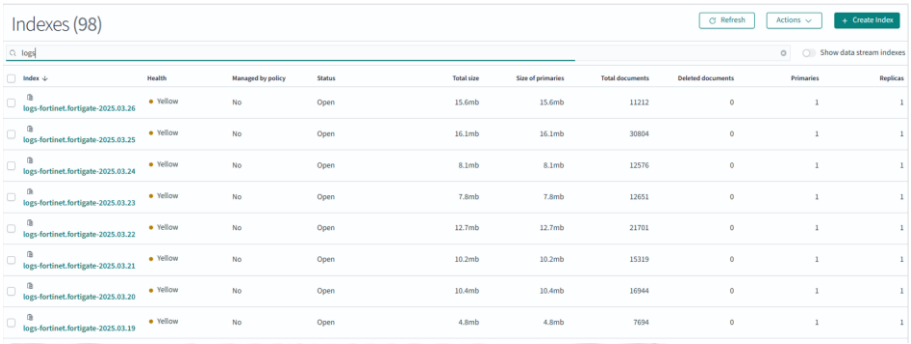
ภาพที่ 3-15 การตั้งค่า Filter สำหรับ Destination IP กับไฟล์ MISP Blacklist

conf.d > 101-opensearch-output.conf

```
output {
    if [input][type] == "udp" {
        if [data_stream][dataset] {
            opensearch {
                hosts => ["https://pbs-node-11:9200"]
                index => "%{[data_stream][type]}-%{[data_stream][dataset]}-%{+YYYY.MM.dd}"
                action => "create"
                user => "logstashwrite"
                password => "[redacted]"
                ssl => true
                ssl_certificate_verification => false
            }
        }
    # End if [input][type] == "udp"
    }
}
```

ภาพที่ 3-16 การตั้งค่าการส่งข้อมูลที่ผ่านการตรวจสอบไปยัง OpenSearch

3.3.2.5 OpenSearch ในงานวิจัยนี้ OpenSearch ทำหน้าที่เป็นเครื่องมือแสดงผลสำหรับการตรวจจับและแจ้งเตือนภัยไซเบอร์ โดยใช้ฐานข้อมูล MISP Threat Sharing ในการเปรียบเทียบ Blacklist IP Address จาก MISP Default Feeds กับ log traffic กระบวนการนี้ทำให้ข้อมูลพร้อมใช้งานโดยการติดแท็ก "blacklist-ip-address" และ "none-blacklist-ip-address" ในเนื้อหา log ผ่านหัวข้อ threat.software.type ซึ่งได้รับจาก Logstash และถูกนำเสนอผ่าน OpenSearch Dashboard โดยไฟล์บันทึกข้อมูลการจราจรเครือข่ายจะถูกเก็บใน indexes ดังแสดงในภาพที่ 3-17



ภาพที่ 3-17 ที่เก็บไฟล์บันทึกข้อมูลการจราจรเครือข่ายใน OpenSearch

3.4 ทดสอบการใช้งานของระบบและอุปกรณ์

3.4.1 ทดสอบการทำงานของฮาร์ดแวร์ โดยตรวจสอบว่าสามารถเชื่อมต่ออินเทอร์เน็ตได้และมีการส่ง Log Traffic ไปยัง Syslog Server อย่างถูกต้องตามที่กำหนด

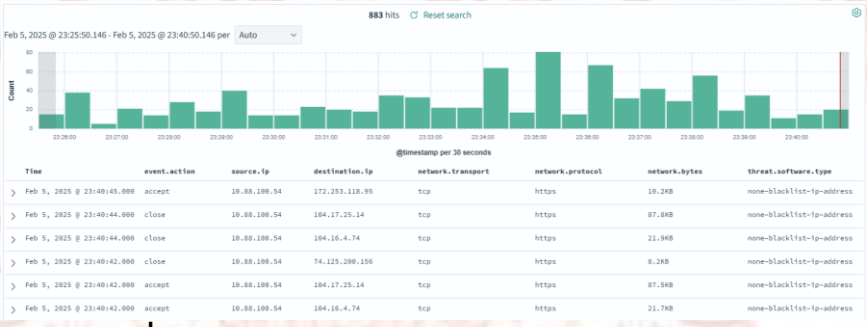
3.4.2 ทดสอบการทำงานของระบบ โดยตรวจสอบว่าสามารถแสดงผล Blacklist IP Address จากฐานข้อมูล MISP Default Feeds ได้

3.4.3 ทดสอบและเปรียบเทียบประสิทธิภาพของระบบ ในการตรวจสอบ Blacklist IP Address กับฐานข้อมูล MISP Default Feeds โดยใช้รายการ Blacklist IP Address ที่เตรียมไว้จำนวน 10 รายการ

บทที่ 4
ผลการดำเนินการวิจัย

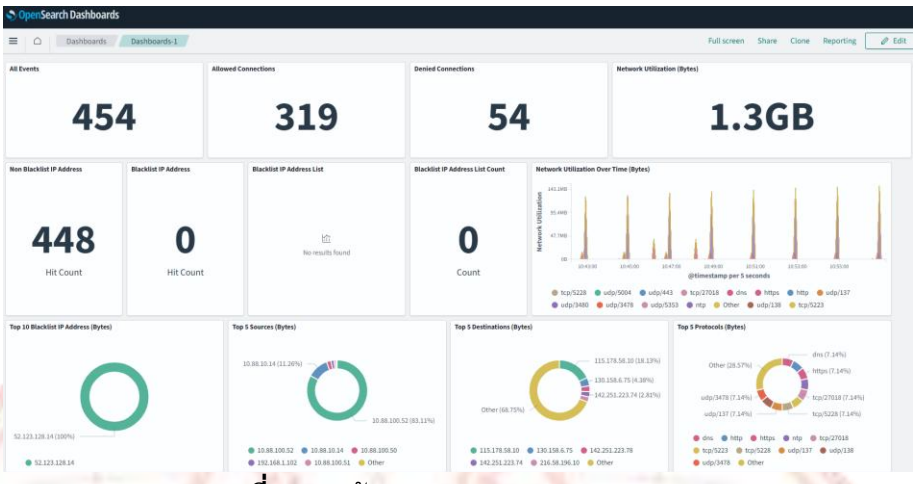
จากการดำเนินการวิจัย ศึกษา และพัฒนาระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing ระบบได้รับการออกแบบให้ดึงข้อมูล Blacklist IP Address จาก MISP Default Feeds เพื่อนำมาเปรียบเทียบกับ Destination IP Address ที่บันทึกไว้ใน log ของอุปกรณ์ Firewall FortiGate 50E หากพบว่าที่อยู่ IP ตรงกับข้อมูลในบัญชีดำ ระบบจะทำการแจ้งเตือนทันที

โดยค่าปกติในคอลัมน์ threat.software.type ของบันทึกข้อมูลการจราจรเครือข่ายจะแสดงเป็น "none-blacklist-ip-address" ดังแสดงในภาพที่ 4-1 และหากตรวจพบว่าอยู่ในรายการ Blacklist ค่าจะแสดงเป็น "blacklist-ip-address"



ภาพที่ 4-1 บันทึกข้อมูลการจราจรเครือข่ายบน OpenSearch

ทั้งนี้ OpenSearch Dashboard ดังแสดงในภาพที่ 4-2 จะแสดงข้อมูลสำคัญของบันทึกการจราจรเครือข่าย ซึ่งประกอบด้วยจำนวนเหตุการณ์ที่เกิดขึ้นทั้งหมด จำนวนการเชื่อมต่อที่ได้รับอนุญาตและไม่ได้รับอนุญาต ค่าการใช้งานเครือข่าย จำนวน Non Blacklist IP Address และ Blacklist IP Address ในการเชื่อมต่อ รายการ Blacklist IP Address ที่มีการเข้าใช้งาน จำนวนรายการ Blacklist IP Address ที่มีการเข้าใช้งาน ค่าการใช้งานเครือข่ายตามระยะเวลา รวมถึงการจัดอันดับ Blacklist IP Address 10 อันดับแรก ต้นทาง IP Address 5 อันดับแรก ปลายทาง IP Address 5 อันดับแรก และโปรโตคอล 5 อันดับแรก

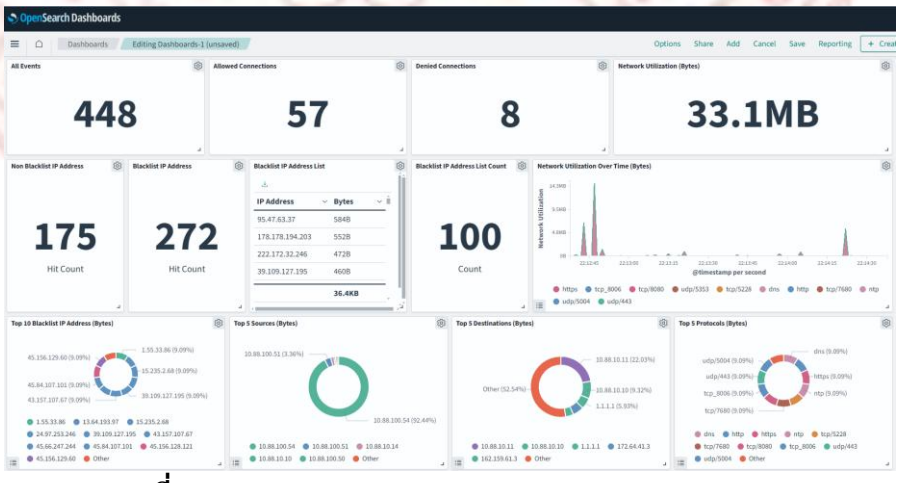


ภาพที่ 4-2 หน้าจอ OpenSearch Dashboard

ในขั้นตอนการทดสอบ ระบบได้ใช้ชุดข้อมูล Blacklist IP Address จากแหล่งข้อมูล IPsum, Myip.ms และ Blocklist.de อย่างละ 100 รายการ ดำเนินการทดสอบทั้งหมด 10 ครั้ง โดยในแต่ละครั้งจะทำการ สุ่ม (Random) ชุด IP Address ใหม่ เพื่อประเมินประสิทธิภาพของระบบในการ ตรวจจับและแจ้งเตือนภัยไซเบอร์ ทั้งนี้ หน้าตาของระบบในกรณีที่ ยังไม่พบ Blacklist IP Address แสดงดังภาพที่ 4-2 ทั้งนี้ ผลลัพธ์การตรวจสอบจะแสดงผ่าน Dashboard ที่พัฒนาขึ้น โดยผลการดำเนินการวิจัยแบ่งออกเป็น 4 ส่วนดังนี้

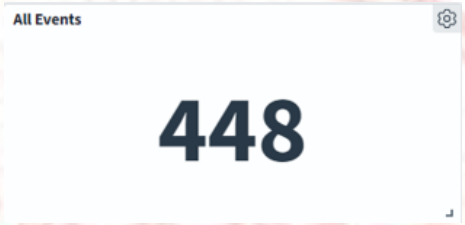
4.1 ผลการทดสอบชุดข้อมูล Blacklist IP Address ของ IPsum

การทดสอบดำเนินการโดยใช้ script สำหรับ telnet ไปยัง host จาก file list ที่สุ่มขึ้นมา (random) ผ่าน service port TCP 443 และตรวจสอบผลลัพธ์ผ่าน OpenSearch Dashboard



ภาพที่ 4-3 ผลการทดสอบ Blacklist IP Address ของ IPsum

ผลการทดสอบ Blacklist IP Address ของ IPsum ดังแสดงในภาพที่ 4-3 พบว่าระบบสามารถตรวจพบ 100 IP Address ที่ถูกสุ่มตรวจสอบครบถ้วน และมีองค์ประกอบของ Dashboard ดังนี้ จากภาพที่ 4-4 แสดงจำนวนเหตุการณ์ทั้งหมดที่เกิดขึ้นภายในระบบจำนวน 448 รายการ ขณะที่ ภาพที่ 4-5 และ ภาพที่ 4-6 แสดงการเชื่อมต่อที่ได้รับอนุญาตจำนวน 57 รายการ และการเชื่อมต่อที่ถูกปฏิเสธจำนวน 8 รายการ

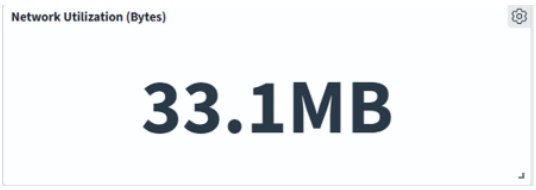


ภาพที่ 4-4 ผลการทดสอบ Blacklist IP Address ของ IPsum (All Events)

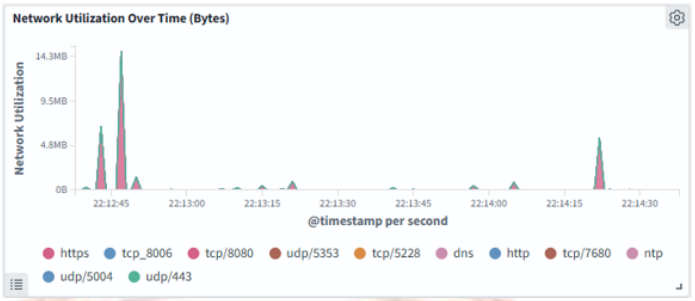
ภาพที่ 4-5 ผลการทดสอบ Blacklist IP Address ของ IPsum (Allowed Connections)

ภาพที่ 4-6 ผลการทดสอบ Blacklist IP Address ของ IPsum (Denied Connections)

ภาพที่ 4-7 แสดงปริมาณการใช้งานเครือข่ายรวมทั้งสิ้น 33.1 MB และภาพที่ 4-8 แสดงการใช้งานสูงสุดที่ประมาณ 14.3 MB ในช่วงเวลา 22:12 น.

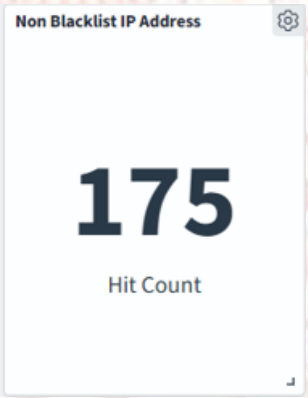


ภาพที่ 4-7 ผลการทดสอบ Blacklist IP Address ของ IPsum (Network Utilization)

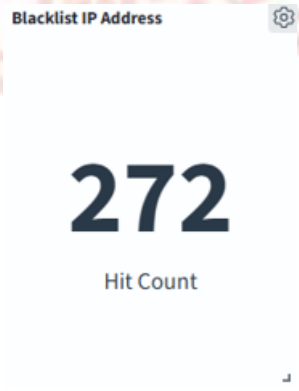


ภาพที่ 4-8 ผลการทดสอบ Blacklist IP Address ของ IPsum (Network Utilization Over Time)

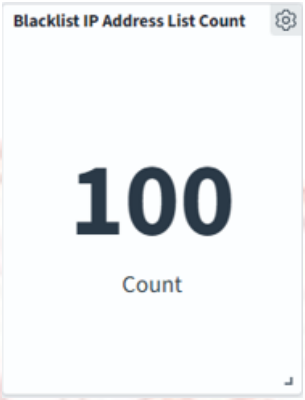
ภาพที่ 4-9 และ ภาพที่ 4-10 แสดงจำนวนการเชื่อมต่อจาก IP Address ที่ไม่อยู่ใน Blacklist จำนวน 175 ครั้ง และที่อยู่ใน Blacklist จำนวน 272 ครั้ง ภาพที่ 4-11 แสดงจำนวน IP Address ที่อยู่ในรายการ Blacklist ทั้งหมด 100 รายการ ภาพที่ 4-12 แสดงรายละเอียดของ IP Address ใน Blacklist พร้อมปริมาณข้อมูลรวม 36.4 KB และภาพที่ 4-13 แสดง IP Address ในอันดับสูงสุด 10 อันดับ โดยแต่ละรายการมีปริมาณข้อมูลเท่ากันที่ 9.09%



ภาพที่ 4-9 ผลการทดสอบ Blacklist IP Address ของ IPsum (Non Blacklist IP Address)

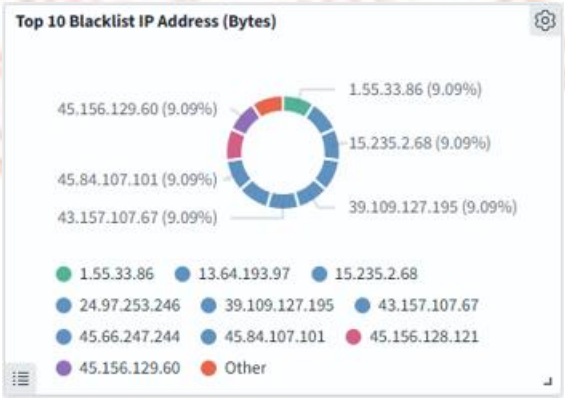


ภาพที่ 4-10 ผลการทดสอบ Blacklist IP Address ของ IPsum (Blacklist IP Address)



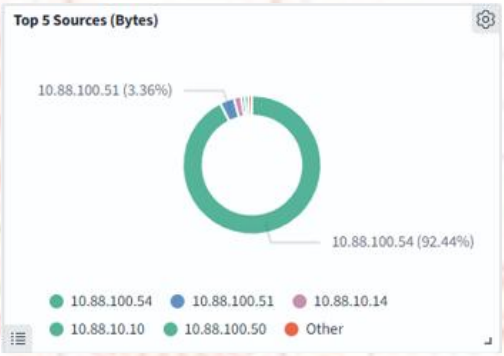
ภาพที่ 4-11 ผลการทดสอบ Blacklist IP Address ของ IPsum (Blacklist IP Address List Count)

ภาพที่ 4-12 ผลการทดสอบ Blacklist IP Address ของ IPsum (Blacklist IP Address List)

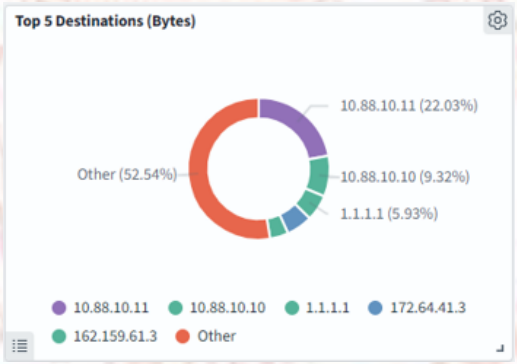


ภาพที่ 4-13 ผลการทดสอบ Blacklist IP Address ของ IPsum (Top 10 Blacklist IP Address)

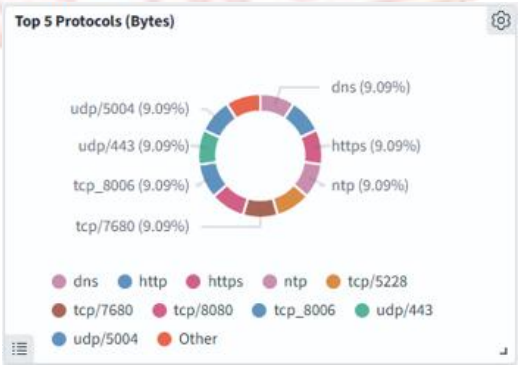
ภาพที่ 4-14 แสดงแหล่งที่มาของข้อมูล โดย IP Address 10.88.100.54 มีปริมาณข้อมูลมากที่สุด คิดเป็น 92.44% ของทั้งหมด ภาพที่ 4-15 แสดงปลายทางของข้อมูล โดย IP Address 10.88.10.11 มีปริมาณมากที่สุด คิดเป็น 22.03% รองลงมาคือ 10.88.10.10 คิดเป็น 9.32% และ 1.1.1.1 คิดเป็น 5.93% ส่วนปลายทางอื่นรวมกันคิดเป็น 52.54% ของข้อมูลทั้งหมด และภาพที่ 4-16 แสดงโปรโตคอลที่ใช้งานมากที่สุด 5 รายการ ได้แก่ dns, https, ntp, tcp/7680 และ tcp_8006 ซึ่งแต่ละรายการมีปริมาณข้อมูลเท่ากันคือ 9.09%



ภาพที่ 4-14 ผลการทดสอบ Blacklist IP Address ของ IPsum (Top 5 Sources)



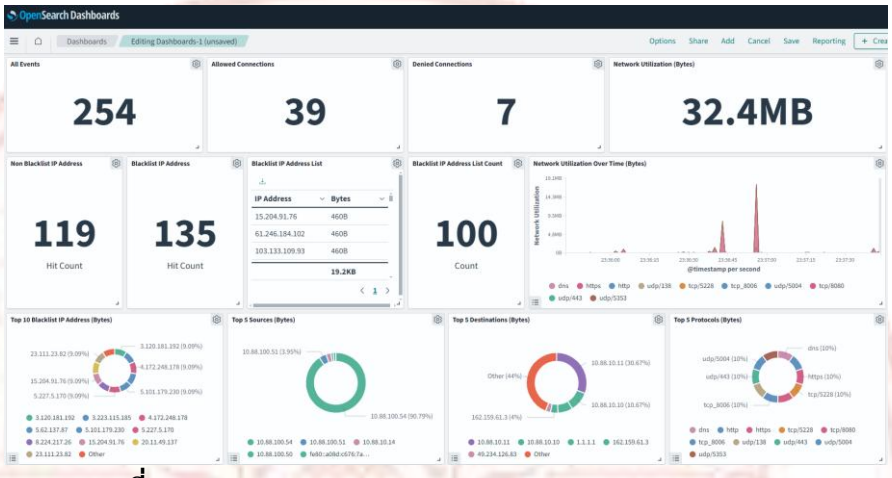
ภาพที่ 4-15 ผลการทดสอบ Blacklist IP Address ของ IPsum (Top 5 Destinations)



ภาพที่ 4-16 ผลการทดสอบ Blacklist IP Address ของ IPsum (Top 5 Protocols)

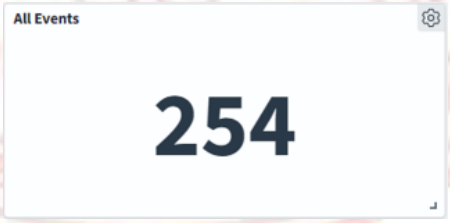
4.2 ผลการทดสอบชุดข้อมูล Blacklist IP Address ของ Myip.ms

การทดสอบดำเนินการโดยใช้ script สำหรับ telnet ไปยัง host จาก file list ที่สุ่มขึ้นมา (random) ผ่าน service port TCP 443 และตรวจสอบผลลัพธ์ผ่าน OpenSearch Dashboard



ภาพที่ 4-17 ผลการทดสอบ Blacklist IP Address ของ Myip.ms

ผลการทดสอบ Blacklist IP Address ของ Myip.ms ดังแสดงในภาพที่ 4-17 พบว่าระบบสามารถตรวจพบ 100 IP Address ที่ถูกสุ่มตรวจสอบครบถ้วน และมีองค์ประกอบของ Dashboard ดังนี้ ภาพที่ 4-18 แสดงจำนวนเหตุการณ์ทั้งหมดที่เกิดขึ้นภายในระบบจำนวน 254 รายการ ขณะที่ ภาพที่ 4-19 และภาพที่ 4-20 แสดงการเชื่อมต่อที่ได้รับอนุญาตจำนวน 39 รายการ และการเชื่อมต่อที่ถูกปฏิเสธจำนวน 7 รายการ

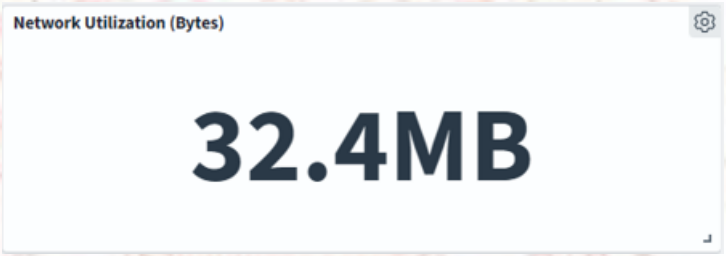


ภาพที่ 4-18 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (All Events)

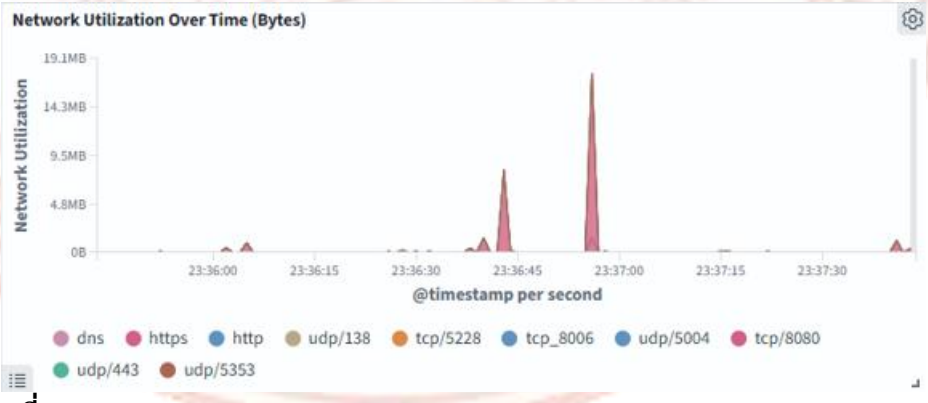
ภาพที่ 4-19 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Allowed Connections)

ภาพที่ 4-20 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Denied Connections)

ภาพที่ 4-21 แสดงปริมาณการใช้งานเครือข่ายรวมทั้งสิ้น 32.4 MB และ ภาพที่ 4-22 แสดงการใช้งานสูงสุดที่ประมาณ 19.1 MB ในช่วงเวลา 23:36 น.



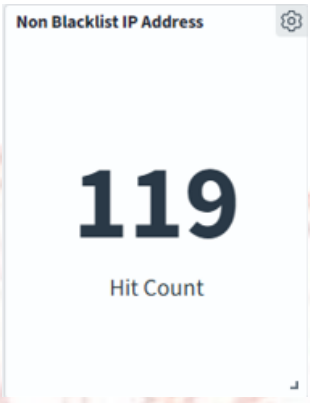
ภาพที่ 4-21 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Network Utilization)



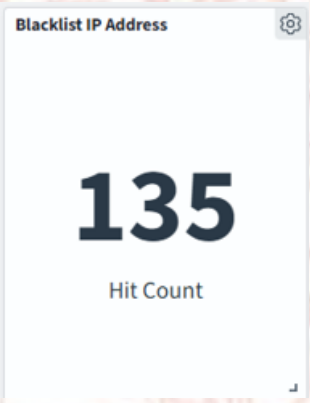
ภาพที่ 4-22 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Network Utilization Over Time)

ภาพที่ 4-23 และภาพที่ 4-24 แสดงจำนวนการเชื่อมต่อจาก IP Address ที่ไม่อยู่ใน Blacklist จำนวน 119 ครั้ง และที่อยู่ใน Blacklist จำนวน 135 ครั้ง ภาพที่ 4-25 แสดงจำนวน IP Address ที่อยู่ในรายการ Blacklist ทั้งหมด 100 รายการ ภาพที่ 4-26 แสดงรายละเอียดของ IP

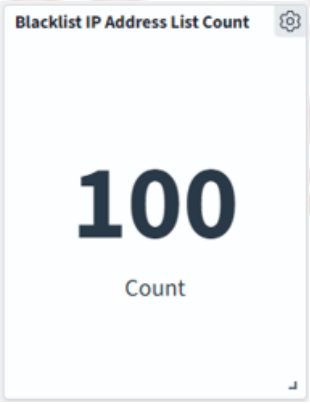
Address ใน Blacklist พร้อมปริมาณข้อมูลรวม 19.2 KB และภาพที่ 4-27 แสดง IP Address ใน อันดับสูงสุด 10 อันดับ โดยแต่ละรายการมีปริมาณข้อมูลเท่ากันที่ 9.09%



ภาพที่ 4-23 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Non Blacklist IP Address)

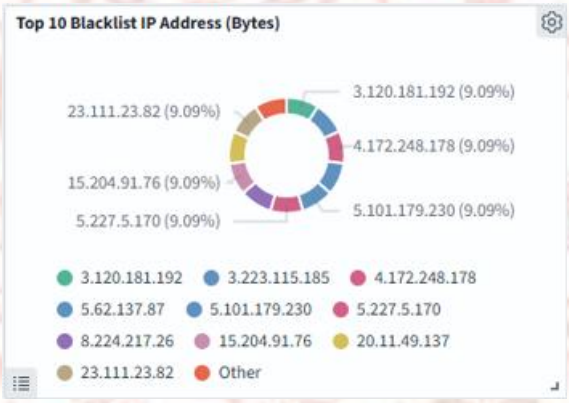


ภาพที่ 4-24 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Blacklist IP Address)



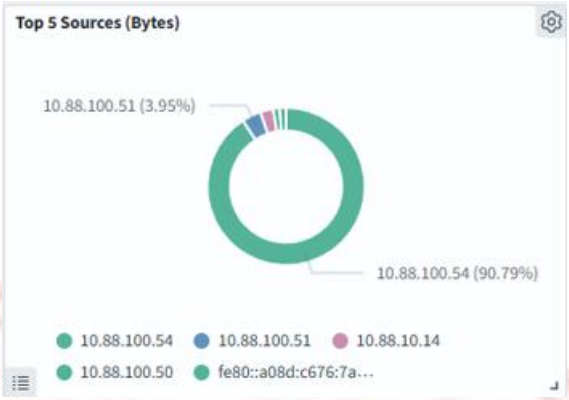
ภาพที่ 4-25 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Blacklist IP Address List Count)

ภาพที่ 4-26 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Blacklist IP Address List)

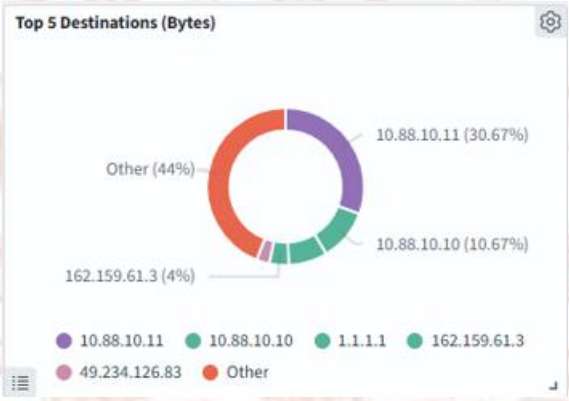


ภาพที่ 4-27 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Top 10 Blacklist IP Address)

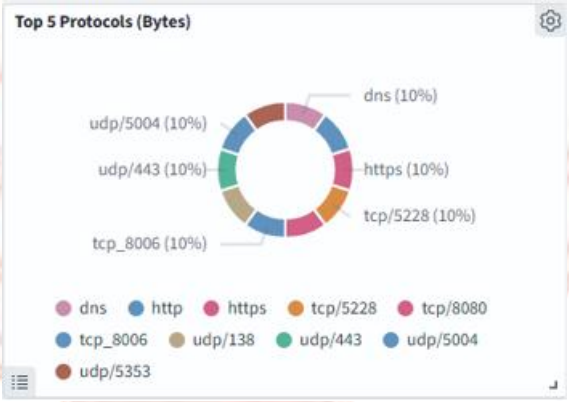
ภาพที่ 4-28 แสดงแหล่งที่มาของข้อมูล โดย IP Address 10.88.100.54 มีปริมาณข้อมูลมากที่สุด คิดเป็น 90.79% ของทั้งหมด ภาพที่ 4-29 แสดงปลายทางของข้อมูล โดย IP Address 10.88.10.11 มีปริมาณมากที่สุด คิดเป็น 30.67% รองลงมาคือ 10.88.10.10 คิดเป็น 10.67% และ 162.159.61.3 คิดเป็น 4% ส่วนปลายทางอื่นรวมกันคิดเป็น 44% ของข้อมูลทั้งหมด และภาพที่ 4-30 แสดงโปรโตคอลที่ใช้งานมากที่สุด 5 รายการ ได้แก่ udp/5004, udp/443, tcp_8006, tcp/7680 และ dns ซึ่งแต่ละรายการมีปริมาณข้อมูลเท่ากันคือ 10%



ภาพที่ 4-28 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Top 5 Sources)



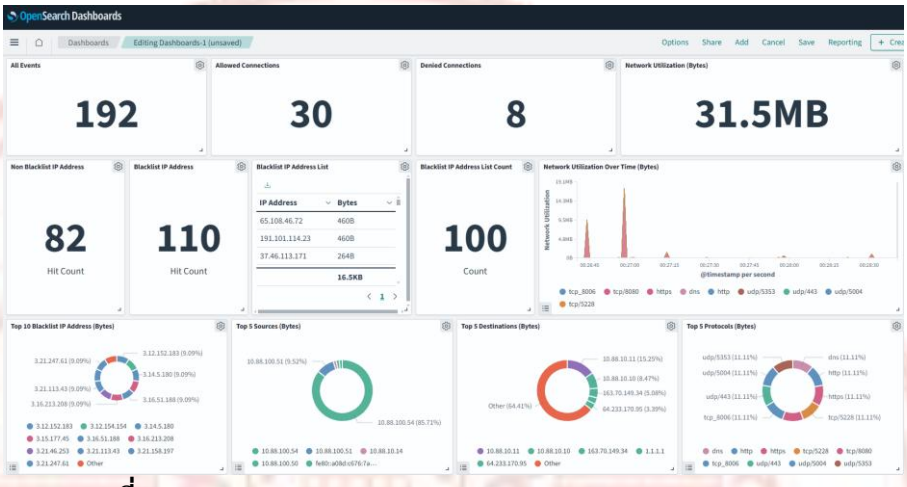
ภาพที่ 4-29 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Top 5 Destinations)



ภาพที่ 4-30 ผลการทดสอบ Blacklist IP Address ของ Myip.ms (Top 5 Protocols)

4.3 ผลการทดสอบชุดข้อมูล Blacklist IP Address ของ Blocklist.de

การทดสอบดำเนินการโดยใช้ script สำหรับ telnet ไปยัง host จาก file list ที่สุ่มขึ้นมา (random) ผ่าน service port TCP 443 และตรวจสอบผลลัพธ์ผ่าน OpenSearch Dashboard



ภาพที่ 4-31 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de

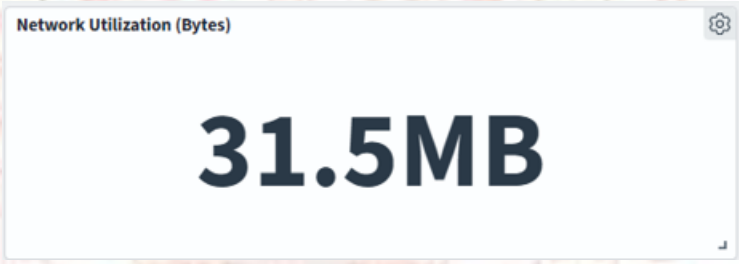
ผลการทดสอบ Blacklist IP Address ของ Blocklist.de ดังแสดงในภาพที่ 4-31 พบว่าระบบสามารถตรวจพบ 100 IP Address ที่ถูกสุ่มตรวจสอบครบถ้วน และมีองค์ประกอบของ Dashboard ดังนี้ ภาพที่ 4-32 แสดงจำนวนเหตุการณ์ทั้งหมดที่เกิดขึ้นภายในระบบจำนวน 192 รายการ ขณะที่ ภาพที่ 4-33 และภาพที่ 4-34 แสดงการเชื่อมต่อที่ได้รับอนุญาตจำนวน 30 รายการ และการเชื่อมต่อที่ถูกปฏิเสธจำนวน 8 รายการ

ภาพที่ 4-32 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (All Events)

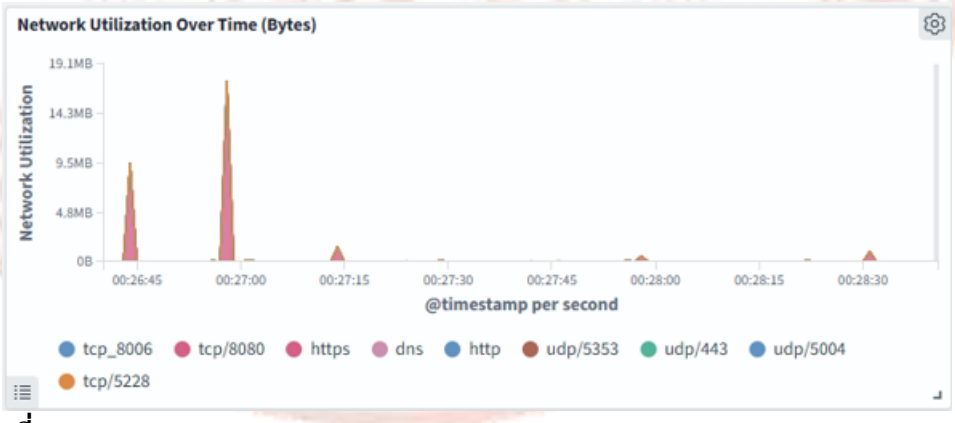
ภาพที่ 4-33 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Allowed Connections)

ภาพที่ 4-34 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Denied Connections)

ภาพที่ 4-35 แสดงปริมาณการใช้งานเครือข่ายรวมทั้งสิ้น 31.5 MB และ ภาพที่ 4-36 แสดงการใช้งานสูงสุดที่ประมาณ 19.0 MB ในช่วงเวลา 00:26 น.



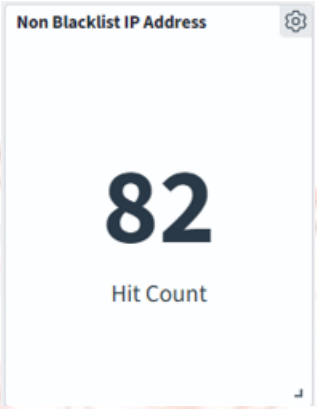
ภาพที่ 4-35 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Network Utilization)



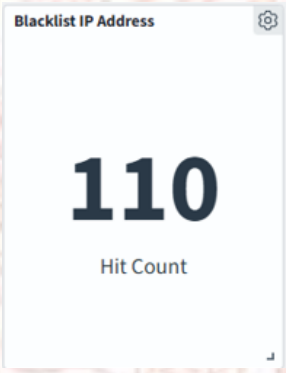
ภาพที่ 4-36 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Network Utilization Over Time)

ภาพที่ 4-37 และ ภาพที่ 4-38 แสดงจำนวนการเชื่อมต่อจาก IP Address ที่ไม่อยู่ใน Blacklist จำนวน 82 ครั้ง และที่อยู่ใน Blacklist จำนวน 110 ครั้ง ภาพที่ 4-39 แสดงจำนวน IP Address ที่อยู่ในรายการ Blacklist ทั้งหมด 100 รายการ ภาพที่ 4-40 แสดงรายละเอียดของ IP

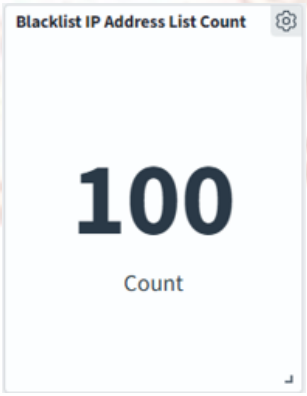
Address ใน Blacklist พร้อมปริมาณข้อมูลรวม 16.5 KB และ ภาพที่ 4-41 แสดง IP Address ในอันดับสูงสุด 10 อันดับ โดยแต่ละรายการมีปริมาณข้อมูลเท่ากันที่ 9.09%



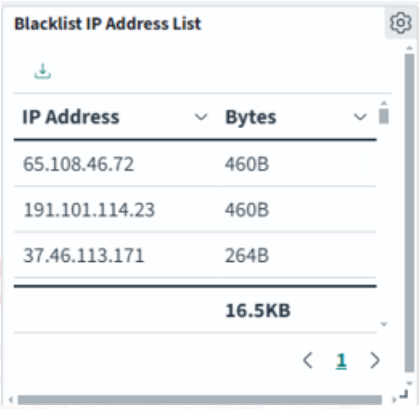
ภาพที่ 4-37 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Non Blacklist IP Address)



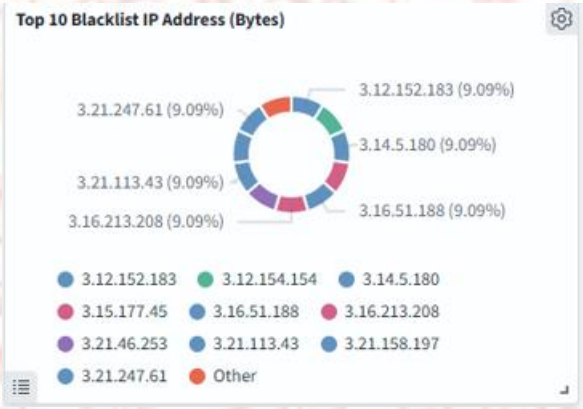
ภาพที่ 4-38 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Blacklist IP Address)



ภาพที่ 4-39 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Blacklist IP Address List Count)

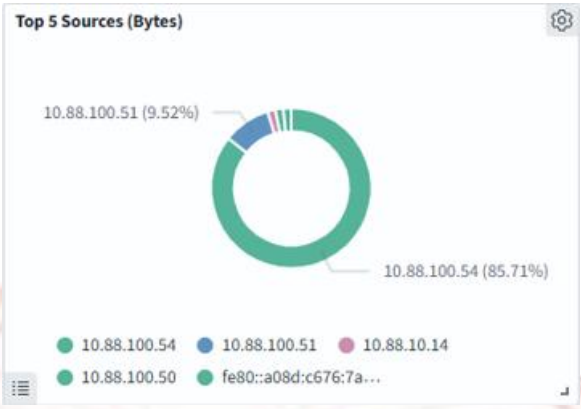


ภาพที่ 4-40 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Blacklist IP Address List)

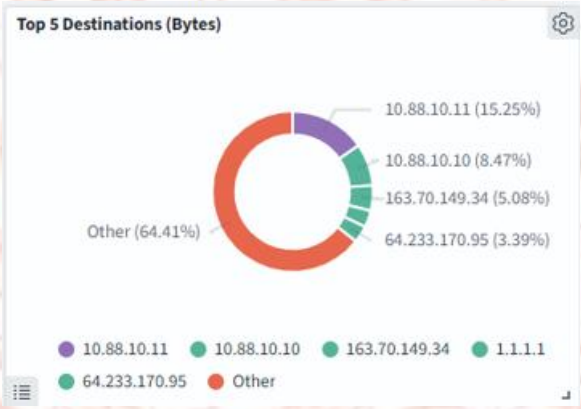


ภาพที่ 4-41 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Top 10 Blacklist IP Address)

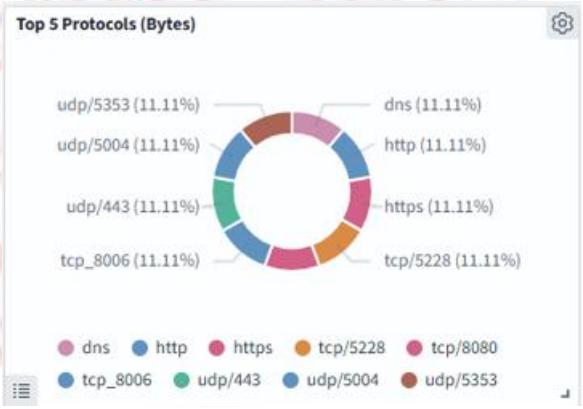
ภาพที่ 4-42 แสดงแหล่งที่มาของข้อมูล โดย IP Address 10.88.100.54 มีปริมาณข้อมูลมากที่สุด คิดเป็น 85.71% ของทั้งหมด ภาพที่ 4-29 แสดงปลายทางของข้อมูล โดย IP Address 10.88.10.11 มีปริมาณมากที่สุด คิดเป็น 15.25% รองลงมาคือ 10.88.10.10 คิดเป็น 8.47% 163.70.149.34 คิดเป็น 5.08% และ 64.233.170.95 คิดเป็น 3.39% ส่วนปลายทางอื่นรวมกันคิดเป็น 64.41% ของข้อมูลทั้งหมด และภาพที่ 4-30 แสดงโปรโตคอลที่ใช้งานมากที่สุด 5 รายการ ได้แก่ udp/5353, udp/5004, udp/443, tcp_8006, tcp/5228, http, https และ dns ซึ่งแต่ละรายการมีปริมาณข้อมูลเท่ากันคือ 11.11%



ภาพที่ 4-42 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Top 5 Sources)



ภาพที่ 4-43 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Top 5 Destinations)



ภาพที่ 4-44 ผลการทดสอบ Blacklist IP Address ของ Blocklist.de (Top 5 Protocols)

4.4 ผลการเปรียบเทียบความสามารถในการตรวจจับ Blacklist IP Address

ผลการทดสอบทั้งหมดแสดงในตารางที่ 4-1 โดยดำเนินการทดสอบทั้งหมด 10 ครั้ง แต่ละครั้งมีการสุ่ม 100 IP และในทุกครั้งระบบสามารถตรวจจับ blacklist IP ได้ครบถ้วน ตัวอย่างผลการทดสอบแสดงดังภาพที่ 4-3, 4-17 และ 4-31

ตารางที่ 4-1 การเปรียบเทียบผลลัพธ์จากการทดสอบ Blacklist IP Address

No.	สุ่มตรวจสอบ Blacklist IP Address จำนวน 100 IP	จำนวนที่ตรวจพบในระบบ		
		IPsum	Myip.ms	Blocklist.de
1	สุ่มตรวจสอบ ครั้งที่ 1	100	100	100
2	สุ่มตรวจสอบ ครั้งที่ 2	100	100	100
3	สุ่มตรวจสอบ ครั้งที่ 3	100	100	100
4	สุ่มตรวจสอบ ครั้งที่ 4	100	100	100
5	สุ่มตรวจสอบ ครั้งที่ 5	100	100	100
6	สุ่มตรวจสอบ ครั้งที่ 6	100	100	100
7	สุ่มตรวจสอบ ครั้งที่ 7	100	100	100
8	สุ่มตรวจสอบ ครั้งที่ 8	100	100	100
9	สุ่มตรวจสอบ ครั้งที่ 9	100	100	100
10	สุ่มตรวจสอบ ครั้งที่ 10	100	100	100

บทที่ 5
สรุปผลและข้อเสนอแนะ

การค้นคว้าอิสระครั้งนี้มุ่งเน้นที่การพัฒนาระบบ ตรวจจับและแจ้งเตือนภัยไซเบอร์ โดยอาศัยฐานข้อมูลจาก MISP Threat Sharing ซึ่งระบบทำงานโดยการเปรียบเทียบ Blacklist IP Address กับฐานข้อมูล MISP Default Feeds เพื่อแจ้งเตือนทันทีเมื่อมีการเรียกใช้งาน Blacklist IP Address จากเครือข่ายภายในองค์กร ช่วยให้สามารถตรวจสอบและป้องกันความเสี่ยงที่อาจก่อให้เกิดภัยคุกคามทางไซเบอร์ก่อนที่จะส่งผลกระทบต่อองค์กร

ในการทดสอบระบบ ได้ใช้ชุดข้อมูล Blacklist IP Address จากแหล่งข้อมูล IPsum, Myip.ms และ Blocklist.de โดยทำการทดสอบทั้งหมด 10 ครั้ง ในแต่ละครั้งจะทำการ สุ่ม (Random) ชุด IP Address ใหม่ จำนวน 100 รายการ เพื่อประเมิน ประสิทธิภาพในการตรวจจับและแจ้งเตือนภัยไซเบอร์ ของระบบ พบข้อสรุปและข้อเสนอแนะในการพัฒนาระบบเพิ่มเติม ดังนี้

5.1 สรุปผล

ผลการทดสอบการตรวจจับและแจ้งเตือนภัยไซเบอร์โดยอาศัยฐานข้อมูลจาก MISP Threat Sharing ซึ่งทำการเปรียบเทียบ Blacklist IP Address กับฐานข้อมูล MISP Default Feeds พบว่าระบบสามารถตรวจจับ IP Address จากแหล่งข้อมูล IPsum, Myip.ms และ Blocklist.de ได้ครบถ้วนทุกครั้ง (ตรวจพบครบ 100 IP ในการทดสอบทั้ง 10 ครั้ง) แสดงถึงความแม่นยำในการตรวจจับที่สูงถึง 100% พร้อมทั้งแสดงถึงความเสถียรของระบบในการประมวลผลและแจ้งเตือนภัยไซเบอร์ได้อย่างมีประสิทธิภาพ ระบบนี้จะช่วยให้ผู้ดูแลระบบขององค์กรขนาดเล็กที่มีงบประมาณจำกัด สามารถรับทราบความผิดปกติของระบบเครือข่ายได้อย่างรวดเร็วจากการตรวจจับและแจ้งเตือนภัยไซเบอร์ที่เกิดจาก IP Address ที่เป็นอันตราย อีกทั้งยังสามารถตรวจสอบข้อมูลย้อนหลังในแต่ละช่วงเวลาได้อย่างสะดวกผ่าน Dashboard ซึ่งช่วยเพิ่มประสิทธิภาพในการติดตามและจัดการกับภัยคุกคามได้

5.2 ข้อจำกัดและปัญหาของการพัฒนา

5.2.1 ระบบนี้สามารถตรวจสอบได้เฉพาะ IPv4 เท่านั้น ไม่สามารถตรวจสอบ IPv6 และ Domain Name ได้

5.2.2 เนื่องจาก Log Traffic ที่ได้รับไม่เหมือนกับการทำ Sniffer Packet จึงทำให้การประยุกต์ใช้งานร่วมกับ MISP Threat Sharing มีข้อจำกัดในการนำข้อมูลมาใช้งานได้ไม่ครบถ้วน

5.3 ข้อเสนอแนะ

งานวิจัยนี้สามารถนำไปประยุกต์ใช้กับอุปกรณ์เครือข่ายอื่น ๆ ได้ เพียงแค่มีอุปกรณ์ที่สามารถส่ง Log Traffic ออกมาได้ โดยไม่จำเป็นต้องจำกัดการใช้งานเฉพาะกับ Firewall FortiGate 50E เท่านั้น นอกจากนี้ หากต้องการใช้งาน MISP Threat Sharing ในรูปแบบ Agent-less ให้มีประสิทธิภาพมากยิ่งขึ้น ควรใช้ Log แบบ Mirror Packet เพื่อเพิ่มความแม่นยำในการตรวจจับและแจ้งเตือนภัยไซเบอร์

บรรณานุกรม

1. “Firewall between Next Generation Firewall.” cyfence.com. (n.d.): https://www.cyfence.com/article/what-is-the-difference-between-firewall-and-next-generation-firewall/, Accessed 8 Feb. 2025.
2. “MISP Threat Sharing.” misp-project.org. (n.d.): https://www.misp-project.org/, Accessed 8 Feb. 2025.
3. “Logstash.” elastic.co. (n.d.): https://www.elastic.co/logstash, Accessed 8 Feb. 2025.
4. “OpenSearch.” opensearch.org. (n.d.): https://opensearch.org, Accessed 8 Feb. 2025.
5. “IPsum.” GitHub – stamparm. (n.d.): https://github.com/stamparm/ipsum/tree/master, Accessed 8 Feb. 2025.
6. “Myip.ms.” myip.ms. (2024): https://myip.ms/info/about, Accessed 8 Feb. 2025.
7. “Blocklist.de.” blocklist.de. (2024): https://www.blocklist.de/en/index.html, Accessed 8 Feb. 2025.
8. “MISP Default Feeds.” misp-project.org. (n.d.): https://www.misp-project.org/feeds/?utm_source=chatgpt.com/, Accessed 8 Feb. 2025.
9. “Header & Body Fields Log Firewall FortiGate.” fortinetweb.s3.amazonaws.com. (n.d.): https://fortinetweb.s3.amazonaws.com/docs.fortinet.com/v2/attachments/44936b5c-1a02-11e9-9685-f8bc1258b856/FortiOS_LogReference_v5.2.1.pdf, Accessed 8 Feb. 2025.
10. “Logstash Pipeline.” elastic.co. (n.d.): https://www.elastic.co/guide/en/logstash/current/pipeline.html, Accessed 8 Feb. 2025.
11. “OpenSearch Dashboards.” opensearch.org. (n.d.): https://opensearch.org/docs/latest/dashboards/, Accessed 8 Feb. 2025.
12. “Firewall FortiGate 50E.” brasiline.com.br. (n.d.): https://brasiline.com.br/wp-content/uploads/2021/12/FortiGate_FortiWiFi_50E_Series.pdf, Accessed 8 Feb. 2025.
13. Amori, F., Antonelli, S., Ciaschini, V., Falabella, A., Fattibene, E., Fornari, F., Lattanzio, D., Michelotto, D., and Morganti, L. “General purpose data streaming

platform for log analysis, anomaly detection and security protection.” EPJ Web of Conferences. 295 (2024): 1–7.

14. Poongkuyil, M., Shalinie, S., and Stanly, H. “Online Log Analysis (OLA) for Malicious User Activities.” Proceedings of the 2023 2nd International Conference on Paradigm Shifts in Communications, Embedded Systems, Machine Learning and Signal Processing (PCEMS). IEEE. (2023): 1–6.
15. Stoleriu, R., Puncioiu, A., and Bica, I. “Cyber Attacks Detection Using Open Source ELK Stack.” Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE. (2021): 1–6.
16. Hata, M. B. M., Darus, M. Y. B., Shafiee, M. Z. A. B., Petrus, E., and Jamian, Y. A. “A Log Aggregation Design Criteria for Robust SIEM (Security Information and Event Management) in Enhancing Threat Detection.” Proceedings of the 2023 8th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE). IEEE. (2023): 1–6.
17. Fernandes, R., Pinto, P., and Pinto, A. “Controlled and Secure Sharing of Classified Threat Intelligence between Multiple Entities.” Proceedings of the 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). IEEE. (2021): 525–530.
18. George, A. T., Neve, H. R., and Muraleedharan, N. “A Trust Score Calculation Approach for Zero Trust Access System.” Proceedings of the 2023 IEEE 20th India Council International Conference (INDICON). IEEE. (2023): 392–397.
19. Marinho, R. and Filho, R. H. “A Framework Proposal for Early Cyber Threat Identification and Profiling.” Proceedings of the 2023 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE. (2023): 1–7.
20. Umar, I. A., Salami, A. F., Ismaila, I. D., and Adeshina, S. A. “Designing a Unified Threat Management System for Home Networks.” Proceedings of the 2021 International Conference on Computer, Control and Communication Engineering (IC4E). IEEE. (2021): 80930–8094

ประวัติผู้เขียน

ชื่อ นายวุฒิชัย อ่อนสำลี

ชื่อการค้นคว้าอิสระ ระบบตรวจจับและแจ้งเตือนภัยไซเบอร์โดยใช้ฐานข้อมูล MISP Threat Sharing

สาขาวิชา การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประวัติ

ประวัติการศึกษา

สำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยบูรพา

ประวัติการทำงาน

พ.ศ. 2558 – 2566 ทำงาน โรงพยาบาลจุฬาลงกรณ์ สภากาชาดไทย

ตำแหน่ง เจ้าหน้าที่ระบบงานคอมพิวเตอร์

พ.ศ. 2566 – 2568 ทำงาน บริษัท กรีนไลน์ ซินเนอร์จี้ จำกัด

ตำแหน่ง วิศวกรเครือข่ายอาวุโส

พ.ศ. 2568 – ปัจจุบัน ทำงาน บริษัท ไอเอสเอส เรสโซลูชั่น จำกัด

ตำแหน่ง วิศวกรโซลูชัน