



การศึกษาระบบวิเคราะห์ และ แจ้งเตือนความผิดปกติจาก Log ในระบบสารสนเทศด้วย

**Line และ Discord**

(SIEM and Notification for Logging System)(SIEM and Notification for Logging System)

**วิชา INFORMATION TECHNOLOGY INFRASTRUCTURE SECURITY**

จัดทำโดย

นายกิตติชินทักษิ ธรรมานนท์ ใช้ติ รหัสนักศึกษา 65070018

นายณัฐนนท์ วงศ์หน่องแวง รหัสนักศึกษา 65070076

นายณัฐพงศ์ มาสำราญ รหัสนักศึกษา 65070078

เสนอ

ผศ.ดร. สุเมธ ประภาวัต

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา

**INFORMATION TECHNOLOGY INFRASTRUCTURE SECURITY**

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง

หัวข้อโครงการ การศึกษาระบบวิเคราะห์ และ แจ้งเตือนความผิดปกติจาก Log ในระบบสารสนเทศ  
ด้วย Line และ Discord

ผู้จัดทำ นายกิตติชินทักษิ บรรยานนท์โชติ รหัสนักศึกษา 65070018

นายณัฐนันท์ วงศ์หน่องแวง รหัสนักศึกษา 65070076

นายณัฐพงษ์ มาสำราญ รหัสนักศึกษา 65070078

ปีการศึกษา 2567

## บทคัดย่อ

ในปัจจุบันระบบสารสนเทศเป็นสิ่งจำเป็นในการประกอบกิจการสำคัญต่างๆ มากมาย เนื่องจาก  
ระบบสารสนเทศมีความซับซ้อนมากจึงมีการเก็บ Log เพื่อสามารถตรวจสอบเช็คการกระทำการผิดปกติหรือข้อมูลการ  
กระทำการซ่อนหลังเพื่อลดระดับความซับซ้อนจึงต้องทำการวิเคราะห์เพื่อให้เข้าใจง่ายยิ่งขึ้น และ เนื่องจากการจะ  
ตรวจสอบ Log ไม่สามารถทำได้ตลอดเวลา ระบบการแจ้งเตือนความผิดปกติจาก Log จะสามารถแจ้งเตือนการ  
กระทำการที่อาจมีอันตรายจากการ โจมตีทางไซเบอร์ทำให้สามารถไปจัดการและรับมือได้อย่างทันท่วงที โดย  
ปัจจุบันมีการทำระบบแจ้งเตือนออกรอบแล้ว แต่ในการศึกษานี้จะมุ่งเน้นไปที่การแจ้งเตือนไปที่โปรแกรม Line  
และ Discord เพราะสามารถเข้าถึงได้ง่ายทำให้รู้ตัวได้ไวกว่าเมื่ออุบัติเหตุ

## สารบัญ

บทที่ 1.....	5
ที่มาของโครงการ .....	5
วัตถุประสงค์ .....	5
ขอบเขตของระบบ.....	5
ประโยชน์ที่คาดว่าจะได้รับ.....	6
ความเป็นไปได้ของโครงการ .....	6
บทที่ 2.....	9
งานวิจัยที่เกี่ยวข้อง.....	9
บทที่ 3.....	11
เทคโนโลยีที่เกี่ยวข้อง .....	11
แผนการดำเนินงาน.....	15
บทที่ 4.....	17
ขั้นตอนการดำเนินงาน.....	17
Topology .....	17
การตั้งค่าสำหรับ Network .....	17
การติดตั้งและตั้งค่า Program และ Service ต่างๆ .....	18
ภาษาใน GNS3 .....	18
ภาษานอก GNS3 .....	22
การตั้งค่า SIEM เพื่อ Alert ไปยัง Line.....	23
การตั้งค่า SIEM เพื่อ Alert ไปยัง Discord.....	27
บทที่ 5.....	33

ผลการทดลองและปัญหาที่เกิดขึ้น.....	33
ผลการทดลอง Alert ไปยัง Line .....	33
ผลการทดลอง Alert ไปยัง Discord.....	35
.....	2
บรรณานุกรม .....	37

## บทที่ 1

### ที่มาของโครงงาน

ในสมัยนี้ การรักษาความปลอดภัยของข้อมูลนั้นเป็นสิ่งที่สำคัญมาก เพราะสิ่งนี้เป็นเกราะป้องกันข้อมูล และ แสดงถึงความสามารถในการป้องกันข้อมูลซึ่งทำให้เกิดความเชื่อมั่นต่อองค์กร ดังนั้นในหลายบริษัท และ องค์กร จึงมีแผนกที่คุ้มครองข้อมูล หรือ IT Security

ในการทำงานนี้ส่วนใหญ่ผู้ดูแลจะทำการใช้โปรแกรม Monitoring เพื่อตรวจสอบหาจุดที่น่าสงสัยในเครือข่าย เช่น มีการทำ DDOS การเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต โดยภัยต่างๆ เหล่านี้เป็นอันตรายต่อองค์กรเป็นอย่างมาก ผู้ดูแลจึงใช้เวลาอย่างมากในการดูโปรแกรม Monitor ซึ่งอาจก่อให้เกิดความเหนื่อยล้า และยากต่อการสังเกตุโดยระบบวิเคราะห์และสรุปผลสามารถลดความซับซ้อน และ ความผิดพลาดลงได้

ตามมาด้วยความผิดพลาดหากผู้ดูแลคาดคะเนผิดพลาด ทางเราเล็งเห็นปัญหานี้ และ ได้ตั้งเป้าหมายที่จะทำการดูแล สามารถทำได้โดยไม่จำเป็นต้องดูโปรแกรม Monitor เพียงอย่างเดียว จึงเกิดเป็นโครงงาน Notification For Logging System ที่จะให้การดูแลไม่ใช่แค่การดู Monitor เพียงอย่างเดียว แต่เพิ่มการ Notify หรือแจ้งเตือนให้กับผู้ดูแล หากเกิดเหตุไม่คาดคิด ทำให้ผู้ดูแลรักษาความปลอดภัยได้อย่างมีประสิทธิภาพ

### วัตถุประสงค์

- เพื่อสร้างระบบ notification จาก Log Management System ที่สามารถแจ้งเตือนไปยัง Line และ Discord ได้
- เพิ่มระบบการแจ้งเตือนด้วยเสียงใน Discord ด้วย bot
- เพิ่มระบบวิเคราะห์ Log ก่อนนำมาแจ้งเตือน

### ขอบเขตของระบบ

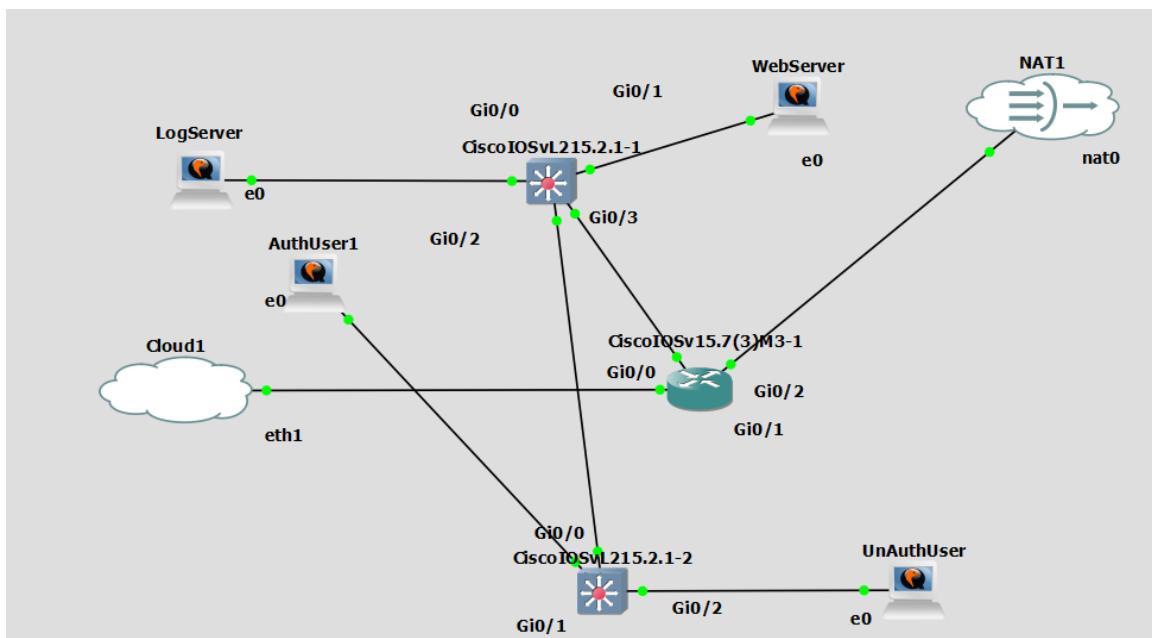
1. สามารถรวม Log จาก Snort มาแสดงที่ SIEM Dashboard ได้
- สามารถนำ Log มาวิเคราะห์ และ ส่งออกได้
  - มีการแจ้งเตือนเมื่อ Log เกิดความผิดปกติไปที่ Line และ Discord ผ่าน Chat Text ได้
  - Bot Discord สามารถเข้ามาแจ้งเตือนผู้ใช้งานที่อยู่ใน Voice Channel ได้ด้วยเสียง

## ประโยชน์ที่คาดว่าจะได้รับ

2. ได้เรียนรู้และทดลองติดตั้งระบบ monitoring log ในส่วนต่างๆ
  - ได้ศึกษาทดลองใช้ IDS เช่น snort และการทำ SIEM
  - ประยุกต์ความรู้จากการเรียนในรายวิชามาใช้กับระบบ notification
  - ได้ศึกษาค้นคว้าการทำระบบแจ้งเตือนจาก Log ไปยัง Line และ Discord
  - ได้เรียนรู้การสร้าง Bot ใน Discord สำหรับนำไปต่อรองระบบ notification
  - ความรู้ด้านการอ่าน Log dashboard จากการทดลอง

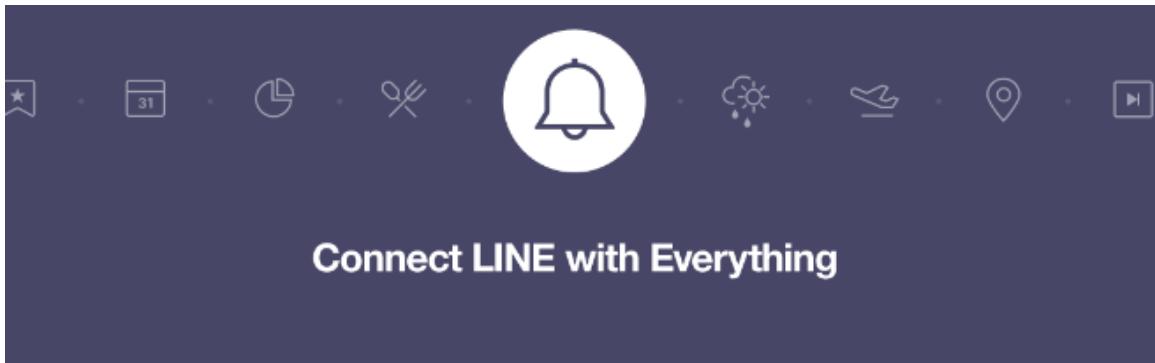
## ความเป็นไปได้ของโครงงาน

โดยการทำโครงงานจะนำผลลัพธ์จากการทดลองแลป 2b จากรายวิชา INFORMATION TECHNOLOGY INFRASTRUCTURE SECURITY มาพัฒนาต่อโดยในแลป 2b มีการทดลองเปิด Webserver และการทำ Port security เพื่อป้องกันการเข้าสู่ระบบจากผู้ไม่ได้รับอนุญาต



โดยในโครงงานนี้จะนำ Log ความปลอดภัยของ Snort มาเพื่อแสดง Dashboard ของ Log จากนั้นจึงนำข้อมูลจาก Dashboard มาเขียน Automation เพื่อแจ้งเตือนเข้าไปที่ระบบ Line Notification

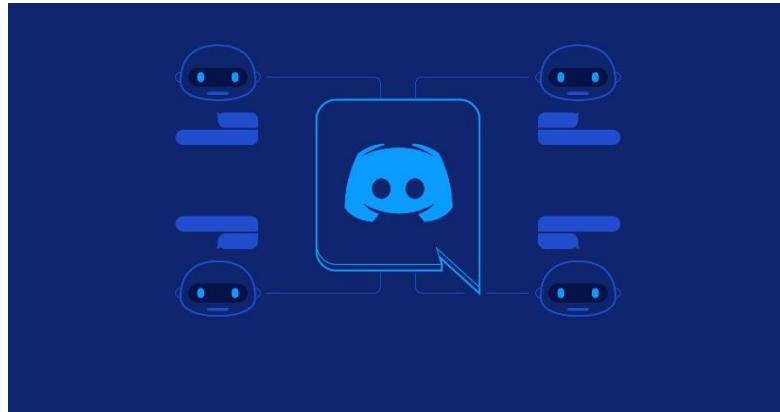
The screenshot shows the Splunk Enterprise web interface. At the top, there's a navigation bar with links for Administrator, Messages, Settings, Activity, Help, and a search bar. Below the navigation is a sidebar titled "Hello, Administrator" with sections for Apps, Bookmarks, and a search bar. The main content area is titled "CPU Usage: Instance" and includes a sub-header: "This dashboard consolidates panels from other Monitoring Console dashboards to help you assess the CPU usage of a Splunk instance." There are several filter dropdowns: Role (All), Group (All), and Instance (jammy). A message below the filters says "Search produced no results." Below the filters are three buttons: "Select views: All", "Snapshot", and "Historical". The main panel is titled "Snapshots" and contains a table for "CPU Usage". The table has columns for Instance, Role, CPU Cores (Physical / Virtual), CPU Usage (%), and Load Average. One row is shown for the "jammy" instance, which is an Indexer with 1 core, using 8.83% CPU and having a load average of 1.85. A large green bar chart represents the CPU Usage over Splunk Process Type.



## รับการแจ้งเตือนจากเว็บเซอร์วิสทาง LINE

หลังจากติดต่อกับเว็บเซอร์วิสแล้ว คุณจะได้รับการแจ้งเตือนจากบัญชีทางการ "LINE Notify" ซึ่งให้บริการโดย LINE  
คุณสามารถเชื่อมต่อกับบริการที่หลากหลาย และรับการแจ้งเตือนทางกลุ่มได้ด้วย

และอีกรอบที่ทางเราอยากรำลึกถึงคือ การแจ้งเตือนผ่านทาง Discord Bot



สมมุติว่าเราอยู่ในห้อง security room เราให้คอมเครื่องนึงเข้า discord และเปิดลำโพงไว้แล้วนำ bot ที่เราสร้างขึ้นมา ที่นี่เราจะไม่จำเป็นต้องนั่งดู monitor dashboard ให้มือยตา สามารถนั่งเล่นเกมอย่างมีความสุข หากเกิดปัญหา โดยโจนใจมี Bot ในห้อง discord จะส่งเสียงร้องออกมาก เช่น Warning, Attack ฯลฯ และให้ข้อมูลอย่างละเอียดในห้องข้อความของ discord

แนวทางที่วางแผนคือนำ log ที่ได้มาอ่าน โดยใช้ regular expression หา pattern log ที่น่าสนใจ จากนั้น trigger ให้ bot ออกเสียง

## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

การสร้าง bot discord (**Code a Discord Bot with Python - Host for Free in the Cloud**)

<https://youtu.be/SPTfmiYiuok>



เกี่ยวกับ SIEM ตอนที่ 1 รู้จักระบบ SIEM เพื่อจัดการและตรวจสอบเหตุการณ์ของระบบเครือข่าย – EZ-GENIUS

เกี่ยวกับ snort IDS Install Snort on window. มาแน่นำและติดตั้ง Snort กันดีกว่า | by Rachata Tongpagdee | Medium



## บทที่ 3

### เทคโนโลยีที่เกี่ยวข้อง

1.GNS3 (สำหรับจำลอง Webserver เพื่อเก็บ Log)



GNS3 (Graphical Network Simulator-3) คือซอฟต์แวร์โอเพ่นซอร์สที่ออกแบบมาเพื่อจำลอง และสร้างระบบเครือข่ายในสภาพแวดล้อมเสมือนจริง โดยไม่ต้องใช้อุปกรณ์จริง ๆ ทำให้ผู้ใช้งานสามารถทดสอบและวิเคราะห์เครือข่ายได้อย่างละเอียดและปลอดภัย

2.VM Workstation Pro (สำหรับ Run GNS3 VM)



VMware Workstation Pro คือซอฟต์แวร์สำหรับการจำลองเครื่องเสมือน (Virtual Machine)

ที่พัฒนาโดยบริษัทVMware โดยซอฟต์แวร์นี้ช่วยให้ผู้ใช้งานสามารถรันระบบปฏิบัติการหลายระบบบนคอมพิวเตอร์เครื่องเดียว ได้พร้อมกันทำให้สามารถทดสอบและพัฒนาซอฟต์แวร์ในสภาพแวดล้อมที่แตกต่างกันโดยไม่ต้องใช้ฮาร์ดแวร์หลายเครื่อง

### 3. Line Notify



LINE Notify คือบริการจาก LINE Corporation ที่ช่วยให้ผู้ใช้สามารถส่งการแจ้งเตือนไปยังกลุ่ม หรือบุคคลผ่านแอป LINE ได้อย่างง่ายดาย โดยไม่ต้องเขียนโค้ดหรือใช้ API ที่ซับซ้อน

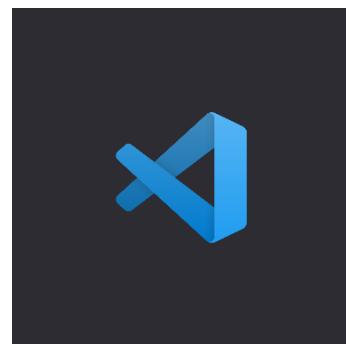
### 4. Discord API



Discord API คือชุดเครื่องมือที่ Discord มอบให้สำหรับนักพัฒนาในการเชื่อมต่อกับแพลตฟอร์ม Discord ผ่านโปรแกรมหรือโค้ดต่าง ๆ โดย Discord API

ช่วยให้นักพัฒนาสามารถสร้างบทที่ทำงานอัตโนมัติในเซิร์ฟเวอร์ Discord หรือรวม Discord เข้ากับแอปพลิเคชันอื่น ๆ ได้

### 5. Visual Studio Code



VS Code หรือ Visual Studio Code คือโปรแกรมแก้ไขโค้ด (code editor) ที่ได้รับความนิยมสูงมากตัวหนึ่ง เป็นโปรแกรมฟรีและโอเพนซอร์ส พัฒนาโดยไมโครซอฟท์

- ใช้งานง่าย: ใช้งานได้รวดเร็ว มีอินเทอร์เฟซที่เข้าใจง่าย
- ปรับแต่งได้: สามารถปรับแต่งให้เข้ากับสไตล์การทำงานของแต่ละคนได้
- มีปลั๊กอินมากมาย: เพิ่มฟังก์ชันต่างๆ ได้ตามต้องการ
- รองรับหลากหลายภาษา: ใช้ภาษารีดได้หลากหลาย

#### 6.Splunk Security Enterprise (SIEM Free Trial 60 days)



Splunk กือ แพลตฟอร์มสำหรับการ รวบรวมข้อมูลจากหลายๆแหล่งแบบไม่มีโครงสร้าง  
สามารถเก็บข้อมูลได้หลายแหล่งมากๆและนำมายังเคราะห์ต่อยอดได้ง่าย โดยเราจะใช้  
Splunk ในการ Alert Security Event ที่อาจเป็นภัยต่อระบบเรา ไปยังภายนอก เช่น Line,  
Discord เพื่อการตอบสนองต่อเหตุการณ์ความปลอดภัยได้อย่างรวดเร็วและมี Dashboard  
อยู่สรุปสิ่งต่างๆที่เกิดขึ้น

#### 7.Prometheus



Prometheus คือระบบการตรวจสอบและเก็บข้อมูลที่พัฒนาขึ้น โดย SoundCloud และปัจจุบันได้รับการสนับสนุนโดย Cloud Native Computing Foundation (CNCF) โดย Prometheus ถูกออกแบบมาเพื่อเก็บรวบรวมและจัดการข้อมูลเมตริก (metrics)

ซึ่งเป็นข้อมูลที่บอกรสภาพการทำงานของระบบต่างๆ เช่น CPU usage, memory consumption, network I/O และอื่น ๆ

#### 8.Grafana



Grafana เป็นซอฟต์แวร์โอเพ่นซอร์สที่ใช้ในการแสดงผลข้อมูลเชิงวิเคราะห์และสร้างแดชบอร์ด (dashboards) ที่มีประสิทธิภาพสำหรับข้อมูลที่ได้จากหลากหลายแหล่งข้อมูล เช่น Prometheus, InfluxDB, Elasticsearch, MySQL, และอื่น ๆ

#### 9.Node export

**Node export** คือการส่งออกข้อมูลจากไฟล์หนึ่งใน Node.js เพื่อให้สามารถนำไปใช้ในไฟล์อื่นได้กระบวนการนี้ทำให้เราสามารถแยกไฟล์ออกเป็นส่วนย่อย ๆ และนำกลับมาใช้ใหม่ในไฟล์หรือโมดูลต่าง ๆ ได้ง่ายขึ้น โดยในที่นี้ใช้เพื่อส่งออกสถานะการทำงานและการใช้ทรัพยากรของเครื่อง Webserver

#### 10. Snort

Snort คือระบบตรวจจับการบุกรุก (Intrusion Detection System หรือ IDS) และป้องกันการบุกรุก (Intrusion Prevention System หรือ IPS) ที่ทำงานแบบโอเพ่นซอร์ส (open-source) พัฒนาโดย Cisco (เดิมที่พัฒนาโดย Sourcefire) หน้าที่หลักของ Snort คือการตรวจสอบрафฟิกเครือข่ายแบบเรียลไทม์เพื่อหากิจกรรมที่น่าสงสัย

## 11.Vercel



Vercel เป็นแพลตฟอร์มที่ให้บริการในการพัฒนาและ deploys แอปพลิเคชันเว็บแบบ serverless โดยเน้นความสะดวกในการนำแอปพลิเคชันเข้าสู่การผลิต (production) อย่างรวดเร็วและง่ายดาย รวมถึงการใช้งานเทคโนโลยีที่ทันสมัย เช่น Next.js ซึ่งเป็นเฟรมเวิร์กยอดนิยมสำหรับการสร้างแอปพลิเคชัน React

## แผนการดำเนินงาน

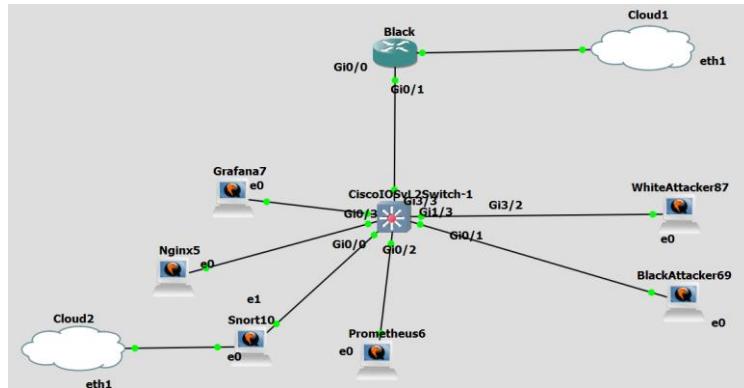
- 1) GNS 3 สำหรับจำลองระบบ Network ที่มี Nginx Prometheus Grafana Snort และ Attacker ข้างในโดยต้องมีการทำ Route จากเครื่อง Host เพื่อเข้าสู่ Network ภายใน และ ต่อ Nat เพื่อออกมายังนอก
- 2) VM Ware สำหรับจำลอง Computer เพื่อติดตั้ง Splunk  
เนื่องจากความใหญ่ของไฟล์โปรแกรมทำให้ไม่สามารถติดตั้งใน GNS3 ได้
- 3) Webserver และ Node export สำหรับทดสอบเบ็ด Webserver สำหรับรับการโจมตีจาก Attacker  
เพื่อนำไปวิเคราะห์
- 4) Prometheus สำหรับนำข้อมูลจาก Node export ออกรายละเอียดที่ Grafana
- 5) Grafana สำหรับแสดงข้อมูลจาก Prometheus มาแสดงเป็น Dashboard (สามารถ Alert เมื่อเกิดสิ่งผิดปกติได้)  
ใน Project นี้เลือกใช้ Splunk ในการ Alert สิ่งผิดปกติใน Network แทน)
- 6) Snort สำหรับตรวจจับข้อมูลใน Network และ ส่ง Log ไปยัง Splunk
- 7) Splunk สำหรับนำ Log ไปวิเคราะห์ให้ง่ายต่อการอ่าน และ สามารถ Alert ตามเงื่อนไขต่างๆ ไปยัง Line และ Discord ได้

- 8) Discord สำหรับเป็นห้องสนทนาร่วมกันที่สามารถให้ Bot เข้ามา Alert ด้วยเสียงได้ และยังเป็นห้องข้อความร่วมกัน
- 9) Line สำหรับเป็นห้องส่งข้อความที่สามารถให้ Line notify เข้ามาส่งข้อความแจ้งเตือนได้

## บทที่ 4

### ขั้นตอนการดำเนินงาน

#### Topology



#### การตั้งค่าสำหรับ Network

3. Router **gi0/0** ip address 20.8.1.1
4. Router **gi0/1** ip dhcp
5. Cloud Nat port **eth1**
6. ตั้ง route ที่ Host Computer เพื่อเข้าไปยัง Network

IPv4 Route Table					
Network Destination	Netmask	Gateway	Interface	Metric	
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.46	35	
0.0.0.0	0.0.0.0	26.0.0.1	26.71.51.44	9257	
20.8.1.0	255.255.255.0	192.168.137.146	192.168.137.1	36	
26.0.0.0	255.0.0.0	On-link	26.71.51.44	257	
26.71.51.44	255.255.255.255	On-link	26.71.51.44	257	
26.255.255.255	255.255.255.255	On-link	26.71.51.44	257	
127.0.0.0	255.0.0.0	On-link	127.0.0.1	331	
127.0.0.1	255.255.255.255	On-link	127.0.0.1	331	
127.255.255.255	255.255.255.255	On-link	127.0.0.1	331	
192.168.1.0	255.255.255.0	On-link	192.168.1.46	291	
192.168.1.46	255.255.255.255	On-link	192.168.1.46	291	
192.168.1.255	255.255.255.255	On-link	192.168.1.46	291	
192.168.56.0	255.255.255.0	On-link	192.168.56.1	281	
192.168.56.1	255.255.255.255	On-link	192.168.56.1	281	
192.168.56.255	255.255.255.255	On-link	192.168.56.1	281	
192.168.88.0	255.255.255.0	On-link	192.168.88.1	291	
192.168.88.1	255.255.255.255	On-link	192.168.88.1	291	
192.168.88.255	255.255.255.255	On-link	192.168.88.1	291	

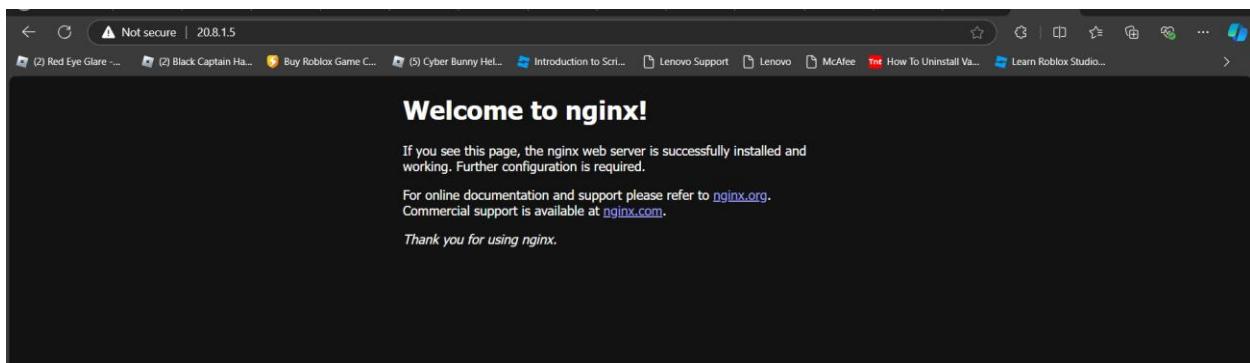
7. ต่อ Cloud Nat port **eth1** เข้าที่ Snort เพื่อส่ง log ออกไปยัง Splunk

## การติดตั้งและตั้งค่า Program และ Service ต่างๆ

### ภายใน GNS3

#### ติดตั้ง Webserver (Nginx) ip address 20.8.1.5

1. sudo apt install nginx
2. sudo systemctl start nginx
8. sudo systemctl enable nginx



#### ติดตั้ง Prometheus ip address 20.8.1.6

1. sudo mkdir /etc/prometheus
2. sudo mkdir /var/lib/Prometheus
3. wget <https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz> (จำเป็นต้องตั้ง nameserver ก่อน)
4. tar -xvf prometheus-2.47.0.linux-amd64.tar.gz
5. sudo cp prometheus-2.47.0.linux-amd64/prometheus /usr/local/bin/
6. sudo cp prometheus-2.47.0.linux-amd64/promtool /usr/local/bin/
7. sudo cp -r prometheus-2.47.0.linux-amd64/consoles /etc/prometheus
8. sudo cp -r prometheus-2.47.0.linux-amd64/console\_libraries /etc/prometheus
9. sudo cp prometheus-2.47.0.linux-amd64/prometheus.yml /etc/prometheus/
10. sudo nano /etc/systemd/system/prometheus.service (นำ config ด้านล่างไปใส่)

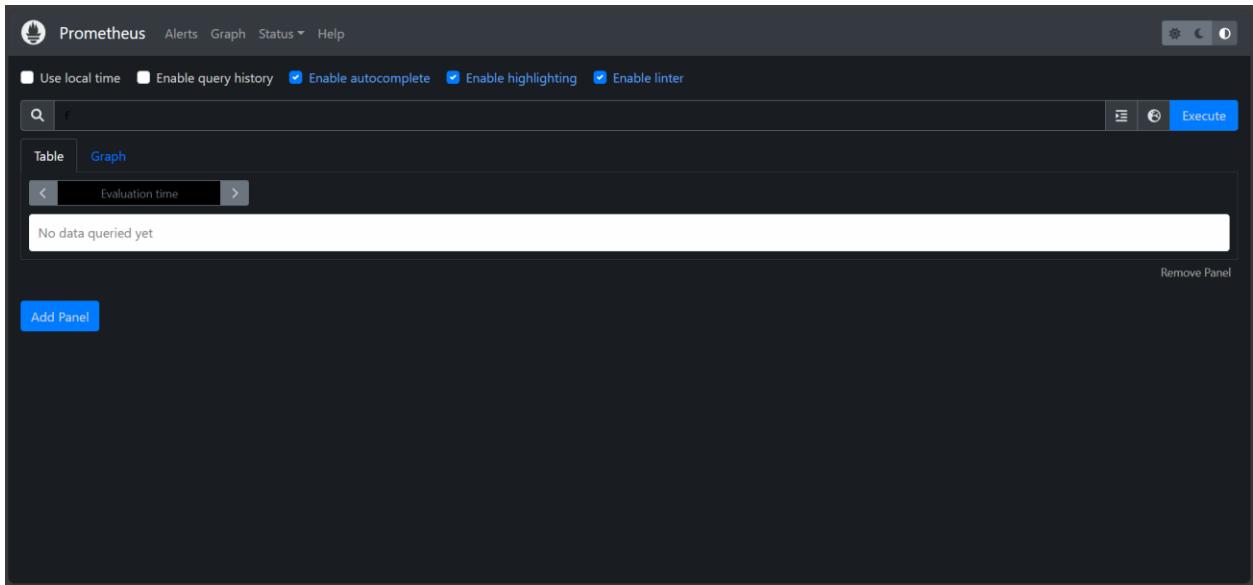
```
[Unit]
Description=Prometheus
After=network.target

[Service]
ExecStart=/usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml
User=nobody

[Install]
WantedBy=multi-user.target
```

9. sudo systemctl daemon-reload
11. sudo systemctl start prometheus
12. sudo systemctl enable Prometheus

ສາມາດເຂົ້າໄຈ Prometheus ຕ່ວຍ <http://20.8.1.6:9090>

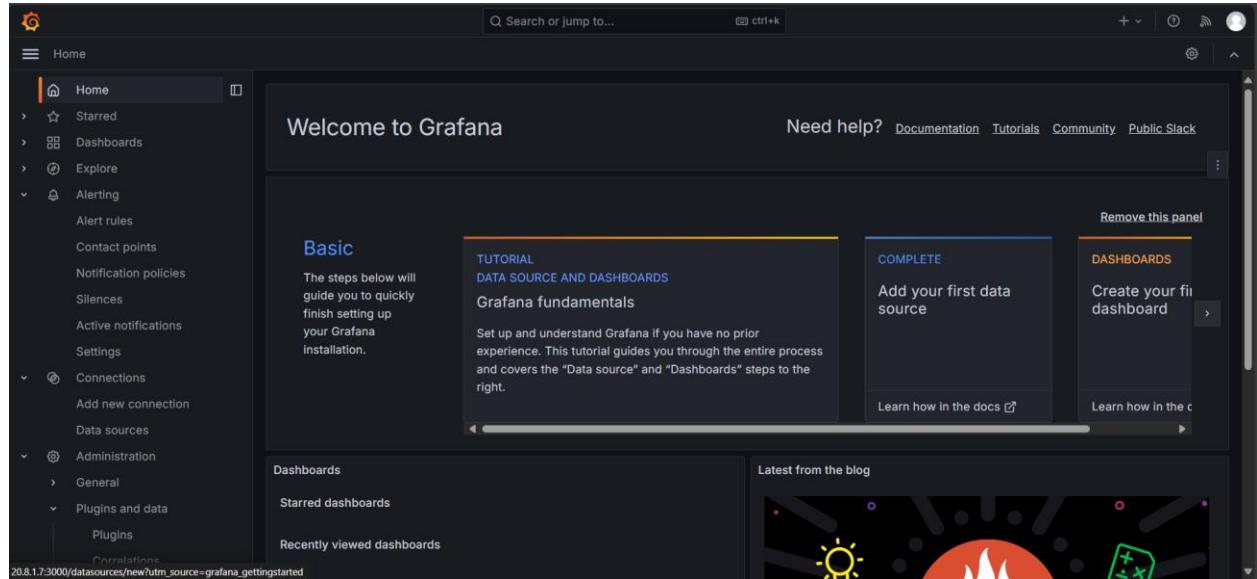


### ຕິດຕັ້ງ Grafana ip address 20.8.1.7

1. sudo apt install -y software-properties-common
2. sudo add-apt-repository "deb <https://packages.grafana.com/oss/deb> stable main"
3. wget -q -O - <https://packages.grafana.com/gpg.key> | sudo apt-key add -
4. sudo apt update
10. sudo apt install grafana -y
5. sudo systemctl start grafana-server

## 6. sudo systemctl enable grafana-server

สามารถเข้าถึง Grafana ด้วย <http://20.8.1.7:3000> username : admin password: admin



ติดตั้ง IDPS (Snort) ip address 20.8.1.10

11. sudo apt install snort

12. sudo nano /etc/snort/snort.conf (ตั้งค่า Network ภายใน ภายนอก)

13. sudo suricata -c /etc/snort/snort.conf -i eth0 (interface ที่เชื่อมต่อ กับ network)

14. sudo systemctl enable suricata

การตั้งค่า Rules ของ Snort !เพื่อ แจ้งเตือนภัย

attack-responses.rules	community-nntp.rules	deleted.rules	netbios.rules	sql.rules
backdoor.rules	community-oracle.rules	dns.rules	nntp.rules	telnet.rules
bad-traffic.rules	community-policy.rules	dos.rules	oracle.rules	tftp.rules
chat.rules	community-sip.rules	experimental.rules	other-ids.rules	virus.rules
community-bot.rules	community-smtp.rules	exploit.rules	p2p.rules	web-attacks.rules
community-deleted.rules	community-sql-injection.rules	finger.rules	policy.rules	web-cgi.rules
community-dos.rules	community-virus.rules	ftp.rules	pop2.rules	web-client.rules
community-exploit.rules	community-web-attacks.rules	icmp-info.rules	pop3.rules	web-coldfusion.rules
community-ftp.rules	community-web-cgi.rules	icmp.rules	porn.rules	web-frontpage.rules
community-game.rules	community-web-client.rules	imap.rules	rpc.rules	web-iis.rules
community-icmp.rules	community-web-dos.rules	info.rules	rservices.rules	web-misc.rules
community-imap.rules	community-web-lis.rules	local.rules	scan.rules	web-php.rules
community-inappropriate.rules	community-web-misc.rules	misc.rules	shellcode.rules	x11.rules
community-mail-client.rules	community-web-php.rules	multimedia.rules	smtp.rules	
community-misc.rules	ddos.rules	mysql.rules	snmp.rules	

โดยค่าตั้งต้น Snort มี Rules เตรียมไว้ให้ใช้งานอยู่แล้วจึงไม่ต้องสร้างอะไรเพิ่มเอง

ภายใน Rules จะมีโครงสร้างดังนี้

```
[alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible SYN flood detected"; threshold: type threshold, track by_src, count )  
Alert(การแจ้ง) tcp(protocol) any(ip_src ต้นทาง) any(port ต้นทาง) -> $HOME_NET(local network) 80(port  
80) (flags: S;คือ SYN msg:"Possible SYN flood" ข้อความที่จะแสดง; )
```

เนื่องไปประมาณนี้ใช้ในการตรวจจับ การ flood ข้อมูลด้วย hping3 ได้

### ติดตั้ง Node export (ที่ Webserver)

15. wget https://github.com/prometheus/node\_exporter/releases/download/v1.6.1/node\_exporter-1.6.1.linux-amd64.tar.gz
16. tar xvfz node\_exporter-1.6.1.linux-amd64.tar.gz
17. sudo mv node\_exporter-1.6.1.linux-amd64/node\_exporter /usr/local/bin/
18. sudo nano /etc/systemd/system/node\_exporter.service (สร้าง system service เพื่อให้สามารถ enable)

```
[Unit]  
Description=Node Exporter  
Wants=network-online.target  
After=network-online.target  
  
[Service]  
User=node_exporter  
ExecStart=/usr/local/bin/node_exporter  
  
[Install]  
WantedBy=default.target
```

19. sudo useradd -rs /bin/false node\_exporter
20. sudo nano /etc/systemd/system/node\_exporter.service

```
[Service]  
User=node_exporter  
ExecStart=/usr/local/bin/node_exporter
```

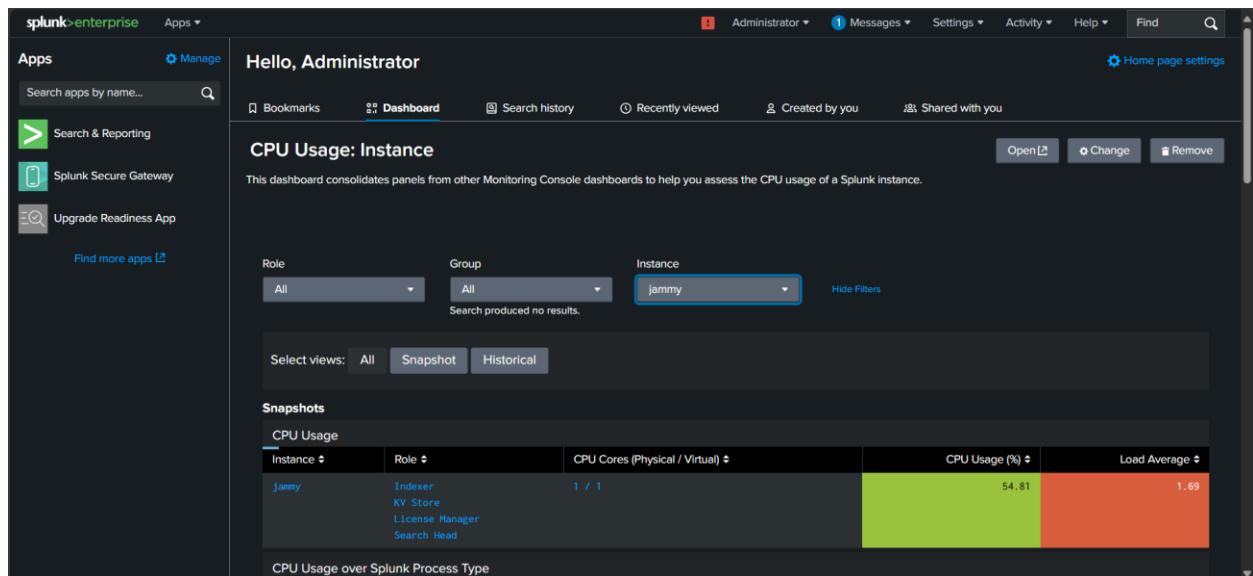
21. sudo systemctl daemon-reload

22. sudo systemctl start node\_exporter
23. sudo systemctl enable node\_exporter

## ภาษาไทย GNS3

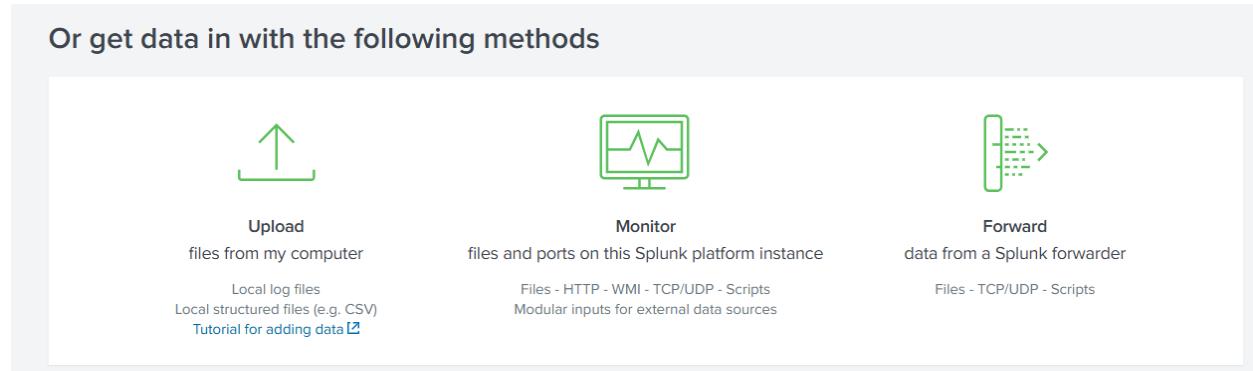
ติดตั้ง SIEM (Splunk) ip address 192.168.137.150 (localhost)

- 1.wget -O splunk-9.3.1-0b8d769cb912-linux-2.6-amd64.deb  
["https://download.splunk.com/products/splunk/releases/9.3.1/linux/splunk-9.3.1-0b8d769cb912-linux-2.6-amd64.deb"](https://download.splunk.com/products/splunk/releases/9.3.1/linux/splunk-9.3.1-0b8d769cb912-linux-2.6-amd64.deb)
- 2.dpkg -i splunk-9.3.1-0b8d769cb912-linux-2.6-amd64.deb (ติดไฟล์ deb)
- 3.sudo /opt/splunk/bin//splunk enable boot-start

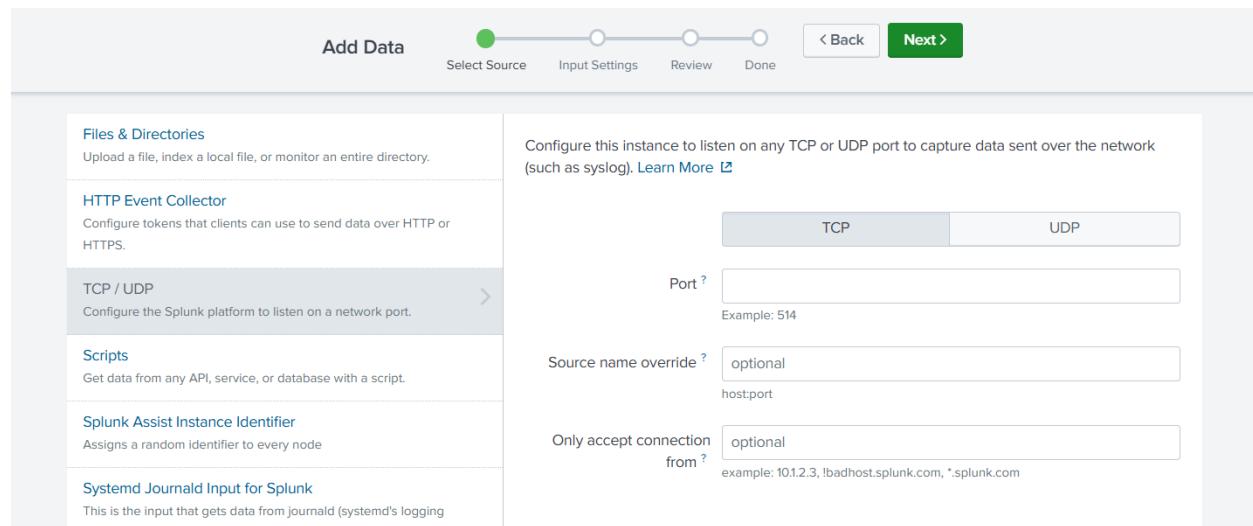


## การตั้งค่า SIEM เพื่อ Alert ไปยัง Line

1. ขั้นแรก เริ่มจาก เก็บ ข้อมูลจาก Source ก็คือ Snort



2. โดยเราจะ Monitor ผ่าน TCP Port 514



3. ตั้งค่าในไฟล์ snort.conf เพิ่มคำสั่งนี้เข้าไป

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

จากนั้นเข้าไปที่ rsyslog.conf และเพิ่มคำสั่งในการส่งข้อมูลผ่าน port 514 แบบ TCP

```
*.* @<Splunk_Server_IP>:514
```

4. กำหนดค่าพื้นฐานที่จะเอาข้อมูลจาก Snort มาเก็บบน Splunk

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Select New Select Source Type ▾

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More ↗](#)

App Context Search & Reporting (search) ▾

Source เลือกเป็น Syslog เพราะส่งข้อมูลมาจาก Syslog

App context ปรับเป็น Search & Reporting สำหรับการค้นหาข้อมูลอย่างง่ายดาย

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More ↗](#)

Index

Default ▾

Create a new index

ส่วน index ให้เราสร้าง new index ชื่อ IDS หรืออะไรก็ได้เพื่อจ่ายต่อการค้นหาข้อมูลของเรา

5. หลังจากเราได้รับข้อมูลพร้อมแล้วจาก Snort เราจะทำการตั้งเงื่อนไขสำหรับการ Alert

i	Time	Event
>	10/13/24 7:17:38.000 AM	<153>Oct 13 07:17:38 ubuntu snort[872]: [1:1418:11] SNMP request tcp [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.69:43746 -> 20.8.1.5:161 host = ubuntu   source = tcp:514   sourcetype = syslog
>	10/13/24 7:17:38.000 AM	<153>Oct 13 07:17:38 ubuntu snort[872]: [1:1418:11] SNMP request tcp [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.69:43746 -> 20.8.1.5:161 host = ubuntu   source = tcp:514   sourcetype = syslog
>	10/13/24 7:17:36.000 AM	<153>Oct 13 07:17:36 ubuntu snort[872]: [1:1421:11] Nmap Scanning detected [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.6 9:36354 -> 20.8.1.5:705 host = ubuntu   source = tcp:514   sourcetype = syslog

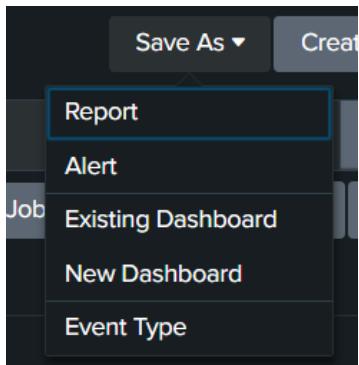
หน้าตาของข้อมูลที่เข้ามามาจะเป็นแบบนี้

หลังจากนี้เราจะใช้ SPL ซึ่งเป็นภาษาของ Splunk ใช้ในกรองข้อมูลเฉพาะส่วนที่เราต้องการ เช่นอย่างกรองเฉพาะการแจ้งเตือน Nmap มาใช้ในการ Alert

[index=ids Nmap Scanning] ผลลัพธ์ที่ได้จะเป็นดังนี้

i	Time	Event
>	10/13/24 7:17:36.000 AM	<153>Oct 13 07:17:36 ubuntu snort[872]: [1:1421:11] Nmap Scanning detected [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.6 9:36354 -> 20.8.1.5:705 host = ubuntu   source = tcp:514   sourcetype = syslog
>	10/13/24 7:17:36.000 AM	<153>Oct 13 07:17:36 ubuntu snort[872]: [1:1421:11] Nmap Scanning detected [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.6 9:36354 -> 20.8.1.5:705 host = ubuntu   source = tcp:514   sourcetype = syslog
>	10/13/24 7:12:06.000 AM	<153>Oct 13 07:12:06 ubuntu snort[872]: [1:1421:11] Nmap Scanning detected [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.6 9:34640 -> 20.8.1.5:705 host = ubuntu   source = tcp:514   sourcetype = syslog

เมื่อเรากรองเรียบร้อย ต่อไปจะเป็นการ ตั้งค่าการ Alert ออกไปแจ้งเตือน



ทำการ Save As ไปยัง Alert

Save As Alert

**Settings**

- Title: Title
- Description: Optional
- Permissions: Private (selected)
- Alert type: Scheduled
- Run every week ▾
- On: Monday ▾ at: 6:00 ▾
- Expires: 24 hour(s) ▾

**Trigger Conditions**

- Trigger alert when: Number of Results ▾
- is greater than ▾ 0

**Buttons:** Cancel, Save

โดยเราจะได้ข้อมูลลงไป

24. Title ชื่อการแจ้งเตือน
25. Des คำอธิบาย
26. Permissions ควรสามารถแก้ไขได้บ้าง

27. Alert Type ตั้งเวลา(Scheduled) หรือ แจ้งเตือนแบบเรียลไทม์(Real-time)

28. Trigger Conditions เงื่อนไขการ Alert

โดยส่วนสำคัญที่สุดของการ Alert คือ

The screenshot shows the 'Trigger Actions' configuration page. Under the 'When triggered' section, there is a single action listed: 'Run a script'. A warning message states: 'The run a script alert action is officially deprecated. Create a custom alert action to package a custom script instead.' Below the action, there is a 'File name' input field containing a placeholder path: 'Located in \$SPLUNK\_HOME/bin/scripts or \$SPLUNK\_HOME/etc/apps/search/bin/scripts'.

Trigger Actions จะให้แจ้งเตือนยังไง

โดยใน Splunk ไม่รองรับการส่ง Line Notification โดยตรงเราจึงจำเป็นต้องเขียน Script สำหรับ Alert

ด้วยตัวเอง โดยเราจะเขียน Script ไว้ใน Ubuntu Server ที่รัน Splunk ใน Directory /opt/splunk/bin/scripts

```
GNU nano 7.2                                         line.py
import requests

def send_line_alert(message):
    url = 'https://notify-api.line.me/api/notify'
    token = 'UJ6CpT0hiIdxh0FoYI7M4qeJAb0ayOdkRekGU9WWh03'
    headers = {
        'Authorization': f'Bearer {token}'
    }
    data = {
        'message': 'Someone is Scanning Your Network!'
    }
    requests.post(url, headers=headers, data=data)

send_line_alert("Splunk alert: Someone Scanning your network!")
```

โดยนี้คือตัวอย่าง Script ที่เราเขียนสำหรับการ Alert Nmap ไปยัง Line

จากนั้นนำชื่อไฟล์ที่ตั้ง line.py ไปใส่ไว้ใน Trigger Actions เพ่านี้ เมื่อมีการตรวจพบ Nmap

Splunk จะส่งข้อความ Alert ไปยัง Line ของเราทันที

นอกจากนี้ยังมี script ที่สามารถ Alert DDOS, SSH

```

GNU nano 7.2                                     flood.py | GNU nano 7.2                         auth_alert.py
import requests
def send_line_alert(message):
    url = 'https://notify-api.line.me/api/notify'
    token = 'UJ6CpT8hIdxh0FoYI7M4qeJAb0ay0dkRekGU9Wh03'
    headers = {
        'Authorization': f'Bearer {token}'
    }
    data = {
        'message': 'SYN Flood detected at port 80'
    }
    requests.post(url, headers=headers, data=data)
send_line_alert("Splunk alert: Your alert message here!")

def send_line_alert(message):
    url = 'https://notify-api.line.me/api/notify'
    token = 'UJ6CpT8hIdxh0FoYI7M4qeJAb0ay0dkRekGU9Wh03'
    headers = {
        'Authorization': f'Bearer {token}'
    }
    data = {
        'message': 'Someone ssh connected to splunk server with root account Please Check If it Not Safe'
    }
    requests.post(url, headers=headers, data=data)
send_line_alert("Splunk alert: Your alert message here!")

```

## การตั้งค่า SIEM เพื่อ Alert ไปยัง Discord

### หลักการเบื้องต้น

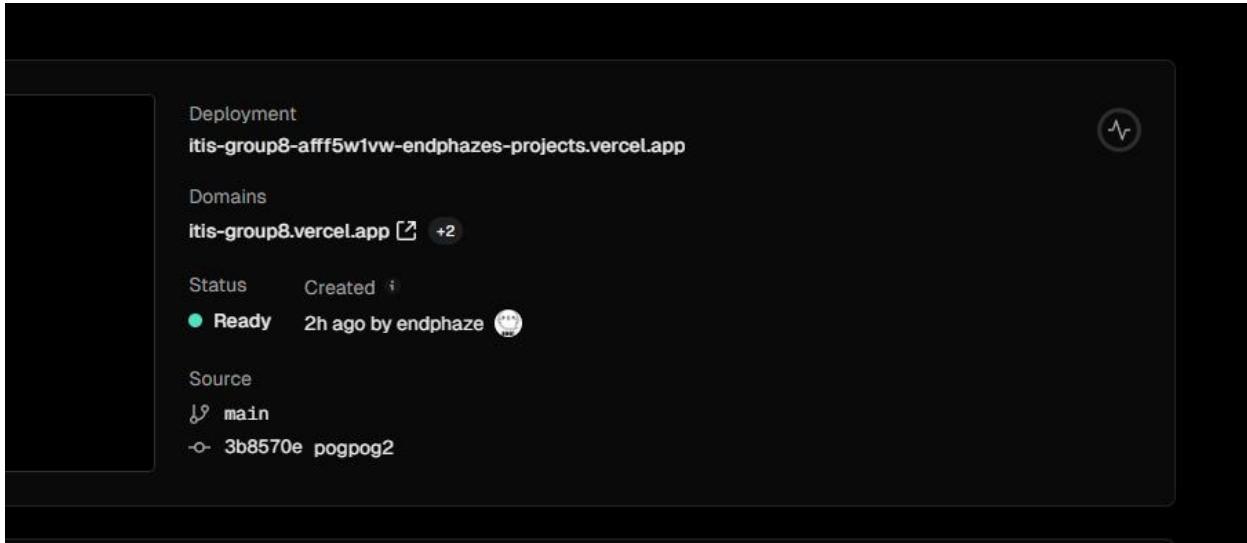
29. SIEM post data ไปยัง [itis-group8.vercel.app/siem\\_hook](https://itis-group8.vercel.app/siem_hook)

30. [itis-group8.vercel.app/siem\\_hook](https://itis-group8.vercel.app/siem_hook) post data discord webhook url

สาเหตุที่ไม่สามารถใส่ discord webhook ที่ siem ตรงๆได้เนื่องจาก discord webhook api

นั้นมีโครงสร้างข้อมูลที่ถูกกำหนดมาแล้วจากทาง discord เอง ทำให้ siem ที่ไม่สามารถปรับแต่ง body ได้ ต้อง post ไป vercel ก่อน จากนั้น เสียนโค้ดให้ vercel post ไปยัง discord webhook api อีกที โดยปรับแต่ง body ให้ตรงกับ discord webhook

ตัว Vercel นำความรู้จากวิชา IoT มาประยุกต์ทำ api server



ตัวโกดกที่รันใน vercel จะเห็นว่าเมื่อมีการ route /webhook ก็จะนำข้อมูลจาก siem มาใส่ใน 'content' ซึ่งเป็นโครงสร้างที่ discord webhook api กำหนดไว้ จากนั้น post ไปยัง discord webhook

```
 6  import os
 7
 8  app = FastAPI()
 9
10
11  load_dotenv()
12
13
14  @app.post('/webhook')
15  def hello_world(data: dict):
16      url = os.environ.get("discord_webhook_url")
17
18      json_obj = json.dumps(data, indent=4)
19      payload = {
20          'content' : f'''json\n{json_obj}'''
21      }
22
23      requests.api.post(url, payload)
```

ใน siem ให้ตั้งค่าเมื่อ alert ให้ส่ง post ไปที่ vercel

+ Add Actions ▾

Webhook

URL <https://itis-group8.vercel.app/webhook>

Specified URL to send JSON payload via HTTP POST (ex., <https://your.server.com/api/v1/webhook>).

Learn More ↗

Remove

เมื่อ post ไป discord webhook แล้วสำเร็จ จะเกิดผลดังนี้ ข้อมูลนี้สามารถนำไปพัฒนาต่อได้

# network-bot-and-ids

}

SIEM notification APP Today at 7:00 PM

```
{  
    "sid": "rt_scheduler__admin__search__RMD58f4bbb2409743c2f_at_1728809666_11.218",  
    "search_name": "Scanning Alert",  
    "app": "search",  
    "owner": "admin",  
    "results_link": "http://Admin:8000/app/search/search?  
q=%7Cloadjob%20rt_scheduler__admin__search__RMD58f4bbb2409743c2f_at_1728809666_11.218%20%7  
&latest=now",  
    "result": {  
        "_confstr": "source::tcp:514|host::ubuntu|syslog",  
        "_eventtype_color": "",  
        "_indextime": "1728820784",  
        "_kv": "1",  
        "_raw": "<153>Oct 13 11:59:44 ubuntu snort[872]: [1:1418:11] SNMP request tcp [Cl  
[Priority: 2] {TCP} 20.8.1.69:41364 -> 20.8.1.5:161",  
        "_serial": "90",  
        "_si": [  
            "jammy",  
            "ids"  
        ]  
    }  
}
```

การแจ้งเตือนด้วยเสียง ใน Discord

นำข้อมูลก่อนหน้านี้มาพัฒนาต่อยอด ให้มีความสามารถมากขึ้น ด้วยการแจ้งเตือนเสียง

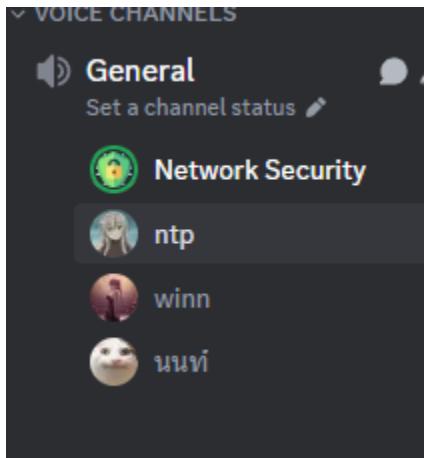
เริ่มแรก สร้าง Bot Discord ให้มีการตรวจจับว่ามีการส่งข้อความอะไรมา  
จากนั้นแจ้งเตือนทั้งผ่านข้อความและเล่นเสียง ไปที่ Discord

```
if '"search_name": "Scanning Alert"' in message.content:  
    print("nmap detect! playing nmap_alert sound!")  
    voice_client = await client.get_channel(1292666201654300676).connect()  
    source = FFmpegPCMAudio("nmap.mp3")  
    voice_client.play(source, after=lambda e: client.loop.create_task(leave(voice_client)))  
  
    embed = nextcord.Embed(title=f"🔴 {alert.get('result').get('_raw')}", color=nextcord.Color.red)  
    await client.get_channel(1294973927092916285).send(embed=embed)  
  
if '"search_name": "SYN FLOOD"' in message.content:  
    print("dos detect! playing dos_alert sound!")  
    voice_client = await client.get_channel(1292666201654300676).connect()  
    source = FFmpegPCMAudio("dos.mp3")  
    voice_client.play(source, after=lambda e: client.loop.create_task(leave(voice_client)))  
  
    embed = nextcord.Embed(title=f"⚡ {alert.get('result').get('_raw')}", color=nextcord.Color.red)  
    await client.get_channel(1294973927092916285).send(embed=embed)  
  
if '"search_name": "SSH"' in message.content:  
    print("dos detect! playing dos_alert sound!")  
    voice_client = await client.get_channel(1292666201654300676).connect()  
    source = FFmpegPCMAudio("ssh.mp3")  
    voice_client.play(source, after=lambda e: client.loop.create_task(leave(voice_client)))  
  
    embed = nextcord.Embed(title=f"⚠ {alert.get('result').get('_raw')}", color=nextcord.Color.red)  
    await client.get_channel(1294973927092916285).send(embed=embed)
```

หลักการทำงาน โค้ดนี้คือ ดักตรวจข้อความที่เป็นข้อมูลจาก siem และแยกว่าการแจ้งเตือนนั้นเป็นประเภทอะไร  
จากนั้นก็ทำการเล่นเสียงผ่านและแจ้งเตือนข้อความที่เข้าใจง่ายผ่าน Discord

จำเป็นต้องรันโค้ดนี้ในเครื่องตลอด ให้ติดต่อกับ Server Discord อุปกรณ์เดียวกัน เพื่อให้ Bot สามารถทำงานได้

การแจ้งเตือนด้วยเสียงจะมี Bot มาเล่นเสียงในห้องพูดคุย



การแจ้งเตือนด้วยข้อความให้สามารถอ่านง่ายมากขึ้น

## Welcome to # 🔊 network-ids-notification!

This is the start of the # 🔊 network-ids-notification channel.

[Edit Channel](#)

October 13, 2024

Network Security APP Today at 6:22 PM

🔒 2024-10-13T11:22:15.799837+00:00 Admin systemd-logind[804]: New session 44 of user jammy.

⌚ <153>Oct 13 11:23:01 ubuntu snort[872]: [1:200001:1] Possible SYN flood detected {TCP} 20.8.1.69:1438 -> 20.8.1.5:80

Network Security APP Today at 7:00 PM

⌚ <153>Oct 13 11:59:44 ubuntu snort[872]: [1:1418:11] SNMP request tcp [Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.69:41364 -> 20.8.1.5:161

หากสนใจอยากเห็นการใช้งานจริง สามารถเข้า server discord โดยลิ้งค์ด้านล่างได้ และลิ้งค์ github ที่ใช้งานกับ vercel

<https://discord.gg/R74hPfbGwy>

[https://github.com/endphaze/ITIS\\_GROUP8](https://github.com/endphaze/ITIS_GROUP8)

## บทที่ 5

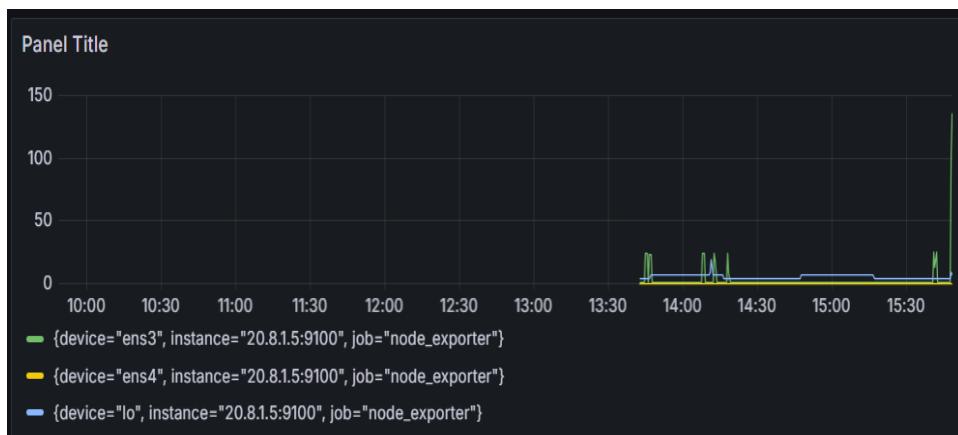
### ผลการทดสอบและปัญหาที่เกิดขึ้น

ผลการทดสอบ Alert ไปยัง Line

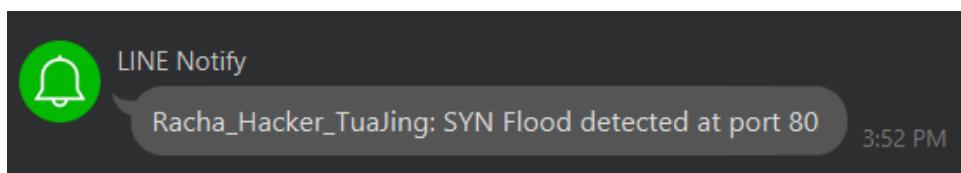
ทดสอบ hping ไปยัง Webserver

```
[sudo] password for non:  
non@ubuntu:~$ sudo hping3 -S --flood 20.8.1.5
```

ผลการรับ Packet ของ Webserver จาก Grafana



ผลการแจ้งเตือนจาก Line Notify



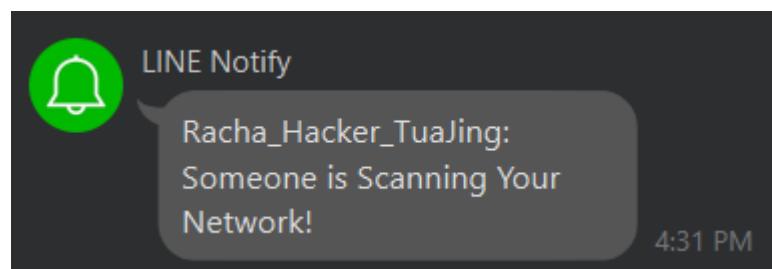
ผลการวิเคราะห์ Ip address ต่อ Packet ที่ส่งเข้ามา จาก SIEM



ทดสอบ nmap ไปยัง 20.8.1.0/24

```
non@ubuntu:~$ nmap 20.8.1.0/24
```

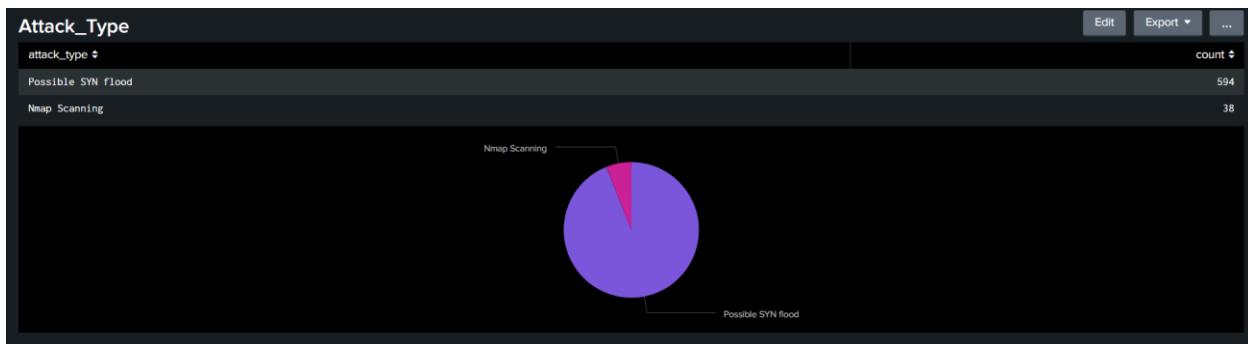
ผลการแจ้งเตือนไปยัง Line



ผลการวิเคราะห์ ip address ต่อ packet ที่ส่งเข้ามายัง SIEM



ผลการจำแนกประเภทการโจมตีด้วย SIEM ต่อ Packet ที่รับเข้ามา



ผลการทดลอง Alert ไปยัง Discord

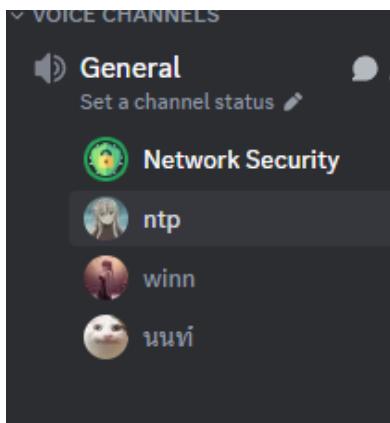
Alert with text

SIEM notification wall วันนี้ เวลา 18:20



```
{  
    "sid": "rt_scheduler__admin__search__RMD58f4bbb2409743c2f_at_1728809666_11.176",  
    "search_name": "Scanning Alert",  
    "app": "search",  
    "owner": "admin",  
    "results_link": "http://Admin:8000/app/search/search?  
q=%7Cloadjob%20rt_scheduler__admin__search__RMD58f4bbb2409743c2f_at_1728809666_11.176%20%  
7C%20head%201%20%7C%20tail%201&earliest=0&latest=now",  
    "result": {  
        "_confstr": "source::tcp:514|host::ubuntu|syslog",  
        "_eventtype_color": "",  
        "_indextime": "1728818423",  
        "_kv": "1",  
        "_raw": "<153>Oct 13 11:20:22 ubuntu snort[872]: [1:1418:11] SNMP request tcp  
[Classification: Attempted Information Leak] [Priority: 2] {TCP} 20.8.1.69:56210 ->  
20.8.1.5:61",  
        "_serial": "78",  
        "_si": [  
            "jammy",  
            "ids"  
        ],  
        "_sourcetype": "syslog",  
        "_time": "1728818422",  
        "date_hour": "11",  
        "date_mday": "13",  
        "date_minute": "20",  
        "date_month": "october",  
        "date_second": "22",  
        "date_wday": "sunday",  
        "date_year": "2024",  
        "date_zone": "local",  
        "eventtype": "",  
        "host": "ubuntu",  
    },  
}
```

Alert with Sound



รับชมเพิ่มเติมใน demo

## ปัญหาที่เกิดขึ้น

Plugin สำหรับ alert จาก Splunk ของ Grafana ไม่สามารถใช้งานได้ (คาดว่าเลิกให้บริการแล้ว) แก้ปัญหาโดยให้ Grafana แสดง dashboard ของทรัพยากร Webserver และ ให้ Splunk ทำหน้าที่ Alert แทนผ่านการ run script api ของ Line และ Discord

## บรรณานุกรม

### แหล่งอ้างอิง

[Get started with Grafana and Prometheus | Grafana documentation](#)

[Monitoring Linux host metrics with the Node Exporter | Prometheus](#)

[How to Start, Stop, and Restart Nginx \(systemctl & Nginx Commands\) \(phoenixnap.com\)](#)

[Rules - Snort 3 Rule Writing Guide](#)

[Detection and Monitoring w/Splunk & Snort | by Dion Alexander | Medium](#)

[How to Create a Music Bot Using Discord.js – Step-by-Step Tutorial \(freecodecamp.org\)](#)

<https://docs.nextcord.dev/en/stable/>

