ระบบบริหารข้อมูลและเหตุการณ์สำหรับศูนย์ปฏิบัติการรักษาความปลอดภัย

SECURITY INFORMATION AND EVENT MANAGEMENT FOR SECURITY OPERATION CENTER

ศวีระ อภินทนาพงศ์
อนุรักษ์ จันนาวัน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบบริหารข้อมูลและเหตุการณ์สำหรับศูนย์ปฏิบัติการรักษาความปลอดภัย

SECURITY INFORMATION AND EVENT MANAGEMENT FOR SECURITY OPERATION CENTER

ศวีระ อภินทนาพงศ์
อนุรักษ์ จันนาวัน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ปริญญานิพนธ์ปีการศึกษา 2560

ภาค วิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบบริหารข้อมูลและเหตุการณ์สำหรับศูนย์ปฏิบัติการรักษาความปลอดภัย

SECURITY INFORMATION AND EVENT MANAGEMENT FOR SECURITY OPERATION CENTER

ผู้จัดทำ

นายศวีระ อภินทนาพงศ์ 57011229

นายอนุรักษ์ จันนาวัน 57011470

อาจารย์ที่ปรึกษา

(ผศ.อัครเดช วัชระภูพงษ์)

ระบบบริหารข้อมูลและเหตุการณ์
สำหรับศูนย์ปฏิบัติการรักษาความปลอดภัย

นายศวีระ อภินทนาพงศ์ 57011229

นายอนุรักษ์ จันนาวัน 57011470

ผศ.อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2560

บทคัดย่อ

ความปลอดภัยทางไซเบอร์เป็นส่วนสำคัญอย่างมากในเทคโนโลยีสารสนเทศเนื่องจากข้อมูลนั้นได้กลายเป็นสิ่งที่มีค่ามากที่สุดสำหรับทางเทคโนโลยีสารสนเทศ เพราะการโจมตีทางไซเบอร์ที่กำลังเพิ่มขึ้นอย่างต่อเนื่องทุกวัน อย่างเช่น การโจมตีเชิงปริมาณที่ก่อให้เกิดปัญหาทางเทคโนโลยีสารสนเทศอย่างมาก ดังนั้นต้องมีการจัดการอย่างไรจึงจะป้องกันความเสียหายที่เกิดขึ้นจากการโจมตีเหล่านี้ได้

ในโลกของการรักษาความปลอดภัยข้อมูลได้มีการนำ Security Information and Event Management (SIEM) หรือระบบจัดเก็บและวิเคราะห์ข้อมูลความปลอดภัยของระบบเครือข่าย ไปใช้จัดการและวิเคราะห์การโจมตีเชิงปริมาณเหล่านี้แต่ปัญหาอย่างหนึ่งของ SIEM คือการใช้งานที่ซับซ้อนสำหรับผู้นักวิเคราะห์ความปลอดภัย โปรเจคนี้จึงมีแนวคิดในการเพิ่มความสะดวกสบายในการใช้งาน SIEM แก่นักวิเคราะห์ความปลอดภัยทั่วๆไป

เพื่อที่จะแก้ปัญหากับการโจมตีขนาดใหญ่เหล่านี้จึงจำเป็นต้องมีการใช้ซอฟต์แวร์ในการจัดการ ซึ่งในโปรเจคนี้ได้ใช้ SIEM ลิขสิทธิ์ที่มีไลเซนส์แบบโอเพนซอร์ส (Open Source Software) ที่ชื่อว่า Metron มาใช้ในการจัดการข้อมูลและนำมาปรับปรุงในส่วนของการจัดการ Log เพื่อให้ใช้งานได้ง่ายขึ้นสำหรับนักวิเคราะห์ความปลอดภัย

SECURITY INFORMATION AND EVENT MANAGEMENT FOR SECURITY OPERATION CENTER

Mr.Saweera	Apintanapong	57011229
Mr.Anurak	Jannawan	5701470
Asst.Prof.Akkradach	Watcharapupong	Advisor

Academic 2017

Abstract

Cybersecurity is important in the field of information technology. The information become information technology's the most valuable. Cyber attacks are increasing everyday especially quantitative attack so how to prevent these damage that cause by the attack.

In the world of information security, Security information and Event Management (SIEM) is used to deal and analyze these quantitative attack but one of the problem with SIEM is complex for normal user. The Idea of this project is to improve SIEM user's convenient.

To deal with these massive network attacking is using of some management software. This project use Open Source software called "Metron" to improve log management that parse raw data to suitable message appropriate for security analyst.

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ เนื่องจากได้รับคำแนะนำ คำปรึกษา และคำชี้แจงในการดำเนินงาน แนวทางแก้ไขปัญหาที่เกิดขึ้นอย่างต่อเนื่องจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์อัครเดช วัชระภูพงษ์ ทางคณะผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษาเป็นอย่างสูง

ขอขอบคุณเป็นอย่างสูงสำหรับแรงบันดาลใจและคำแนะนำจากพี่ที่ปรึกษาที่คอยให้คำแนะนำเรื่องความรู้ที่ใช้เกี่ยวกับระบบที่นำมาพัฒนา ขอบคุณสำหรับซอฟแวร์จากทาง hortonworks ที่ทำ open source project สำหรับเครื่องมือนี้ ขอบคุณสำหรับห้องวิจัย ISAG ที่เอื้อเฟื้อสถาที่ในการดำเนินงาน

ศวีระ อภินทนาพงศ์

อนุรักษ์ จันนาวัน

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูป **หน้า**

สารบัญตาราง

ตาราง **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ในปัจจุบันการรักษาความปลอดภัยในระบบสารสนเทศถือเป็นสิ่งสำคัญอย่างมากในระบบธุรกิจที่ไม่สามารถมองข้ามไปได้ ด้วยการโจมตีที่เป็นปัญหาเชิงปริมาณที่เกิดขึ้นในปัจจุบัน ทำให้องค์กรต่างๆหันมาลงทุนทางด้านความปลอดภัยระบบเครือข่ายมากยิ่งขึ้นเพื่อไม่ให้เกิดปัญหากับระบบเครือขายขององค์กร เพื่อรับมือกับรูปแบบการโจมตีที่กล่าวมา เทคโนโลยีรักษาความปลอดภัยต่างๆ หลายรูปแบบก็ได้ถูกพัฒนาและนำมาใช้งานในระบบเครือข่ายองค์กร ไม่ว่าจะเป็น Next Generation Firewall, IPS, Email Gateway, Anti-Virus และอื่นๆ แต่การโจมตีจากผู้ประสงค์ร้ายก็ยังหลุดรอดมาได้โดยตลอดในปัจจุบันเป็นที่มาของ Security Information and Event Management (SIEM) ที่เป็นอุปกรณ์หลักในการตรวจจับและยับยั้งการโจมตีให้ได้มากที่สุด โดย SIEM ต้องมีความจำเป็นที่ตอบสนองความต้องการของเจ้าหน้าที่วิเคราะห์ความปลอดภัยว่าจะตรวจสอบข้อมูลอะไรและให้ออกมาในรูปแบบไหนที่เหมาะแก่การใช้งานกับเจ้าหน้าที่วิเคราะห์ความปลอดภัยให้มากที่สุด

1.2 แนวคิดของโครงงาน

จากปัญหาที่เกิดขึ้น อย่างเช่นการโจมตีทางไซเบอร์ซึ่งเป็นปัญหาทั่วไปของระบบเครือข่ายที่ก่อให้เกิดปัญหาอย่างบ่อยครั้ง เป็นปัญหาเชิงปริมาณที่ยากต่อการจัดการ ซึ่งในส่วนนี้ SIEM นั้นสามารถเข้ามาจัดการในส่วนนี้ได้ โดย SIEM สามารถวิเคราะห์ได้ว่า Log มาจากผู้ที่น่าเชื่อถือหรือไม่ หรือ Log นั้นมาในปริมาณมากหรือน้อยมากเพียงใด ทำให้สะดวกสบายแก่เจ้าหน้าที่วิเคราะห์ความปลอดภัยในการวิเคราะห์ Log และสามารถตรวจสอบ Log การโจมตีรูปแบบอื่นๆที่อาจจะหลุดรอดมาจากอุปกรณ์รักษาความปลอดภัยแต่ละชนิดมาได้ โดยการทำวิเคราะห์สหสัมพันธ์ (Correlation) เพื่อค้นหาพฤติกรรมการโจมตีระบบเครือข่ายที่กำลังเกิดขึ้น และทำการแจ้งเตือนเจ้าหน้าที่วิเคราะห์ความปลอดภัยแบบ Real-time เพื่อทำการหยุดยั้งเหตุการณ์เหล่านั้นให้ได้ทันเวลา

ด้วยเทคโนโลยีในปัจจุบันทำให้ SIEM นั้นมีพัฒนาการไปไกลมาก และมีคุณสมบัติโดดเด่นหลายอย่างและราคาแพงเป็นอย่างมากและค่อนข้างใช้งานยากสำหรับผู้ใช้งานทั่วไปโดย โครงงานนี้จะเป็นการทำ SIEM ที่ให้เหมาะกับผู้ใช้งานทั่วไปให้มากที่สุดโดยไม่กระทบกับความสามารถของ SIEM

1.3 เป้าหมายของโครงงาน

นำ Framework ของ Security Information and Event Management หรือ SIEM ซึ่งมีความสามารถในการจัดการ Log เบื้องต้นเพื่อที่สามารถตรวจสอบได้ว่าข้อมูลประเภทไหนเป็นอย่างไรและนำ Framework นั้นมาดัดแปลงให้ใช้งานได้สะดวกขึ้น โดย SIEM ต้องประมวลผลแบบ Real-Time ส่วนของรูปแบบโครงสร้างของข้อมูลที่มีจะถูกจัดให้อยู่ในรูปแบบดังต่อไปนี้

1) มีเหตุการณ์อะไรเกิดขึ้น เช่น Action Threat Priority
2) มาจากใคร เช่น Source IP, Source Port, Destination IP, Destination Port
3) มาจากที่ไหน เช่น Hostname, Hostname Alias , URL,
4) เวลาของการเกิดเหตการณ์ เช่น generated, persisted
5) อื่นๆ เช่น Protocol, Attack Type, Raw Logs

1.4 วัตถุประสงค์ของโครงงาน

1) เพื่อให้เจ้าหน้าที่วิเคราะห์ความปลอดภัยมีการทำงานที่ไม่ต้องลงไปทำการปรับแต่งในระดับล่างของ Apache Metron
2) เพื่อลดระยะเวลาการติดตั้ง Apache Metron และไม่ต้องศึกษาการติดตั้ง Apache Metron
3) เพื่อให้เจ้าหน้าที่วิเคราะห์ความปลอดภัยใช้งาน SIEM ได้อย่างสะดวกขึ้น

1.5 สิ่งที่คาดว่าจะได้รับ

1) เพื่อศึกษาเทคโนโลยีที่เกี่ยวกับ SIEM ในปัจจุบัน
2) เพื่อพัฒนาแนวทางของ SIEM ให้ก้าวหน้ายิ่งขึ้น
3) เพื่อช่วยแก้ปัญหาเกี่ยวข้องกับการติดตั้ง SIEM จาก Apache Metron
4) เพื่อแก้การทำงานของ Apache Metron ในขั้นตอนการทำ Correlation ให้ดีขึ้น
5) เข้าใจเทคโนโลยีที่เป็นส่วนประกอบของ Apache Metron

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Security Information and Event Management (SIEM) คืออะไร

Security Information and Event Management (SIEM) เป็นชุด software ที่ให้บริการเกี่ยวกับการจัดการ Security Information ร่วมกับการจัดการ Security Event มีการจัดการแบบ real-time analysis ของการแจ้งเตือนด้าน Security ที่ส่งมาจาก applications และ network hardware อีกทั้ง SIEM นั้นยังมีการให้ทำงานเกี่ยวกับ log security data แล้ว SIEM ยังมีการสร้างรายงานผล (Report) ของ Security Information และ Security Event ด้วย

SIEM นั้นมี Capabilities และ Components หลักดังต่อไปนี้

- Data aggregation

  การจัดการ log รวมข้อมูลจากหลายแหล่งรวมทั้ง network, security, server, database, applications ให้ความสามารถในการรวบรวมข้อมูลที่ได้รับการตรวจสอบเพื่อช่วยหลีกเลี่ยงเหตุการณ์สำคัญที่ขาดหายไป

- Correlation

  การรวม attribute ทั่วไปและเชื่อมโยง event เข้าด้วยกันเป็นกลุ่ม เทคโนโลยีนี้ให้ความสามารถในการดำเนินการต่างๆของเทคนิคความสัมพันธ์เพื่อรวมแหล่งข้อมูลต่างๆเพื่อที่จะทำให้ข้อมูลเป็นข้อมูลที่เป็นประโยชน์ โดยปกติ correlation จะเป็น function ของส่วนแนวทางของการจัดการความปลอดภัย SIEM เต็มรูปแบบ

- Alerting

  ระบบที่วิเคราะห์ correlation event ได้อัตโนมัติแล้วส่งการแจ้งเตือน เพื่อแจ้งเตือนปัญหาได้ทันท่วงที ระบบการแจ้งเตือนสามารถเป็นได้ทั้งในรูปแบบ dashboard หรือ third party channel อย่างเช่น e-mail ได้

- Dashboard

  เครื่องมือที่สามารถนำเอา event data มาทำเป็น chart เพื่อช่วยในการมอง pattern และชี้ให้เห็น activity ที่ไม่ได้อยู่ในรูปแบบปกติ

- Compliance

Application สามารถรวบรวมข้อมูลได้โดยอัตโนมัติ สร้างรายงาย (Report) ที่ปรับเข้ากับระบบความปลอดภัย (Security) , ระบบการจัดการ (governance) และการตรวจสอบที่มีอยู่

- Retention

เป็นแหล่งเก็บข้อมูลในระยะยาวได้เพื่ออำนวยความสะดวกให้กับ correlation ที่อยู่ในช่วงเวลานั้นๆ และยังมีการเก็บข้อกำหนดที่จำเป็นด้วย

- Forensic analysis

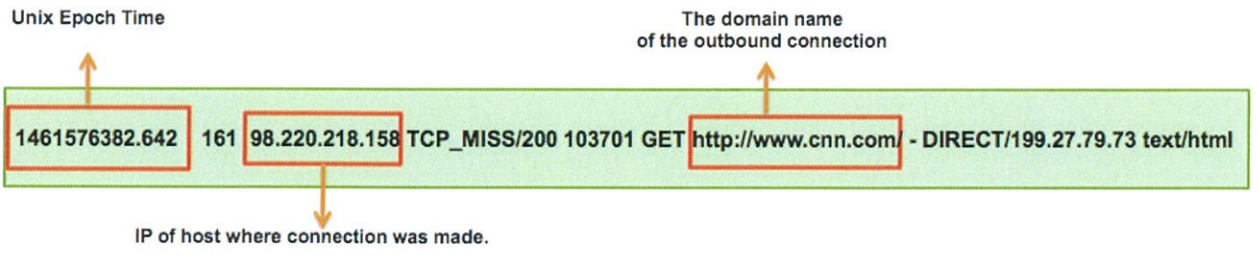
มีความสามารถในการค้นหาข้อมูลที่ต่างที่ (node) ต่างเวลา โดยมีเกณฑ์ที่เฉพาะเจาะจง ช่วยลดความยุ่งยากในการรวบรวมข้อมุลที่จะค้นหานับข้อมูลนับพันๆ ครั้ง

2.2 Kibana

Kibana คือ เครื่องมือ Visualize สำหรับแสดงผลข้อมูลจาก Elasticsearch ในรูปแบบต่างๆ เช่น กราฟแบบต่างๆ ตาราง แผนที่ และสามารถสร้างการแสดงผลข้อมูล หรือ Dashboard ได้ตามความต้องการ

2.3 Squid Proxy

Squid เป็น proxy สำหรับเว็บที่สนับสนุน HTTP, HTTPS, FTP และอื่น ๆ ลด Bandwidth และปรับปรุงเวลาในการตอบสนอง สมมติเมื่อใช้ squid ทำการเชื่อมต่อ http ขาออกไปยัง https://www.cnn.com จากโฮสต์ที่ระบุรายการต่อไปนี้จะถูกเพิ่มลงในไฟล์ Squid ที่เรียกว่า access.log



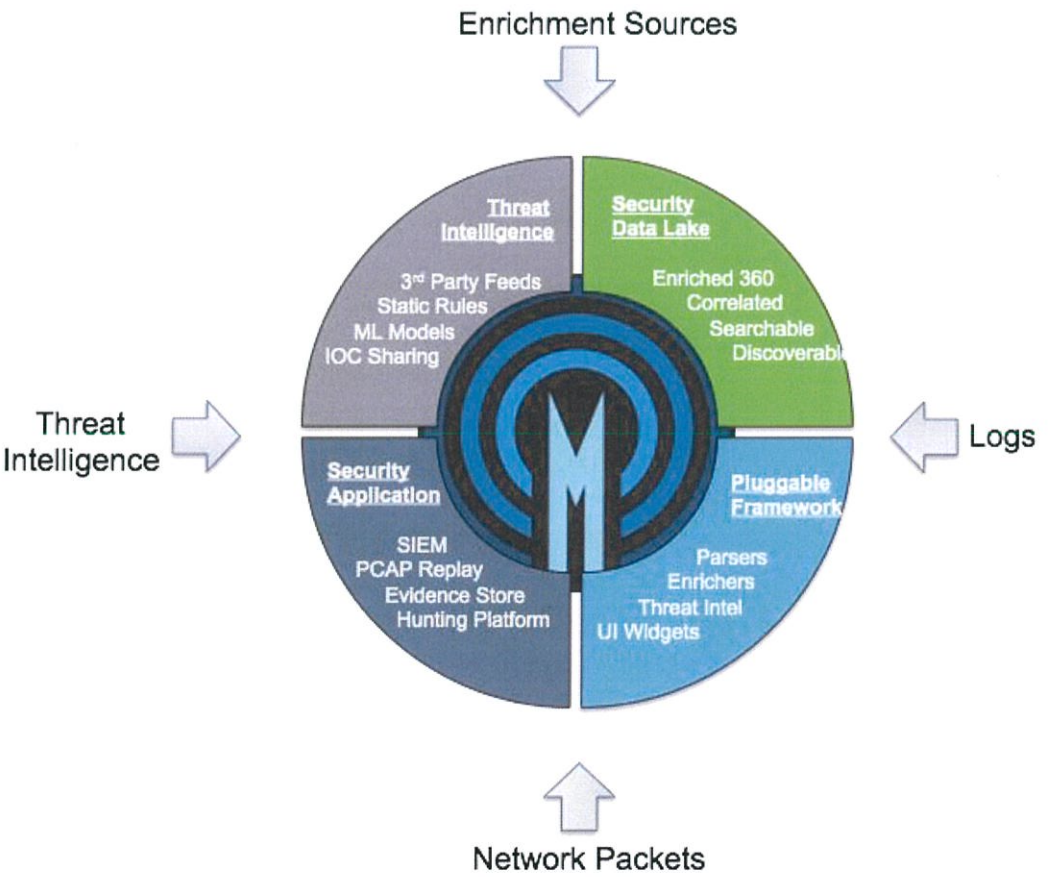
รูปที่ 2.1 รูปแบบตัวอย่าง Log ที่ได้จาก Squid

2.4 Apache Metron

Apache Metron นี้เป็น Framework สำหรับระบบ Big Data Cybersecurity โดยเฉพาะ ด้วยการนำเทคโนโลยี Open Source ในส่วนของ Big Data หลากหลายมารวมกันกลายเป็นเครื่องมือสำหรับการทำ Security Monitoring และ Security Analysis ที่รวมถึงความสามารถทั้งการรับข้อมูล, การวิเคราะห์ข้อมูล และการบันทึกข้อมูลขนาดใหญ่ที่เกี่ยวข้องกับ Security เช่น Security Data Feed, Log และ Network Metadata โดยสามารถทำ Log Aggregation, Full Packet Capture Indexing, Storage, Advanced Behavioral Analytics และ Data Enrichment ได้ในตัว ทำให้รองรับได้ทั้งข้อมูล Threat Intelligence (ข้อมูลภัยคุกคามอัจฉริยะ) ในปัจจุบัน ไปจนถึง Security Telemetry ได้อย่างครบถ้วนภายในระบบเดียว

2.4.1 ความสามารถหลักและลักษณะการทำงาน

Apache Metron นั้นมี 4 ความสามารถหลักดังนี้



รูปที่ 2.2 ไดอะแกรมเกี่ยวกับความสามารถของ Metron

1) Security Data Lake / Vault

เป็น Platform ที่เก็บ enriched telemetry data สำหรับช่วงเวลานาน Data Lake มีคลังข้อมูลสำหรับ feature engineering ที่ใช้ในการทำวิเคราห์ หลักการค้นหาและการดึงข้อมูลจาก Operational Analytics

2) Pluggable Framework

เป็น Platform ที่มี parser ที่ราคาไม่แพงสำหรับ Security Data Source (เช่น pcap, netflow, bro, snort, fireye, sourcefire) แค่นั้นยังไม่พอยังสามารถเพิ่ม parser สำหรับ data source ใหม่ได้ด้วย และเพิ่ม enrichment services ที่สามารถเพิ่ม contextual info ใน raw streaming data, extensions สำหรับ threat intel (ภัยคุกคาม) feeds และสามารถปรับปรุง dashboards ได้

3) Security Application

Metron นั้นมีความสามารถพื้นฐานของ SIEM (เช่น การแจ้งเตือน, threat intel framework, ตัวช่วยในการสืบค้นข้อมูล) และยังมี packet replay utilities, evidence store และ hunting service ที่ใช้โดย SOC analysts

4) Threat Intelligence Platform

Metron นั้นเป็น next generation defense techniques ที่ประกอบด้วยการใช้ anomaly detection และ machine learning algorithms ที่สามารถปรับใช้ได้แบบ real-time ในการ streaming ข้อมูล

2.4.2 ส่วนประกอบสำคัญที่เกี่ยวข้องกับ Apache Metron

Parsers (การแยกวิเคราะห์) : Parsers เป็นส่วนประกอบที่ใช้ในการแปลงข้อมูลดิบไปเป็นในรูปแบบ JSON และส่งต่อไปยังส่วน Enrichment โดยในที่นี้จะใช้ Grok Parsers ในการเปลลงข้อมูลเพื่อที่จะสามารถนำไปใช้ได้ในการenrichment

Enrichment (การเสริมสมรรถนะ) : เป็นส่วนที่นำข้อมูลที่ผ่านการ parsing data นำมาจัดทำให้อยู่ในรูปแบบที่สามารถบ่งบอกคุณลักษณะเพิ่มเติมอย่างเช่น tag message ลงไปว่าส่วนนี้เป็น alert หรือ สร้างระดับความเสี่ยงขึ้นมากำกับ เป็นต้น

Indexing (การสร้างดัชนี) : เป็นส่วนที่นำข้อมูลที่ผ่านการ enrich เพื่อนำไปเก็บใน elastic search และ HDFS โดยไฟล์ตั้งค่านั้นอยู่ในรูปแบบ JSON

2.4.3 โครงสร้างของ Apache Metron

ในส่วนของโครงสร้างของ Apache Metron สามารถถูกจำแนกได้ 2 แบบดังนี้

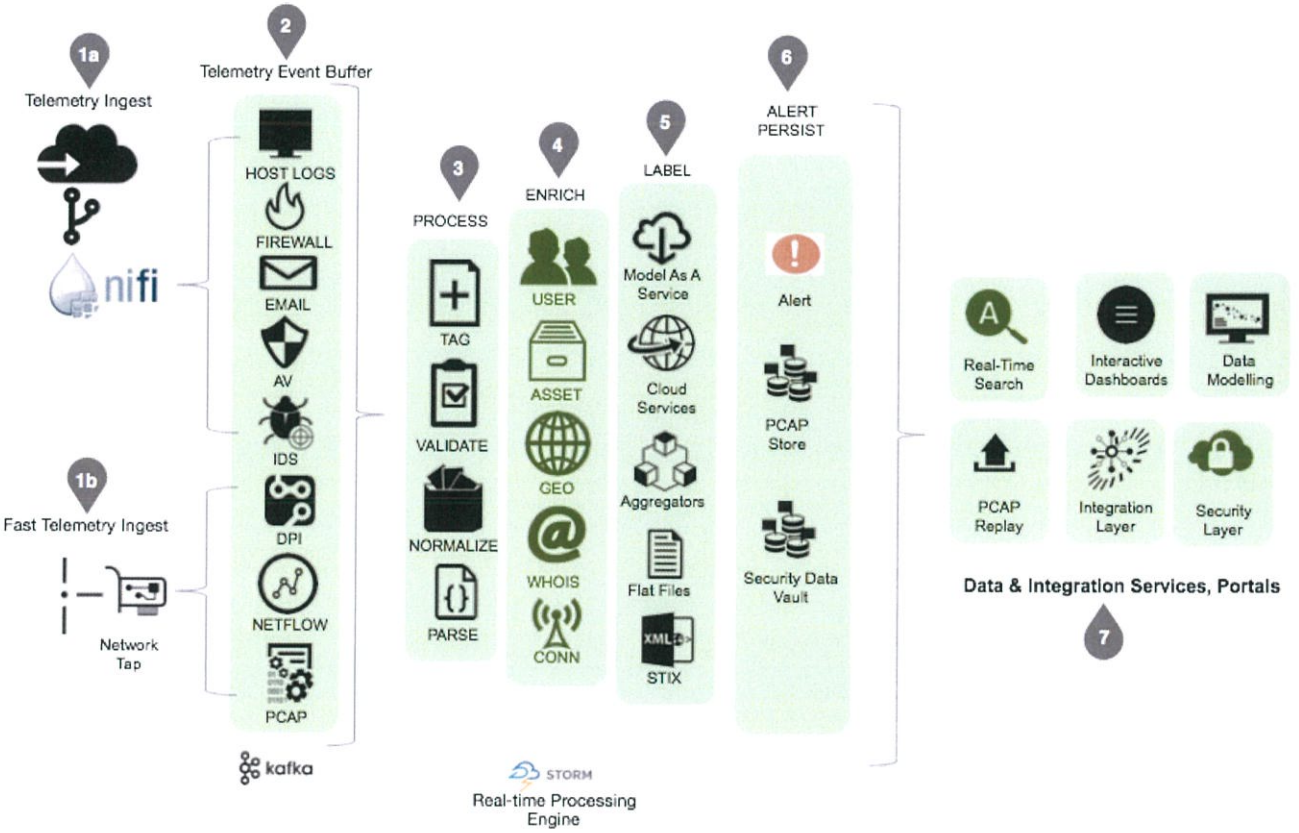
1) Metron Modules

Metron มีส่วนประกอบหลักๆ ดังต่อไปนี้

ตารางที่ 2.1 ส่วนประกอบของ Metron

ชื่อ Module	คำอธิบาย
metron-platform-metron-parsers	Topology สำหรับ normalizing telemetry จากในรูปแบบ native sensor ไปยัง Metron JSON
metron-platform-metron-enrichment	Topology สำหรับ enrichment ของ Metron JSON messages, รวมไปถึงการอ้างอิงสำหรับ threat intel stores และการส่งการแจ้งเตือน
metron-platform -metron-pcap	Topology สำหรับ streaming network packets ไปยัง HDFS สำหรับใช้ใน PCAP service
metron-platform-metron-api	Service สำหรับใช้ run analytics/filtering ใน PCAP files ใน HDFS ส่งมาโดย PCAP Topology
metron-sensors	Sensors ที่ส่งข้อมูลไปยัง Metron dashboards และ analytics
metron-platform-metron-data-management	Loaders สำหรับ bulk loading enrichment และ threat intelligence stores
metron-ui	Metron SOC Analyst UI
metron-deployment	Scripts สำหรับ automating Metron deployments

2) Logical Architecture



รูปที่ 2.3 Logic Architecture Diagram

ในส่วนย่อยของ event จะถูกส่งไปยัง logical components ที่ต่างกันออกไป

1) Telemetry Event Buffer

ทุกข้อมูลดิบ (raw events)จากแต่ละแหล่งข้อมูลจะถูกตรวจจับโดยโปรแกรมตรวจจับต่อจากนั้นจะถูกส่งไปยัง Kafka topic ของตัวเอง telemetry event ที่ถูกส่งมาถึงจะถูกส่งไปยัง ingest buffer เพื่อทำเครื่องหมายว่าส่วนนี้คือจุดเริ่มของการประมวลผลใน Metron

2) Process (Parse, Normalize, Validate and Tags)

แต่ละ raw events จะถูกแยกวิเคราะห์ (Parsing) และทำให้เป็นมาตรฐาน (Normalize) แล้วให้เป็นในรูปแบบ JSON ซึ่งจะทำให้ Topology correlation engine สามารถเชื่อมโยงข้อความที่แตกต่างกันไปข้องแต่ละข้อมูลได้ โดยที่ standard field names ใน JSON จะมี tuple อย่างน้อย 7 tuple เป็นอย่างต่ำเป็นดังนี้

- ip_src_addr: layer 3 source IP
- ip_dst_addr: layer 3 dest IP
- ip_src_port: layer 4 source port
- ip_dst_port: layer 4 dest port
- protocol: layer 4 protocol
- timestamp (epoch)
- original_string: A human friendly string representation of the message

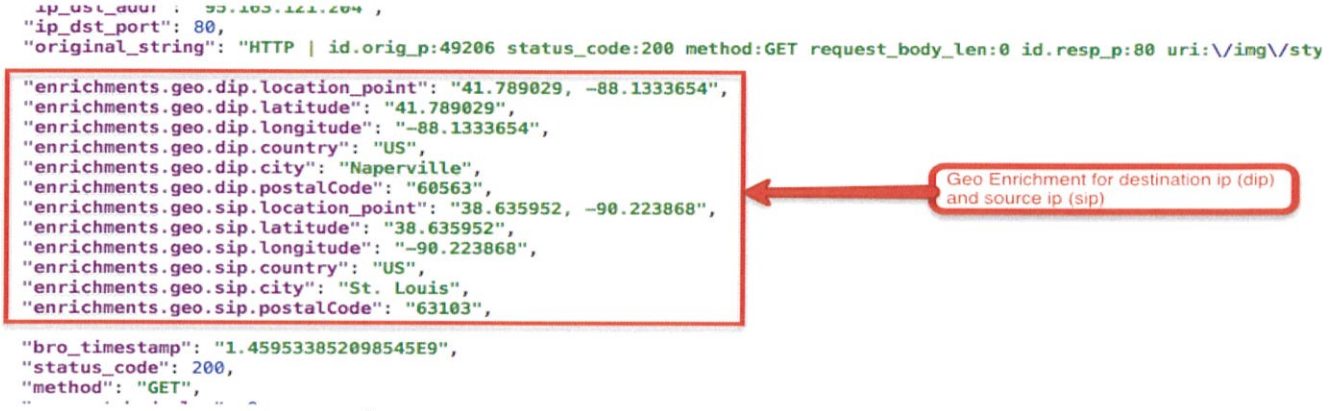
ที่ขั้นนี้ จะสามารถตรวจสอบข้อมูลดิบและ tag มันด้วยข้อมูลที่เป็น metadata ซึ่งจะถูกใช้ใน downstream processing

หลังจากข้อที่ 2 raw event จะเป็นรูปแบบดังต่อไปนี้

รูปที่ 2.4 ตัวอย่าง raw event หลังผ่าน parsing

3) Enrich

เมื่อ raw events ที่ได้รับการ parsing และทำการ normalize แล้วขั้นตอนต่อไปคือเพิ่มองค์ประกอบของข้อมูลที่ถูก normalize ให้ดูมีประโยชน์มากขึ้น ตัวอย่างการ enrich เช่น ข้อมูลพิกัดที่เป็นของ IP ภายนอกที่เป็นรูปแบบละติจูดและลองติจูด หรือข้อมูลที่จะมข้อมูลเกี่ยวกับ Host ของ IP นั้นๆ เป็นต้น



รูปที่ 2.5 ตัวอย่าง raw event หลังผ่าน enrichment

4) Label

หลังจากขั้น enrichment ส่วนนี้จะเป็นเหมือนส่วนเสริม โดยจะตรวจสอบว่าข้อมูล telemetry event นั้นมีส่วนในที่สามารถนำไปทำ threat intel ได้ โดยส่วนนี้จะถูกนำไป label ว่าตรงกับเงื่อนไขใน threat intel หรือไม่

รูปที่ 2.6 ตัวอย่าง raw event หลังผ่าน labeling

5) Persist

ในขั้นตอนนี้ telemetry event ที่ได้ผ่านกระบวนการข้างต้นจะถูกนำไปเก็บใน HDFS และส่วนที่ถูก Index จะถูกเก็บใน Elasticsearch

6) UI

เป็นการนำข้อมูลที่ผ่านกระบวนการข้างต้นทั้งหมดนำมาแสดงผลใน UI อย่างเช่น Kibana

2.5 Apache Storm

2.5.1 Core Concept ของ Apache Storm

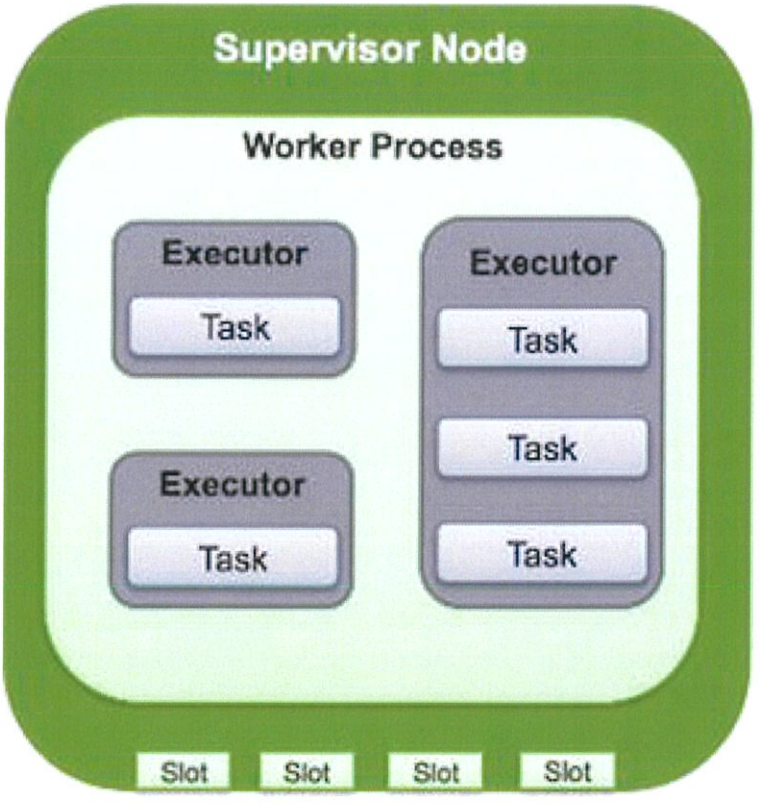
ตารางที่ 2.2 ส่วนประกอบของ Storm

Storm Concept	คำอธิบาย
Tuple	รายการที่ระบุชื่อค่าของชนิดข้อมูลใด ๆ tuple คือโครงสร้างข้อมูลพื้นเมืองที่ Storm ใช้
Stream	ลำดับที่ไม่มีที่สิ้นสุดของ tuples
Spout	สร้าง stream จาก data source แบบ real-time
Bolt	ประกอบด้วยการประมวลผลข้อมูล, persistence และการแจ้งเตือน alert logic นอกจากนี้ยังสามารถปล่อย tuples สำหรับ downstream bolts
Stream Grouping	ควบคุมการหาเส้นทางของ tuples ไปยัง bolts เพื่อทำงาน
Topology	กลุ่มของ spouts และ bolts ที่เชื่อมโยงไปยัง workflow (Storm Application)
Processing Reliability	Storm รับรองเกี่ยวกับการส่ง tuples ใน topology
Workers	Storm Process, Worker อาจจะรันมากกว่า 1 executor
Executors	Storm thread ที่สั่งเริ่มโดย Storm worker, Executor อาจจะรันมากกว่า 1 task
Tasks	Storm job จาก spout หรือ bolt
Parallelism	Attribute ของ distributed data processing ที่ถูกกำหนดว่าจะมีกี่ job กี่จำนวนที่ทำงานพร้อมกันใน topology
Process Controller	Monitor และ restart Storm process ที่ผิดผลาด
Master / Nimbus Node	Host ใน multi-node ของ Storm cluster ที่รัน process controller และ Storm nimbus, UI, และอื่นๆ ที่เกี่ยวกับ daemons
Slave Node	Host ใน multi-node ของ Storm cluster ที่รัน process controller daemon

Apache Storm มีตัวช่วยในการการกำหนดค่า cluster โดยทั่วไประบบยืนยันตัวตน(authentication) และระบบให้สิทธิ์(authorization) โดยทั่วไปจะถูกปิดอยู่ แต่สามารถเปิดได้ถ้าต้องการ ส่วนมาก features จะมีให้ใน Storm-0.10

2.5.2 Worker, Executors and Tasks

Apache Storm Process นั้นเรียกว่า workers ที่ทำการรัน port ที่กำหนดไว้บนเครื่อง host ของ Storm แต่ละ worker process นั้นสามารถมีมากกว่า 1 executors หรือ thread ซึ่งแต่ละ executor เป็น thread ที่ spawn จาก worker process แต่ละ executor นั้นสามารถรันมากกว่า 1 task ใน component เดียวกันได้ โดยที่ component นั้นเป็น spout หรือ bolt จาก topology



รูปที่ 2. 7 โครงสร้าง Thread ของ Storm

2.6 JSON

JSON (JavaScript Object Notation) เป็น lightweight data-interchange format (รูปแบบการแบบการแลกเปลี่ยนข้อมูลที่มีน้ำหนักเบา) ช่วยให้ง่านต่อการอ่านและเขียนของมนุษย์ มันง่ายที่ระบบ (machine) จะทำการ parse และ generate ซึ่งมีพื้นฐานมาจาก JavaScript Programming , Standard ECMA-262

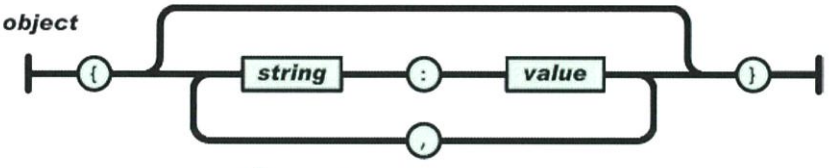
JSON เป็น text format ที่ภาษาที่อิสระสมบูรณ์ (completely language independent) แต่ใช้ข้อกำหนดคล้ายกับภาษา C รวมไปถึง C, C++, C#, Java, JavaScript, Perl, Python และอื่นๆ

JSON นั้นถูกสร้างมาด้วย 2 โครงสร้าง

1) การจัดเก็บในชุดข้อมูลที่มีชื่อข้อมูลและข้อมูลคู่กัน ในภาษาต่างๆ ข้อมูลจะจัดอยู่ในรูปแบบของ Object, record, struct, dictionary, hash table, keyed list หรือ associative array

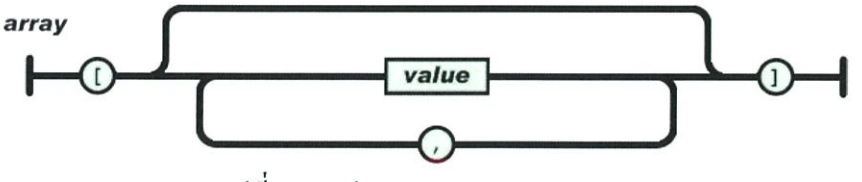
2) ลำดับของค่าข้อมูล ในภาษาโปรแกรมส่วนใหญ่ จะจัดอยู่ในรูปแบบของ array, vector, list หรือ sequence

ใน JSON จะมีรูปแบบดังนี้



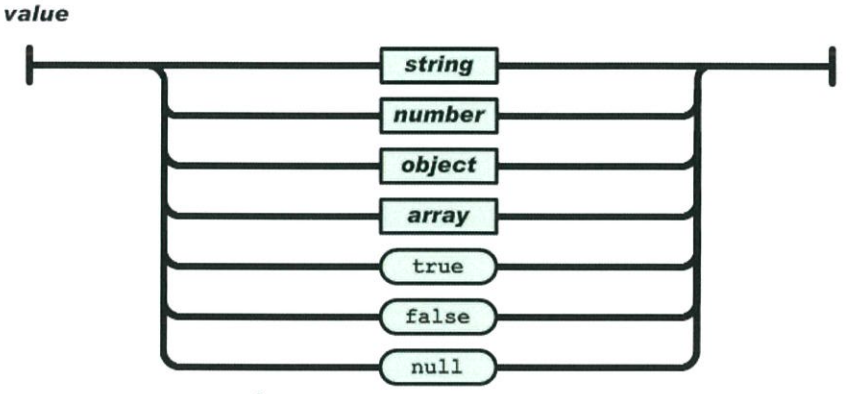
รูปที่ 2.8 รูปแบบ object ของ JSON

Object นั้นเป็นชุดของข้อมูลที่มีชื่อข้อมูลและค่าของข้อมูลนั้นคู่กัน ซึ่งจะถูกเริ่มต้นด้วยเครื่องหมาย { และจะปิดท้ายข้อมูลด้วยเครื่องหมาย } ข้อมูลแต่ละค่าจะมีเครื่องหมาย : กำกับระหว่างชื่อข้อมูลกับค่าของข้อมูล และแต่ละข้อมูลจะมีเครื่องหมาย , คั่น



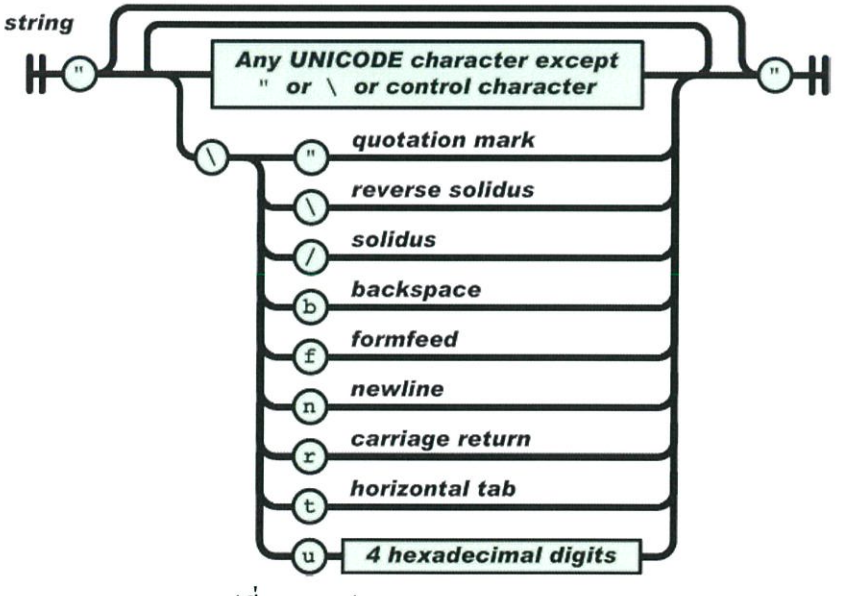
รูปที่ 2.9 รูปแบบ array ของ JSON

Array เป็นลำดับของข้อมูล ซึ่งจะถูกเริ่มต้นด้วยเครื่องหมาย [และจะจบด้วยเครื่องหมาย] แต่ละค่าของข้อมูลจะถูกคั่นด้วยเครื่องหมาย ,



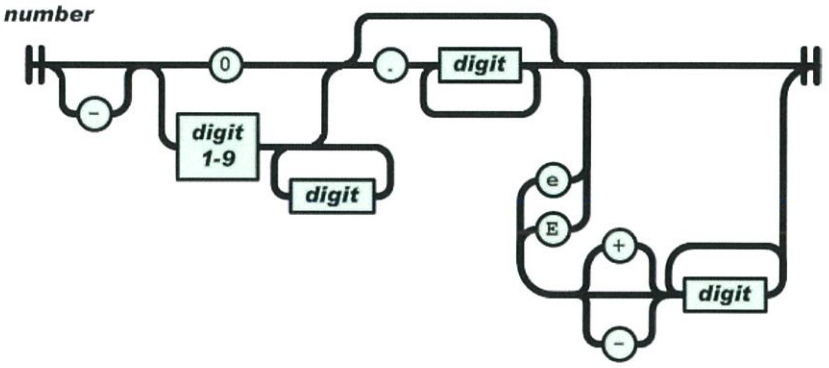
รูปที่ 2.10 รูปแบบ value ของ JSON

Value เป็น String ที่อยู่ในเครื่องหมาย “” หรือตัวเลข หรือค่าทางตรรกศาสตร์ true, false หรือค่า null หรือ object หรือ array ซึ่งโครงสร้างสามารถวางซ้อนกันได้



รูปที่ 2.11 รูปแบบ string ของ JSON

String เป็นลำดับของตัวอักษรตั้งแต่ 0 ตัวอักษรหรือมากกว่า ซึ่งอยู่ภายใต้เครื่องหมาย “” และจะใช้เครื่องหมาย ในการใส่เครื่องหมายกำกับต่างๆ ซึ่งจะมีลักษณะคล้ายกับ String ในภาษา C หรือภาษา Java



รูปที่ 2.12 รูปแบบ number ของ JSON

Number นั้นมีความคล้ายคลึงกับ Number ในภาษา C หรือภาษา Java อย่างมาก แต่จะไม่สามารถใช้เลขฐาน 8 กับเลขฐาน 16 ได้

ช่องว่าง(Whitespace) สามารถที่จะใส่ไว้ระหว่างสัญลักษณ์ต่างๆได้ ยกเว้นรายละเอียดซึ่งเข้ารหัสที่สมบูรณ์ในการบรรยายภาษาต่างๆ

2.7 Virtual Machine

Virtual machine (เครื่องเสมือน) เป็นไฟล์คอมพิวเตอร์ที่เรียกว่า image ทำหน้าที่คล้ายกับคอมพิวเตอร์จริงๆ ในอีกแง่หนึ่ง เป็นเหมือนคอมพิวเตอร์ที่สร้างคอมพิวเตอร์ได้ virtual machine เป็นสภาพแวดล้อมที่เหมาะสำหรับการทดสอบระบบปฏิบัติการอื่น ๆ รวมถึงรุ่นเบต้า การเข้าถึงข้อมูลที่ติดเชื้อไวรัส การสร้างการสำรองข้อมูล ระบบปฏิบัติการและการเรียกใช้ซอฟต์แวร์หรือแอพพลิเคชันบนระบบปฏิบัติการที่พวกเขาไม่ได้มีไว้สำหรับเดิม

Virtual machine สามารถทำงานพร้อมกันบนคอมพิวเตอร์ที่มีอยู่จริงเครื่องเดียวได้ สำหรับ Server ระบบปฏิบัติการหลายระบบ จำทำงานกับซอฟแวร์ที่เรียกว่า Hypervisor เพื่อจัดการกับซอฟแวร์เหล่านี้ในขณะที่คอมพิวเตอร์ที่เป็นแบบ Desktop ทั่วไปใช้ระบบปฏิบัติการหนึ่งเครื่องเพื่อเรียกใช้ระบบปฏิบัติการอื่นๆ ภายใน virtual machine แต่ละเครื่องจะมี virtual hardware(ฮาร์ดแวร์เสมือน)ของตัวเอง รวมไปถึง CPU, หน่วยความจำ(memory), แหล่งเก็บข้อมูล (hard drive), อุปกรณ์เชื่อมต่อต่างๆ (interface)

Virtual hardware นั้นจะถูกจับคู่เข้ากับฮาร์แวร์จริง(physical hardware) บนเครื่องกายภาพซึ่งช่วยประหยัดค่าใช้จ่ายโดยการลดความจำเป็นในระบบฮาร์แวร์ทางกายภาพพร้อมกับลดค่าใช้จ่ายในการบำรุงรักษาไปพร้อมกัน

Virtual machine ประกอบด้วยไฟล์หลายประเภทที่คุณเก็บไว้ในอุปกรณ์เก็บข้อมูลที่สนับสนุน (supported storage file) ไฟล์สำคัญที่ประกอบขึ้นเป็น virtual machine คือไฟล์การตั้งค่า (configuration file), ไฟล์ไดร์เสมือน (virtual disk file), ไฟล์การตั้งค่า NVRAM (NVRAM setting file)

ใน Apache Metron มีการใช้งาน Service ของ virtual machine จากชุด virtual machine ดังนี้

2.7.1 VirtualBox

VirtualBox เป็นผลิตภัณฑ์ virtualization x86 และ AMD64 / Intel64 ที่มีประสิทธิภาพสำหรับองค์กรรวมทั้งใช้ในบ้าน ไม่เพียงแต่ VirtualBox เป็นผลิตภัณฑ์ที่มีประสิทธิภาพและมีประสิทธิภาพสูงสำหรับองค์กรเท่านั้น แต่ยังเป็นวิธีแก้ปัญหาระดับมืออาชีพเพียงอย่างเดียวที่สามารถใช้งานได้ฟรีในรูปแบบ Open Source Software ภายใต้เงื่อนไขของ General Public License (GPL) version 2

2.7.2 Vagrant

Vagrant เป็นเครื่องมือสำหรับการสร้างและจัดการสภาพแวดล้อมของเครื่องเสมือนใน workflow เดียว ด้วยworkflow ที่ง่ายต่อการใช้งานและมุ่งเน้นไปที่ระบบอัตโนมัติ Vagrant ช่วยลดเวลาในการติดตั้งระบบการผลิตเพิ่มความเท่าเทียมกันในการผลิต

Vagrant ช่วยให้ง่ายในการกำหนดค่าทำซ้ำและสภาพแวดล้อมการทำงานแบบ portable ที่สร้างขึ้นบนเทคโนโลยีมาตรฐานอุตสาหกรรมและควบคุมด้วยเวิร์กโฟลว์ที่สอดคล้องกันเพื่อช่วยเพิ่มประสิทธิภาพ

2.8 Django Framework

เป็น framework ที่ใช้ในการสร้าง Web Application ในฝั่งของ Back End ที่พัฒนาด้วยภาษา Python โดยในตัว framework จะมีส่วนประกอบทุกอย่างที่จำเป็นตั้งแต่การเชื่อมต่อฐานข้อมูล ไปจนถึงการ render ข้อมูลออกมาให้ฝั่ง Front End แสดงผลข้อมูลเหล่านั้นได้

Django ช่วยให้ developer สร้าง website ด้วยความรวดเร็วและยังมีการช่วยป้องกันปัญหาด้าน Security ที่ developer อาจจะทำให้เกิดมากมาย ตัวอย่างเช่น SQL injection, cross-site scripting, cross-site request forgery และ clickjacking เป็นต้น

Django เป็น framework ที่มีการใช้งานแบบ stand alone application มีการเก็บสร้าง database ในตัวเองเหมาะแก่การ scale ของ application มีการช่วยจัดการ task ของ Web application ที่เกิดขึ้นได้ดี การที่ Django ใช้ Python ในการพัฒนา ทำให้ สามารถนำ Python มา migrate เข้ากับการสั่งงานตัว Server ของ Apache Metron ได้ ในการเขียน Script เพื่อเชื่อมโยงจัดการ Component ต่างๆ ของ Apache Metron

2.9 Apache Nifi

Apache Nifi เป็นซอฟต์แวร์โอเพนซอร์สที่ไว้ใช้จัดการเรื่องการไหลของข้อมูล เนื่องจากเซนเซอร์ข้อมูลที่มีข้อมูลไหลออกมาตลอดเวลาทำให้จำเป็นต้องมีซอฟต์แวร์มาคอยตรวจจับข้อมูลเหล่านี้

บทที่ 3

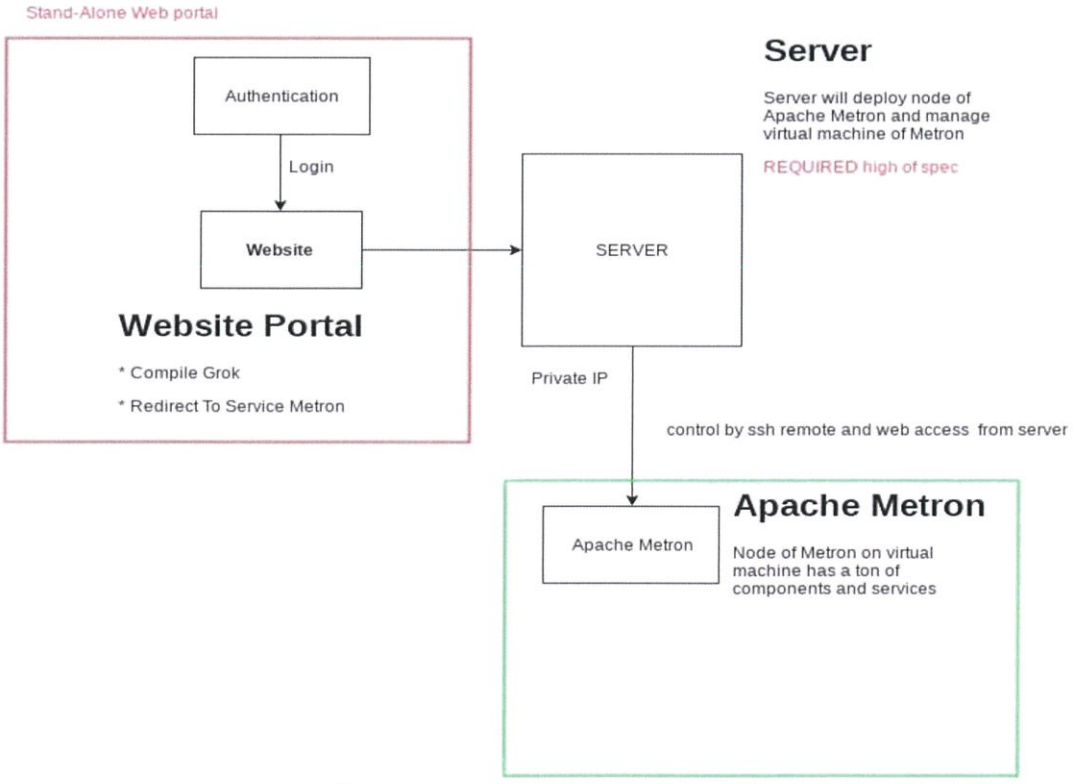
การออกแบบพัตนา

3.1 ความต้องการพื้นฐานของโปรแกรม

เนื่องจาก Log แต่ละประเภท จะมีข้อมูลใน Log ในแต่ละประเภทที่ไม่เหมือนกัน โดยในการออกแบบพัฒนาในส่วนนี้นั้นจะใช้ Squid Proxy เนื่องจากมีความเรียบง่ายในการตรวจจับ Log โดยจะนำส่วนพื้นฐานในการตรวจสอบ Log อย่างเช่น Source IP Address , Destination IP Address และ URL มา และมีหน้าเว็บไซต์ที่สามารถจัดการในส่วนของการ Parse Log ได้เลยโดยที่ไม่จำเป็นต้องเข้าไปในเซิฟเวอร์ ซึ่งมีการออกแบบกรณีตัวอย่างตามเงื่อนไขดังต่อไปนี้

1. Log จำเป็นต้องเข้ามาตามเวลาความเป็นจริง (Real-Time)
2. Log จำเป็นต้องอยู่ในรูปแบบ JSON เพื่อให้ Apache Metron เข้าใจ
3. Log จำเป็นต้องมีการเพิ่มคุณค่าของข้อมูลเพื่อใช้ในการตรวจสอบข้อมูลอื่นๆ ต่อไป
4. มีการแจ้งเตือนมีการเข้าถึงเว็บไซต์อันตราย

3.2 หลักการออกแบบเว็บไซต์



รูปที่ 3.1 ไดอะแกรมการออกแบบเว็บไซต์

ใช้ Django ที่เป็น Python มาเป็น Stand-Alone Website ที่รวมเข้าเครื่อง Server หลักในการจัดการละควบคุมการใช้ Metron ที่อยู่ยน Virtual Server จากด้านในตัวเซิฟเวอร์ เนื่องจาก Python เป็นภาษาที่ใช้จัดการกับเครื่องเซิฟเวอร์ ให้ง่ายแก่การใช้ร่วมกับเครื่องเซิฟเวอร์ เพื่อเชื่อมโยงไปยัง Virtual Node ของ Metron

การออกแบบจะให้ระบบ login เพื่อจะเข้าใช้งานตัว SIEM ในหน้าควบคุมหลัก แล้วเชื่อมไปยังส่วนระบบการใช้งานระบบบริการที่ SIEM มีมาให้ผนวกกับส่วนหน้าเว็บไซต์ที่ได้สร้างขึ้นเองเข้าไปด้วย

ในส่วนของการออกแบบหน้าเว็บไซต์ที่มีการใช้งานง่ายนั้นก็คือการประมวณผลหรือเพิ่ม Telemetry ใหม่เข้าไปใน Topology ของ ระบบ SIEM ซึ่งจะออกแบบมีช่องสำหรับรับ Text บนหน้าเว็บไซต์ 3 ส่วน คือ

- Grok Statement สำหรับรับ Grok Statement แล้วนำไปใช้
- Path of Your Log File คือ Path ที่เก็บ Log ไว้
- Topic Names คือ ชื่อของ Topic ที่ต้องการ

นำไปรวมกับระบบการทำงานเบื้องหลังของการเข้าไปสั่งงาน Node ของ Metron ที่อยู่ใน Private IP ที่สามารถเข้าถึงได้จากแค่ localhost

3.3 การเพิ่มข้อมูลที่ต้องการสำหรับการตรวจจับ Log

ทดลองสร้าง Log พื้นฐานเพื่อทำการตรวจจับ Log ตามความต้องการพื้นฐานที่กำหนด โดยใช้ Squid Proxy ในการสร้าง Log ตัวอย่างมา

ตัวอย่างที่ 3.1 ตัวอย่างไฟล์ Log ที่ได้จาก Squid Proxy

```
1767102357.401 406 127.0.0.1 TCP_MISS/200 337891 GET
http://www.pravda.ru/science/? - DIRECT/207.109.73.154
text/html
```

จาก Log ที่สร้างมานั้น เห็นได้ว่าสามารถแบ่งรูปแบบของ Log เป็นส่วนๆได้ดังต่อไปนี้

ตารางที่ 3.1 คำอธิบายส่วนประกอบของ Log ใน Squid

1767102357.401	เวลา (Timestamp)
406	เวลาที่ใช้ในการติดต่อ (Time elaspsed)
127.0.0.1	Client Address
TCP_MISS/200	สถาะของ http

337891	Bytes
GET	http method
http://www.pravda.ru/science/?	URL
DIRECT/207.109.73.154	peerstatus/host
text/html	ประเภทของเว็บไซต์

ซึ่งรูปแบบของ Log นั้นจะแตกต่างไปตามอุปกรณ์ที่ไปเก็บ Log มา ต่อมาทำการสร้าง Grok Statement ที่ได้รับมาจากหน้าเว็บไซต์

3.4 วิเคราะห์หลักการออกแบบ

3.4.1 นำ Grok Statement ไปใช้งาน

นำ Grok Statement ที่ได้จากเว็บไซต์นำไปสร้างไฟล์เพื่อเตรียมในการใช้งานในระบบสำหรับการ Parse Log ต่อไป โดยนำ Log จากตัวอย่างที่ 3.1 มาใช้กับ Grok Statement

ต้องแยกประเภทว่าเราต้องการประเภทไหนใน Grok Statement เพื่อนำแยกแยะประเภท Log แต่ละรูปแบบซึ่งเราจะใช้ ประเภท Grok ดังต่อไปนี้ NUMBER, INT, IP, WORD โดยแต่ละประเภทต้องอยู่ใน Grammar ดังต่อไปนี้

- INT (?:[+-]?(?:[0-9]+))
- NUMBER (?:%{BASE10NUM})
- IP (?:%{IPV6}|%{IPV4})
- WORD \b\w+\b

จากตัวอย่างที่ 3.1 เราสามารถแบ่งประเภทของ Log ได้เป็นดังตารางที่ 3.2 โดยแต่ละประเภทจะใช้กับข้อมูลที่แตกต่างกันออกไป

- INT และ NUMBER รูปแบบข้อความที่เป็นตัวเลข อย่างเช่น เวลาหรือขนาด เพื่อที่จะได้สามารถนำไปคำนวณต่อไปได้
- WORD คือรูปแบบข้อความที่เป็นตัวอักษรใน Log อย่างเช่นรูปแบบ Method อย่าง GET หรือ POST
- IP คือรูปแบบข้อความที่เป็น IP
- NOTSPACE คือรูปแบบข้อความที่ต้องการไม่ให้มีช่องว่างเมื่อนำไปใช้งานซึ่งเหมาะแก่ URL

ตารางที่ 3.2 ตารางข้อมูลในการแบ่งประเภทของ Log เพื่อนำไปใช้กับ Grok Statement

1767102357.401	เวลา (Timestamp)	%{NUMBER:timestamp}
406	เวลาที่ใช้ในการติดต่อ (Time elaspsed)	%{INT:elapsed}
127.0.0.1	Client Address	%{IP:ip_src_address}
TCP_MISS/200	สถาะของ http	%{WORD:action}/%{NUMBER:code}
337891	Bytes	%{NUMBER:bytes}
GET	http method	%{WORD:method}
http://www.pravda.ru/science/?	URL	%{NOTSPACE:url}
DIRECT/207.109.73.154	peerstatus/host	.*%{IP:ip_dst_addr}
text/html	ประเภทของเว็บไซต์	ไม่ได้ใช้งาน

โดยสามารถเขียน Grok Statement ได้ตามโปรแกรมที่ 3.1 โดย SQUID_DELIMITED ที่ใส่ไปนั้นเป็น Tag ที่ไว้ตรวจสอบความถูกต้องไม่ได้มีผลกับ Statement ที่เหลือ

โปรแกรมที่ 3.1 Grok Statement สำหรับการ parsing Log

```
SQUID_DELIMITED %{NUMBER:timestamp}.*%{INT:elapsed}
%{IP:ip_src_address} %{WORD:action}/%{NUMBER:code}
%{NUMBER:bytes} %{WORD:method}
%{NOTSPACE:url}.*%{IP:ip_dst_addr}
```

3.4.2 ทำการสร้าง Topology ใหม่เพื่อเตรียมรับข้อมูลใหม่

ทำการสร้าง Topic ใน metron ตามชื่อที่ได้รับมาจากหัวข้อ โดยจำเป็นต้องหยุดการทำงานของ Topology นั้น ๆ ก่อน ซึ่ง Topology ที่ถูกสร้างนั้นจะมีการตั้งค่าในขึ้นตอนต่อๆไป เพื่อที่จะสามารถนำ Topology

3.4.3 ทำการตั้งค่า Parser

ตั้งค่า Parser ในรูปแบบ JSON ว่า Parser นั้นจะ Topic นั้นมีชื่อว่าอะไร รูปแบบ Pattern สำหรับการจัดการ Log นั้นอยู่ที่ไหนบ้างและต้องการให้ข้อมูลของ Log นั้นมีรูปแบบอย่างไรเมื่อนำไปใช้งาน ต่อมาจะแปลง Domain หรือ URL ที่ได้มาจาก Log ซึ่งจะตัด Subdomain ออกไปหรือที่เรียกว่า Domain without subdomain

```
http://www.reg.kmitl.ac.th/KMITL/index.php
```

จากข้อความข้างต้นคือ Domain ที่ยังมี Subdomain อยู่

```
reg.kmitl.ac.th
```

จากข้อความข้างต้นคือ Domain ที่ตัด Subdomain แล้ว

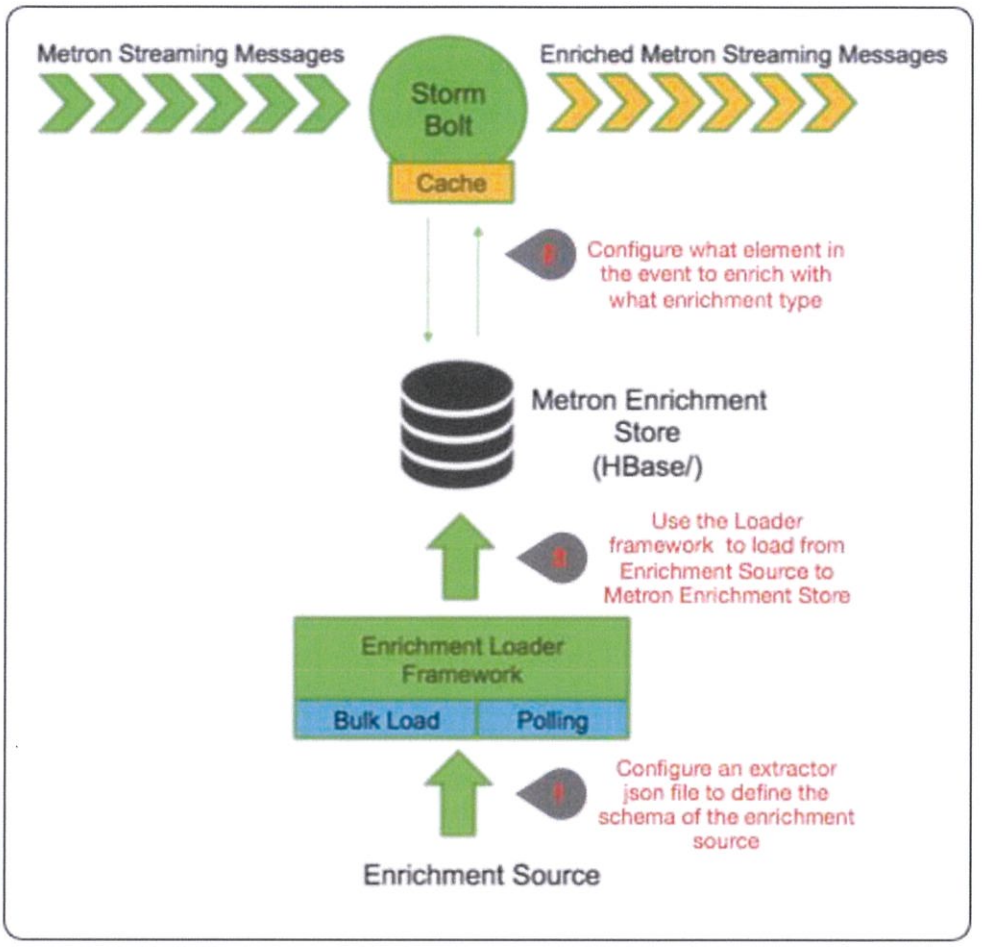
เหตุผลที่ตัด subdomain ออกเพราะจะนำ Domain without subdomain นั้นนำไปใช้ในส่วน Enrichment ในส่วนการทำข้อมูล Blacklist เว็บไซต์

3.4.4 ทำการตั้งค่า Index

ตั้งค่า Indexing ในรูปแบบ JSON ว่าต้องการให้ไฟล์ที่ทำการ Indexing นั้นไปเก็บไว้ที่ไหนซึ่งในที่นี้เราจำเป็นต้องให้เก็บไว้ใน HDFS เพื่อที่จะสามารถตรวจสอบไฟล์ที่ Index มาแล้ว

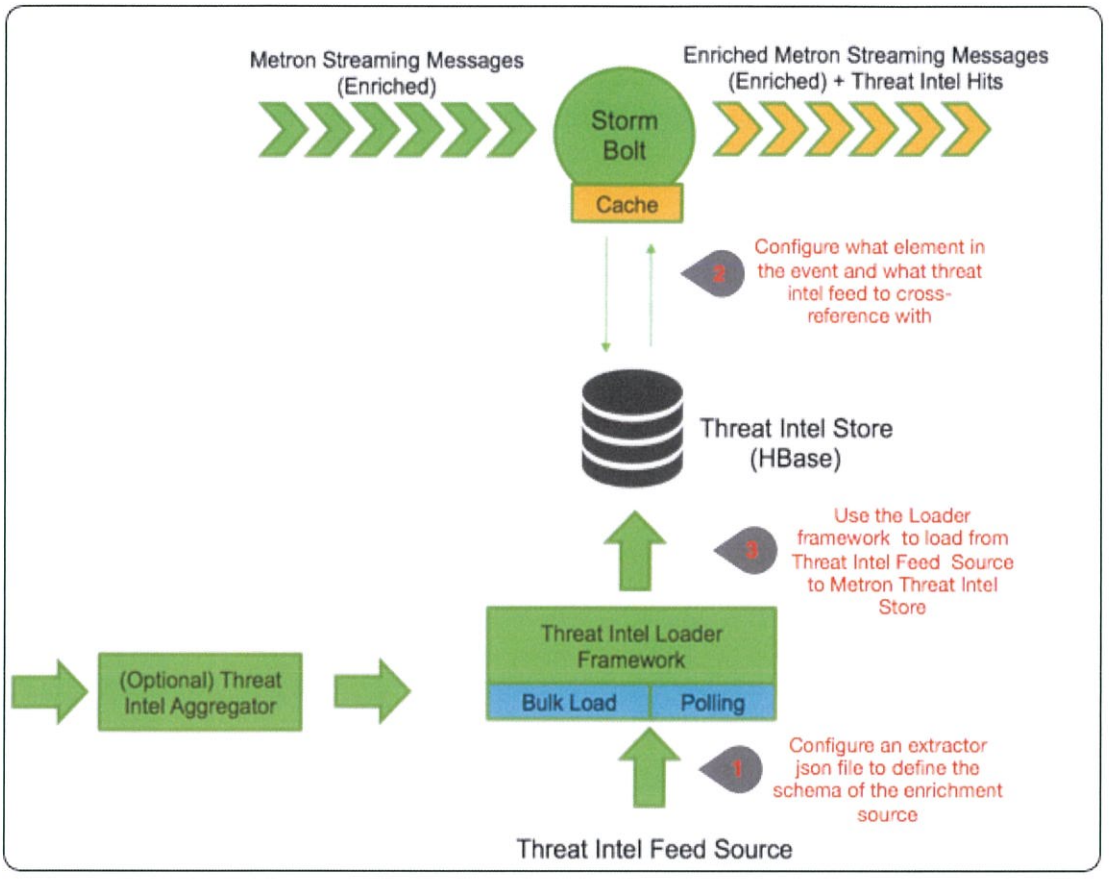
3.4.5 ทำการตั้งค่า Enrichment

ตั้งค่า Enrichment ว่าต้องการเพิ่มคุณประโยชน์ให้แก่ข้อมูลยังในรูปแบบไหนบ้าง สิ่งที่จะเพิ่มมาในส่วน Enrichment นั้นมี 2 ส่วนคือ Whois และ Black List โดย Whois คือข้อมูลที่เป็นการบ่งบอกว่าเว็บไซต์ที่เข้านั้น ใครเป็นเจ้าของ ใครจดทะเบียนเป็นต้น ส่วน Black List คือ ข้อมูลเว็บไซต์ ที่ไม่ปลอดภัย, เว็บไซต์หลอกลวง เป็นต้น หลักการทำงาน และจะมีการแจ้งเตือนแก่ผู้ใช้งาน (Threat intel) เมื่อเข้าเว็บไซต์ Blacklist ที่กล่าวไปข้างต้น หรือที่เรียกว่า Enrichment Topology นั้นจะเป็นตามรูปดังต่อไปนี้



รูปที่ 3.2 หลักการออกแบบ Enrichment

1) ตั้งค่า JSON ไฟล์ที่ต้องการดึงข้อมูลออกมาจากไฟล์ทั้ง Whois และ Blacklist
2) เลือกข้อมูลที่ต้องการนำมาใช้งานสำหรับการตรวจสอบในส่วนของ Whois นั้นเราจะดึงข้อมูลออกมาคือ domain , เจ้าของ, ประเทศที่จดทะเบียน, นายทะเบียน
3) สั่งนำไฟล์ตั้งค่าที่เพิ่งสร้างนั้นโหลดลงไปใน Apache Metron เพื่อให้สามารถเพิ่มข้อมูลต่างๆ ที่เพิ่มลงไปได้



รูปที่ 3.3 หลักการออกแบบ Enrichment และ Threat Intel ของ Blacklist

ในส่วนของ Black List ที่มีการแจ้งเตือนนั้น จะเพิ่มเติมเข้าไปในส่วนของการตั้งค่าก่อนที่จะทำการโหลดเข้าไปใน Metron (ขั้นตอนที่ 3) ดังรูปที่ 3.3

- Whois

สำหรับข้อมูล Whois นั้น จะถูกเก็บอยู่ในรูปแบบ CSV ดังข้อความข้างล่างต่อไปนี้

```
google.com, "Google Inc.", "US", "Dns Admin",874306800000

work.net, "", "US", "PERFECT PRIVACY, LLC",788706000000

capitalone.com, "Capital One Services, Inc.", "US", "Domain
Manager",795081600000

cisco.com, "Cisco Technology Inc.", "US", "Info Sec",547988400000

cnn.com, "Turner Broadcasting System, Inc.", "US", "Domain Name
Manager",748695600000
```

ตารางที่ 3.3 ตารางตัวอย่างข้อมูล Whois

google.com	domain
Google Inc.	เจ้าของ (owner)
US	ประเทศที่จดทะเบียน (Country)
Dns Admin	นายทะเบียน (Registrar)
8743068000	เวลาที่จดทะเบียน (Timestamp domain Create)

- Blacklist

เป็นข้อมูลเว็บไซต์ที่ถูก Blacklist โดยจะนำข้อมูลเว็บไซต์เหล่านั้นจากเว็บไซต์ zeus malware tracker รูปแบบข้อมูล Blaclist จะเป็นตามข้อความดังต่อไปนี้

```
bright.su
brothersmt2.tk
burgerspendingbusiness.kz
burn.settingsdata.store
bytes.darktech.org
canadianonlineagreementservices.kz
capacitacion.inami.gob.mx
casher777soft.pw
cd31411.tmweb.ru
chambercb.tk
```

ซึ่งเป็นข้อความที่มีแต่เว็บไซต์ที่ถูก Blacklist เท่านั้น ในส่วนของการทำ Black List นั้นเราต้องทำการสร้างไฟล์ตั้งค่าที่บ่งบอกว่า หากเว็บไซต์ที่เข้านั้นตรงกับ Blacklist จะให้มีการแจ้งเตือนให้แก่ผู้ใช้งาน

3.4.5 ตรวจสอบความถูกต้องของ Log

จำเป็นต้องมีการตรวจสอบว่า Source IP และ Destination IP นั้นมีความถูกต้องหรือไม่ โดยตั้งค่าไฟล์ในรูปแบบ JSON แล้วสั่งให้โหลดไปที่ zookeeper ซึ่งในจุดนี้เพื่อตรวจสอบ IP ว่า IP ที่เข้ามานั้นถูกต้องหรือไม่ ถ้าผิด IP เหล่านั้นจะไม่ถูกนำมาแสดงผล

3.4.6 สร้าง Topology ของ Parser ที่ได้รับจากเว็บไซต์

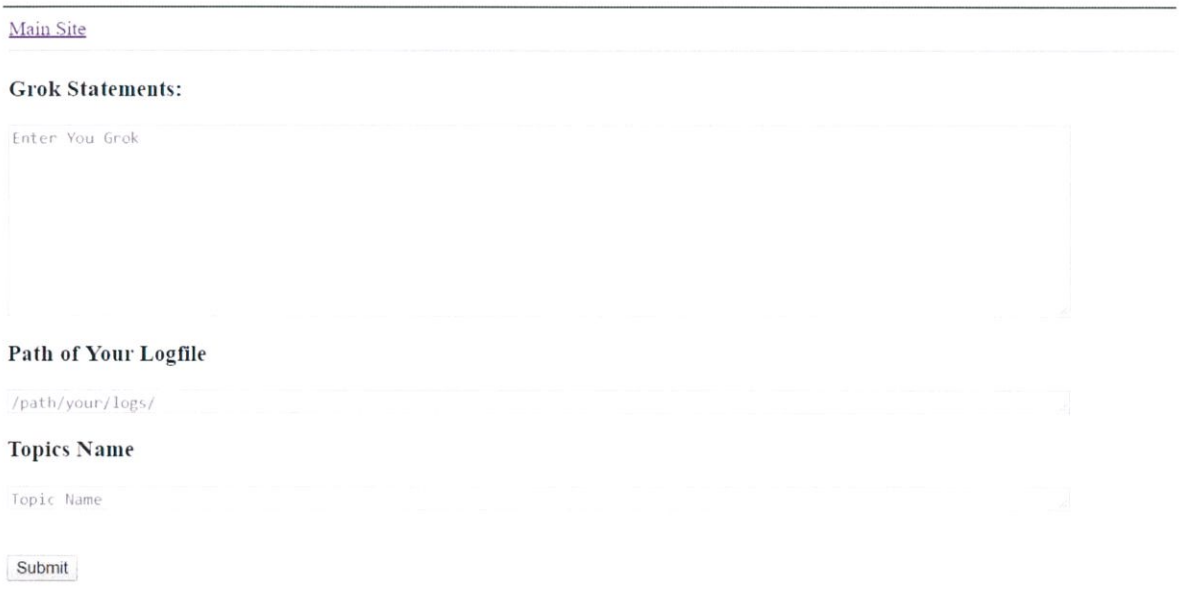
ทำการติดตั้ง Parser ที่สามารถจับรูปแบบ Log ได้ตรงตาม Grok Statement ที่ได้รับมาจากเว็บไซต์ โดย Topology นี้จะนำข้อมลจาก Topic ที่ถูกสร้างโดยชื่อเดียวกันแล้วจะนำข้อมูลมาประมวลผล ให้อยู่ในรูปแบบ JSON และส่งต่อไปยัง Topic enrichment ต่อไป

บทที่ 4

การทดลองและผลการทดลอง

4.1 หน้าแสดงผลของเว็บไซต์

ในส่วนของการทดลองและผลการทดลองนั้นเราได้ทำการทดลองหน้าเว็บไซต์ที่ไว้ใช้ในการสำหรับ Parse Log ให้แก่ผู้ใช้งาน



รูปที่ 4. 1 หน้าจอแสดงผลของเว็บไซต์

โดยในหน้าเว็บไซต์นั้นจะมีตัวรับ Input ทั้งหมด 3 อย่างด้วยกันคือ

- Grok Statements ที่มีไว้รับ Statements ของ Grok ตามรูปแบบ Log ที่ได้รับเข้ามา

รูปแบบ Log คือ

```
Nov 29 09:07:56 host_01456 named 5987: client 192.168.209.43#443
query:inkedin.com IN A +ED
```

ตัวอย่าง Grok Statement ที่สามารถเขียนใช้ในการตรวจสอบ Log ข้างต้นได้

```
Grok Statement : %{SYSLOGTIMESTAMP:timestamp} %{DATA:agent} %{DATA:protocol}
%{POSINT:rqstPort}: %{DATA:client} %{IP:srcIp}#%{POSINT:srcPort} %{DATA}:
%{DATA:domain} %{DATA:namespace} %{GREEDYDATA:typeName}
```

อีกหนึ่งตัวอย่างของรูปแบบ Log

```
Nov 29 09:07:56 host_01456 named 5987: client 192.168.209.43#443
query:inkedin.com IN A +ED
```

ตัวอย่าง Grok Statement ที่สามารถเขียนใช้ในการตรวจสอบ Log ข้างต้นได้

```
%{NUMBER:timestamp}.*%{INT:elapsed} %{IP:ip_src_address}
%{WORD:action}/%{INT:code} %{NUMBER:bytes} %{WORD:method}
%{NOTSPACE:url}.*%{IP:ip_dst_addr}
```

- Path of Your Log file คือที่อยู่ของ Log ที่ต้องการนำมาใช้กับ Grok Statement สมมติให้เก็บไว้ใน var/log/squid/access.log
- Topic Name คือชื่อที่ต้องการตั้ง เมื่อนำไปใช้งานในการแสดง Log ต่าง ๆ อย่างใน elasticsearch และ kibana

4.2 การตั้งค่าและการทำงานภายในเซิฟเวอร์

4.2.1 สร้าง Kafka Topic

ทำการสร้าง Topic ขึ้นมาตาม Topic Name ที่ได้รับมาจากเว็บไซต์โดยทำการรันโดยไฟล์ kafka-topics.sh นั้นเป็นไฟล์ใช้สำหรับการสร้าง Kafka Topic

โปรแกรมที่ 4.1 คำสั่งสร้าง kafka topic

```
kafka-topics.sh --zookeeper $ZOOKEEPER_HOST:2181 --create --topic squid
--partitions 1 --replication-factor 1
```

4.2.2 นำ Grok Statement ไปใส่ในที่จัดเก็บ Pattern ของ Metron

ใช้คำสั่งเพื่อย้ายไปเก็บไว้ใน patterns ของ metron

โปรแกรมที่ 4.2 คำสั่งย้าย Grok Statement ไปไว้ในที่จัดเก็บ

```
hdfs dfs -put /tmp/{Topic Name} /apps/metron/patterns
```

4.2.3 ตั้งค่าไฟล์ Config ของ Parser

ตั้งค่าไฟล์ Config ของ Parser

โปรแกรมที่ 4.3 ไฟล์ตั้งค่าของ Parser

```
{
  "parserClassName":
"org.apache.metron.parsers.GrokParser",
  "sensorTopic": "squid",
  "parserConfig": {
    "grokPath": "/apps/metron/patterns/squid",
    "patternLabel": "SQUID_DELIMITED ",
  },
"fieldTransformations" : [
     {
     "transformation" : "STELLAR"
    ,"output" : [ "full_hostname",
"domain_without_subdomains" ]
    ,"config" : {
                    "full_hostname" : "URL_TO_HOST(url)"
                   ,"domain_without_subdomains" :
"DOMAIN_REMOVE_SUBDOMAINS(full_hostname)"
                   }
     }
]
}
```

ตารางที่ 4. 1 คำอธิบายของไฟล์ JSON ของ parser

parserClassName	เป็นการบ่งบอกว่าจะใช้ Parser อะไร โดยในที่นี้จะใช้ Grokparser
sensorTopic	ชื่อของ Topic
grokPath	ตำแหน่งของ Grok Statement ที่เก็บไว้
output	คือการสร้าง filed ขึ้นมาใหม่ตามชื่อที่ตั้ง
config	คือการตั้งค่าเฉพาะที่ตรงตาม function ในส่วนของ transformation

เนื่องจาก Grok Statement ที่เราตั้งนั้นทำเพียงแค่นำ URL ทั้งหมดออกมาแต่ที่ต้องการจริง ๆ คือ domain หรือ domain ที่ไม่มี subdomain โดยเราจะใช้ Stellar Language ซึ่งเป็นภาษาเฉพาะที่สามารถใช้ผู้ใช้งานสามารถกำหนดการเปลี่ยนแปลงของข้อมูลได้

การตั้งค่าโดยจะมีการใช้ฟังก์ชัน URL_TO_HOST กับ URL เพื่อตรวจสอบ hostname จาก URL ดังกล่าวและใช้ DOMAIN_REMOVE_SUBDOMAINS เพื่อนำ subdomain ออก

ใช้คำสั่งต่อไปนี้เพื่อเก็บค่า Config ลงไปในระบบ

โปรแกรมที่ 4. 4 คำสั่งเพื่อเก็บค่า Config

```
zk_load_configs.sh --mode PUSH -i
/usr/metron/$METRON_VERSION/config/zookeeper -z node12181
```

4.2.4 การตั้งค่าไฟล์ Index

โปรแกรมที่ 4.5 ไฟล์ตั้งค่า Index

```
{
"hdfs" : {
      "index": "squid",
      "batchSize": 5,
      "enabled" : true
},
"elasticsearch" : {
      "index": "squid",
      "batchSize": 5,
      "enabled" : true
}
}
```

ไฟล์ตั้งค่า Index จะมีตัวบ่งบอกว่าจะใช้เขียนข้อมูลที่ได้ทำการ Index แล้วลงไปในที่ไหนบ้าง โดยจะมี hdfs และ elasticsearch ซึ่งจะมีการตั้งค่า ชื่อ, ขนาด Batch และฟังก์ชันที่สั่งให้ hdfs และ elasticsearch ทำงาน

4.2.5 การตั้งค่าเพื่อตรวจสอบความถูกต้องของข้อความ

อีกหนึ่งอย่างที่จำเป็นต้องทำคือการตรวจสอบความถูกต้องของข้อความ โดยเราจะตรวจสอบเพียงแค่ Source IP และ Destination IP นั้นว่าถูกต้องหรือไม่โดยตั้งค่าไฟล์ต่อไปนี้

โปรแกรมที่ 4.6 ไฟล์ตั้งค่าตรวจสอบความถูกต้อง

```
"fieldValidations" : [
   {
      "input" : [ "ip_src_addr", "ip_dst_addr" ],
      "validation" : "IP",
      "config" : {
      "type" : "IPV4"
       }
   }
  ]
```

4.2.6 การตั้งค่าไฟล์ Enrichment

เริ่มจากการตั้งค่าไฟล์ของ Whois ว่าต้องส่วนไหนบ้าง ในส่วนแรกคือตั้งค่าการดึงข้อมูลจากไฟล์ CSV และส่วนสองคือการนำข้อมูลโหลดเข้าไปใน Metron และจะใช้ข้อมูลในไหนในการตรวจสอบกับข้อมูล Whois

โปรแกรมที่ 4.7 ไฟล์ตั้งค่า Whois ส่วนแรก

```
{
"config" : {
   "columns" : {
      "domain" : 0
      ,"owner" : 1
      ,"home_country" : 2
      ,"registrar": 3
      ,"domain_created_timestamp": 4
   }
   ,"indicator_column" : "domain"
   ,"type" : "whois"
   ,"separator" : ","
  }
  ,"extractor" : "CSV"
}
```

จากบทที่ 3 ตารางที่ 3.3 นั้นจะบ่งบอกไว้แล้วว่าส่วนไหนของคอลลัมน์มีข้อมูลอะไรบ้างเราสามารถสั่งตั้งค่าได้เลยโดยเรียงในรูปแบบ domain, owner, home country, registrar, domain created timestamp ตามลำดับ

- Indicator column คือตัวบ่งบอกว่าจะให้ข้อมูล whois นั้นดูจากหัวข้ออะไรเป็นหลัก
- type คือชื่อของประเภท config ที่ต้องการในไปใส่ใน metron
- extractor คือรูปแบบไฟล์ที่ต้องการดึงออกมา ในที่นี้เป็น CSV

ไฟล์ตั้งค่าต่อมาเป็นไฟล์ตั้งค่าว่าต้องการให้ Topic Name ไหนมี Enrichment และต้องการนำค่าไหนมาเชื่อมกับข้อมูล Whois ในที่นี้คือ domain without subdomain และจะให้โหลดข้อมูลการตั้งค่าไปที่ host ไหน โดย host นี่โหลดข้อมูลไปใช้คือ host ของ zookeeper ตามรูปที่ 4.8

โปรแกรมที่ 4.8 ไฟล์ตั้งค่า Whois ส่วนสอง

```
{
      "zkQuorum" : "$ZOOKEEPER_HOST:2181"
     ,"sensorToFieldList" : {
           "squid" : {
              "type" : "ENRICHMENT"
             ,"fieldToEnrichmentTypes" : {
                  "domain_without_subdomains" : [ "whois" ]
               }
           }
     }
}
```

ต่อมาการตั้งค่าไฟล์ของ Blacklist เว็บไซต์ ส่วนนี้จะคล้ายกับ Whois คือ ตั้งค่าการดึงข้อมูลจากไฟล์ และส่วนสองคือการนำข้อมูลโหลดเข้าไปใน Metron และจะใช้ข้อมูลในไหนในการตรวจสอบกับข้อมูล Blacklist

โปรแกรมที่ 4.9 ไฟล์ตั้งค่า Blacklist ส่วนแรก

```
{
 "config" : {
    "columns" : {
        "domain" : 0
        ,"source" : 1
    }
    ,"indicator_column" : "domain"
    ,"type" : "zeusList"
    ,"separator" : ","
  }
  ,"extractor" : "CSV"
}
```

โดยทำการดึงข้อมูล Column คือข้อมูล Domain , type ที่ต้องการตั้งชื่อและรูปแบบไฟล์ที่ต้องการดึงออกมาในที่นี้คือ CSV

ไฟล์ตั้งค่าต่อมาเป็นไฟล์ตั้งค่าว่าต้องการให้ Topic Name ไหนมี Enrichment และต้องการนำค่าไหนมาเชื่อมกับข้อมูล Blacklistในที่นี้คือ domain without subdomain และจะให้โหลดข้อมูลการตั้งค่าไปที่ host ไหน โดย host นี่โหลดข้อมูลไปใช้คือ host ของ zookeeper ตามรูปที่ 4.10

ในส่วนของ Type นั้นการตั้ง Threat intel นั้นคือหากตรงเงื่อนไขหรือก็คือเว็บไซต์ Blaclist นั้นจะทำการแจ้งเตือน

โปรแกรมที่ 4.10 ไฟล์ตั้งค่า Blacklist ส่วนสอง

```
{
      "zkQuorum" : "$ZOOKEEPER_HOST:2181"
     ,"sensorToFieldList" : {
      "squid" : {
            "type" : "THREAT_INTEL"
           ,"fieldToEnrichmentTypes" : {
                 "domain_without_subdomains" : [ "zeusList" ]
            }
      }
    }
}
```

4.2.7 สร้าง Topology ใหม่

หลังจากเราทำการตั้งค่าทุกอย่างสำเร็จหมดแล้วก็สามารถสร้าง Topology ใหม่ได้เลยโดยใช้คำสั่ง

โปรแกรมที่ 4.11 คำสั่งสร้าง Topology

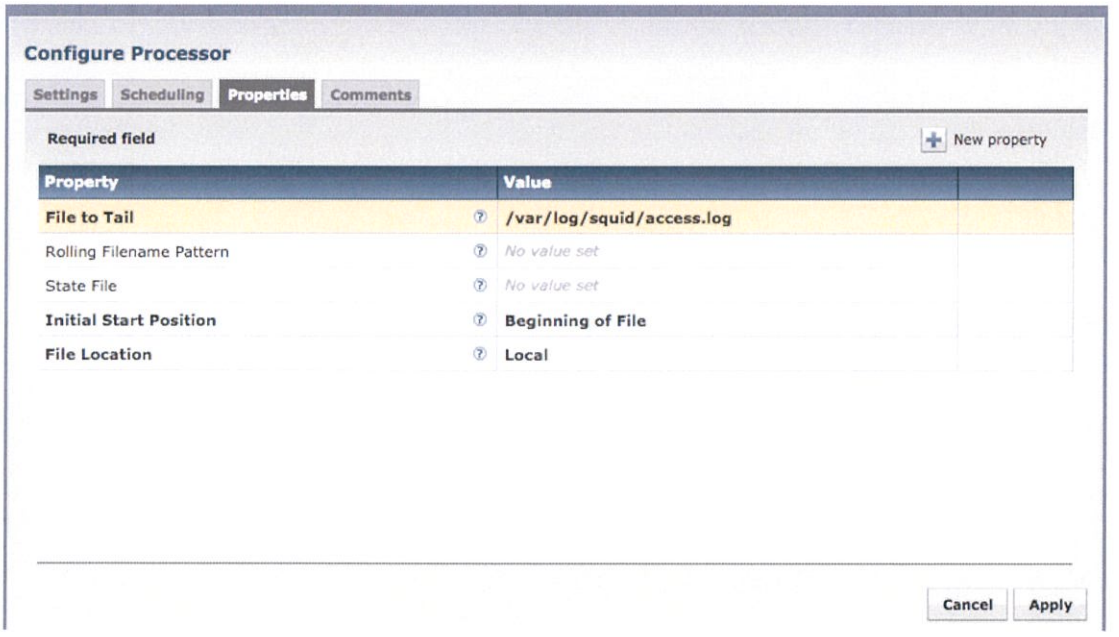
```
/usr/metron/$METRON_VERSION/bin/start_parser_topology.sh -k
$KAFKA_HOST:6667 -z $ZOOKEEPER_HOST:2181 -s squid
```

4.2.8 ใช้ NIFI ในการส่งผ่านข้อมูลโดยอัตโนมัติ

ใช้ Nifi ในการส่งผ่านข้อมูลอัตโนมัติระหว่างระบบซึ่ง Nifi สามารถเก็บข้อมูลและนำไปใช้ลงใน Metron ได้เลยโดยมีวิธีการติดตั้งและตั้งค่าต่อไปนี้

- ดาวน์โหลด Nifi
- แก้ไขไฟล์ conf/nifi.properties เพื่อ Update Port ของ Nifi เป็น port 8089

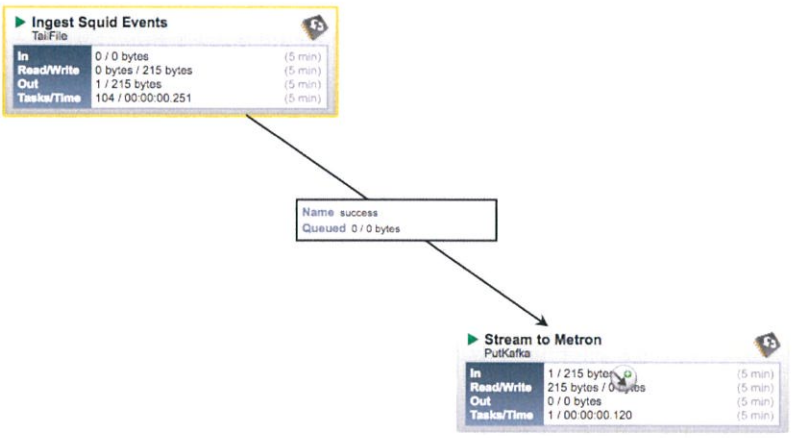
โดยส่วนแรกจะทำการสร้าง Process สำหรับการส่ง Log มาโดยอัตโนมัติซึ่งในส่วนของการตั้งค่านั้นเลือก File to Tail เป็น Path ของ Log ที่ต้องการจะนำส่งข้อมูลให้แก่ Metron



รูปที่ 4.2 หน้าจอ Config Processor ของ Nifi

ส่วนต่อมาทำการสร้าง Process ที่นำข้อมูลเข้าไปยัง Kafka Topic เพื่อประมวลผล โดยทำการตั้งค่า 3 อย่างคือ

- Known Brokers : node1:6667
- Topic Name : ชื่อของ Topic ที่ตั้ง
- Client Name : ชื่อของ Process ที่ตั้ง



รูปที่ 4.3 หน้าจอ Nifi ที่รับ Log เพื่อส่งผ่านไปยัง Metron

เมื่อมีการสร้าง Log ขึ้นมาใน Path ที่ตั้งค่าไว้จะสังเกตเห็นได้ว่ามีข้อมูลถูกส่งผ่านไปยัง Metron

4.2.9 ตรวจสอบว่า Log นั้นแสดงผลได้หรือไม่

Log ที่ถูก Index นั้นจะมีชื่อ {Topic Name}_Index_[Timestamp] และสามารถเรียกดูได้ใน Elasticsearch

node1:9200/_cat/indices?v

health	status	index	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	squid_index_2017.11.23.18	1	0	10	0	20.2kb	20.2kb
green	open	bro_index_2017.11.23.19	1	0	9590	0	11.9mb	11.9mb
green	open	error_index_2017.11.23.18	1	0	15	0	693.2kb	693.2kb
green	open	bro_index_2017.11.23.17	1	0	3720	0	4.8mb	4.8mb
green	open	bro_index_2017.11.23.18	1	0	9600	0	11.9mb	11.9mb
green	open	.kibana	1	0	54	0	68.7kb	68.7kb
green	open	snort_index_2017.11.23.17	1	0	3630	0	3.3mb	3.3mb
green	open	bro_index_2017.11.23.20	1	0	9750	0	12mb	12mb
green	open	snort_index_2017.11.23.18	1	0	9620	0	8.4mb	8.4mb
green	open	snort_index_2017.11.23.19	1	0	9610	0	8.4mb	8.4mb
green	open	bro_index_2017.11.23.21	1	0	2560	0	3.4mb	3.4mb
green	open	squid_index_2017.11.23.21	1	0	20	0	38.3kb	38.3kb
green	open	snort_index_2017.11.23.20	1	0	9770	0	8.8mb	8.8mb
green	open	snort_index_2017.11.23.21	1	0	2560	0	2.5mb	2.5mb

รูปที่ 4.4 Log ที่ถูก Index ไว้ใน Elasticsearch

4.3 หน้าแสดงผลของ JSON

เป็นหน้าจอแสดงผลผ่าน kibana ซึ่งสามารถเลือกการแสดงผล JSON ที่ถูกสร้างขึ้นมาได้โดยโค้ดที่ได้ออกมาดังต่อไปนี้

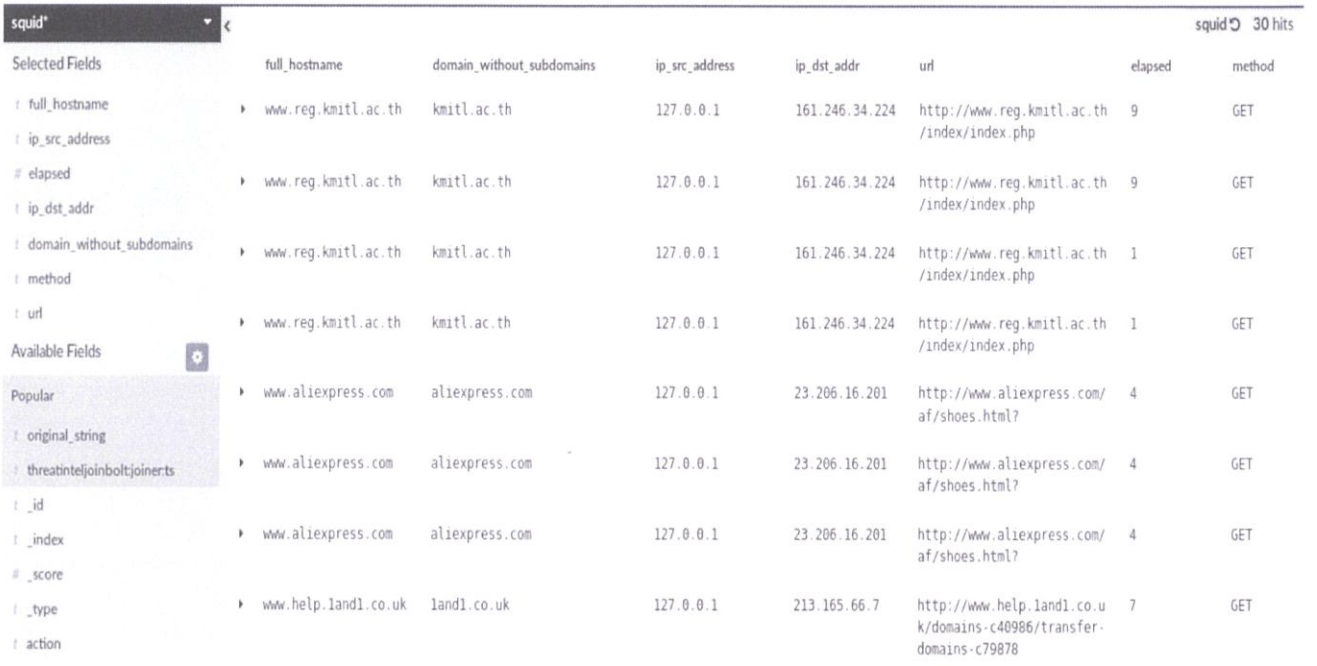
โปรแกรมที่ 4. 5 ไฟล์ JSON ที่ถูกสร้างขึ้นมา

```
{
 "_index": "squid_index_2017.11.23.21",
 "_type": "squid_doc",
 "_id": "AV_qvHbOYyL2rRLjfrUI",
 "_score": null,
 "_source": {
      "full_hostname": "www.reg.kmitl.ac.th",
      "code": 200,
      "ip_src_address": "127.0.0.1",
      "method": "GET",
      "threatinteljoinbolt:joiner:ts": "1511471740544",
      "enrichmentsplitterbolt:splitter:end:ts": "1511471740537",
      "enrichmentsplitterbolt:splitter:begin:ts": "1511471740534",
      "enrichmentjoinbolt:joiner:ts": "1511471740539",
      "url": "http://www.reg.kmitl.ac.th/index/index.php",
      "elapsed": 9,
      "source:type": "squid",
      "ip_dst_addr": "161.246.34.224",
      "original_string": "1511471714.519   1159 127.0.0.1 TCP_MISS/200
110714 GET http://www.reg.kmitl.ac.th/index/index.php -
DIRECT/161.246.34.224 text/html",
      "bytes": 110714,
      "domain_without_subdomains": "kmitl.ac.th",
      "action": "TCP_MISS",
      "guid": "a30fdc87-cda7-4b74-afa3-42ad2abcbae8",
      "threatintelsplitterbolt:splitter:begin:ts": "1511471740541",
      "threatintelsplitterbolt:splitter:end:ts": "1511471740541",
      "timestamp": 1511471714519
 },
 "sort": [
      1511471714519
 ]
}
```

ซึ่งจะเห็นองค์ประกอบหลักๆ ที่ถูกดึงออกมาตาม Grok Statement และ field ใหม่ที่ถูกสร้างใน Grok Parser ที่เขียนไว้

4.4 หน้าแสดงผล UI Dashboard

Kibana มีส่วนที่สามารถแสดง Dashboard และเลือกส่วนที่ต้องการแสดงเฉพาะได้โดยจะสามารถเลือก Field ที่ถูกดึงมาจาก Grok Parser ได้อย่างอิสระว่าอยากเลือกส่วนไหนให้แสดงผลโดยเลือกให้แสดงผล Full_hostname , domain_without_subdomains , ip_src_address , ip_dst_addr และ URL ซึ่งตัวเลือกเหล่านี้จะสามารถเลือกได้จากการ Attribute ที่มีจาก Log อยู่แล้ว และ Attribute ที่ตั้งค่าใน Parser



รูปที่ 4.5 หน้าจอแสดงผลของ Log ที่รับมา

enrichments:hbaseEnrichment:domain_without_subdomains:whois:owner	enrichments:hbaseEnrichment:domain_without_subdomains:whois:
Yahoo! Inc.	Domain Administrator
Yahoo! Inc.	Domain Administrator
Cisco Technology Inc.	Info Sec
Cisco Technology Inc.	Info Sec
.	.
.	.
.	.
.	.
.	.
1 & 1 Internet Ltd	Domain Admin

รูปที่ 4.6 หน้าจอแสดงข้อมูล Whois

	domain_without_subdomains	ip_dst_addr	ip_src_address	url	threatintels:h
▸	yahoo.com	98.137.246.7	127.0.0.1	http://www.yahoo.com/	-
▸	yahoo.com	98.137.246.7	127.0.0.1	https://www.yahoo.com/	-
▸	cnn.com	151.101.129.67	127.0.0.1	https://edition.cnn.com/	-
▸	cnn.com	151.101.9.67	127.0.0.1	http://www.cnn.com/	-
▸	cnn.com	151.101.65.67	127.0.0.1	http://cnn.com/	-
▸	cisco.com	173.222.151.190	127.0.0.1	http://www.cisco.com/	-
▸	cisco.com	104.122.16.62	127.0.0.1	http://www.cisco.com/	-
▸	cisco.com	72.163.4.185	127.0.0.1	http://cisco.com/	-
▸	google.com	74.125.200.103	127.0.0.1	http://www.google.com/	-
▸	google.com	74.125.200.103	127.0.0.1	http://www.google.com/	-
▸	google.com	74.125.200.139	127.0.0.1	http://google.com/	-
▸	vegantravelshow.com	209.191.185.67	127.0.0.1	http://vegantravelshow.com/	alert
▸	vegantravelshow.com	209.191.185.67	127.0.0.1	http://vegantravelshow.com/	alert
▸	vegantravelshow.com	209.191.185.67	127.0.0.1	http://vegantravelshow.com/	alert
▸	vegantravelshow.com	209.191.185.67	127.0.0.1	http://vegantravelshow.com/	alert

รูปที่ 4.7 หน้าจอแสดงการแจ้งเตือนเว็บไซต์ Blacklist

ในรูปที่ 4.5 ในแถบด้านซ้ายมือจะมีตัวเลือก Field และ Field ที่ถูกเลือก ซึ่งสามารถเลือกได้ตามต้องการว่าต้องการให้แสดงผลอะไร ซึ่งจะเห็นว่าสามารถตรวจสอบได้ว่าเครื่องคอมพิวเตอร์เครื่องไหนเข้าเว็บไซต์อะไร ซึ่งสามารถเห็นได้ตามตัวอย่าง ว่ามีการเข้าถึงเว็บไซต์ www.reg.kmitl.ac.th โดยมี IP ปลายทาง (ip_dst_addr) บ่งบอกอยู่ด้วยและมีการบ่งบอก URL ที่มีการเข้าถึงและวิธีในการเข้าถึงเว็บไซต์ ซึ่งสิ่งต่างๆเหล่านี้เป็นระบบพื้นฐานของการตรวจจับเหตุการณ์ต่างๆ ใน SIEM

ในรูปที่ 4.6 นั้นจะแสดงให้เห็นถึงตัวอย่างของข้อมูล Whois ซึ่งจะเห็นชื่อบริษัทและนายทะเบียนที่รับการจดทะเบียน

ในรูปที่ 4.7 นั้นจะแสดงให้เห็นถึงการแจ้งเตือนให้แก่ผู้ใช้งานเมื่อมีการเข้าเว็บไซต์ Blacklist โดยการแจ้งเตือนจะขึ้นว่า alert เป็นการบ่งบอก

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุป

- มีการศึกษาโครงสร้างของ Apache Metron ในหลายๆ ส่วน ว่ามีการทำงานตั้งต้นของ Metron อย่างไรใช้ส่วนประกอบอะไรบ้างในการสร้าง Metron ขึ้นมา และศึกษาว่าส่วนแต่ละส่วนประกอบนั้นทำงานอย่างไร
- สามารถแสดงผลตามมาตรฐาน SIEM ได้อย่างเช่นการแสดงผล Hostname , domain , Source IP , Destination IP และ URL ซึ่งค่าพวกนี้จะแตกต่างกันไปตามอุปกรณ์ที่ไปรับ Log มาและการตั้งค่าในส่วน Parser ของ Metron

5.2 อุปสรรคในการดำเนินงาน

- เอกสารที่เกี่ยวข้องกับ Apache Metron นั้นค่อนข้างมีจำนวนน้อยทำให้มีอุปสรรคในบางส่วนเมื่อต้องการจะศึกษาว่าส่วนประกอบบางส่วนนั้นทำงานอย่างไร ทำให้ใช้เวลาในการศึกษาส่วนเหล่านี้ค่อนข้างนาน
- ภาษาที่ใช้ในการทำงานใน Apache Metron เช่น Grok และ JSON เป็นสิ่งที่ไม่เคยศึกษามาก่อนทำให้ใช้เวลานานในการทำความเข้าใจ
- เอกสารหลายส่วนของ Apache Metron นั้นไม่ได้มีการอัพเดทให้ตรงตามเวอร์ชันของโปรแกรมทำให้ได้ข้อมูลศึกษาไม่ตรงตามต้องการ
- เอกสารในส่วนของการพัฒนานั้นมีอยู่ไม่มากส่วนใหญ่นั้นจะมีแต่การสอนการใช้งานเป็นส่วนใหญ่
- เอกสารของส่วนผู้พัฒนาหรือ API นั้นไม่มีอยู่ และตัว โปรแกรมที่เป็นส่วนประกอบของ Metorn นั้นมีเยอะมาก ทำให้การที่จะทดลองพัฒนาจากเครื่องที่มีสเปคของ Server ที่ใหญ่มากเพื่อทำการใช้งาน metron ความรู้ที่เกิดจากการพัฒนาของ Metron นั้นต้องเกิดจากการเข้าไปศึกษาในโค้ด ท่านั้นทำให้ใช้เวลายาวนานในการเรียนรู้ส่วนประกอบและหลักการทำงานของ Apache Metron
- เซิฟเวอร์จากทางของผู้ให้ความร่วมมือในการจัดการพัฒนานั้นเกิดปัญหา ทำให้ต้องแก้ไขและย้าย Environment ในช่วงใกล้ส่งถึงเวลาส่ง

- Apache Metron เป็น Legacy Software ที่หลายตัว Out-date ไปแล้ว ในการติดตั้งหรือ Deploy อาจจะเกิดปัญหาที่เกิดขึ้นกับเครื่อง Host ที่จะใช้งาน
- ในการติดตั้ง Apache Metron ตัวโปรแกรมต่างๆที่เป็นส่วนประกอบนั้นมี dependency ที่หลากหลาย ทำให้ในบางครั้งเกิดข้อผิดพลาดในการติดตั้งและจำเป็นต้องจัดการกับข้อผิดพลาดนั้นเองเนื่องจากไม่มีการแจ้งเตือนต่างๆ

บรรณานุกรม

[1] Apache Metron. 2016. **Official Site of Apache Metron**. [Online]

Available: metron.apache.org

[2] Apache Metron Documentation. 2016. **Official Documents Website of Apache Metron**. [Online].

Available: https://cwiki.apache.org/confluence/display/METRON/Documentation

[3] Apache Storm. 2016. **Apache Storm Git Repository**. [Online]

Available: https://github.com/apache/storm

[4] Hortonworks Data Platform. 2017. **Apache Storm Component Guide**. [Online]

Available:https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.2/bk_storm-component-guide/bk_storm-component-guide.pdf

[5] Hortonworks. 2016. **Apache Metron Project**. [Online]

Available: https://hortonworks.com/apache/metron/

[6] Horton Community. 2016. **Apache Metron Community**. [Online]

Available: https://community.hortonworks.com/topics/Metron.html

[7] JSON. **Official Site of JSON**. [Online]

Available: https://www.json.org/

[8] Microsoft Azure. 2016. **What is a virtual machine**. [Online]

Available: https://azure.microsoft.com/en-us/overview/what-is-a-virtual-machine/

[9] Vagrant. 2016. **Vagrant Official Site**. [Online]

Available: https://www.vagrantup.com/

[10] VirtualBox. 2016**. VirtualBox Official Site**. [Online]

Available: https://www.virtualbox.org/

[11] VMware. 2016. **What is a virtual machine.** [Online].

Available:https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.vm_admin.doc%2FGUID-CEFF6D89-8C19-4143-8C26-4B6D6734D2CB.html

[12] Hortonworks Cybersecurity Package Run Book. [Online].

Available:https://docs.hortonworks.com/HDPDocuments/HCP1/HCP-1.4.0/bk_runbook/bk_runbook.pdf