ระบบสแกนช่องโหว่ด้วยราสเบอร์รี่ไพน์
VULNERABILITY SCANNING SYSTEM USING RASPBERRY PI

โดย

นายนนทวัฒน์ ทับทิม

นายสุปกรณ์ สวัสดิ์พุทรา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

ระบบสแกนช่องโหว่ด้วยราสเบอร์รี่ไพน์
VULNERABILITY SCANNING SYSTEM USING RASPBERRY PI

โดย

นายนนทวัฒน์ ทับทิม **60010495**

นายสุปกรณ์ สวัสดิ์พุทรา **60011097**

อาจารย์ที่ปรึกษา

ผศ.ดร.ธเนศ พัฒนธาดาพงษ์

ผศ.ดร.นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

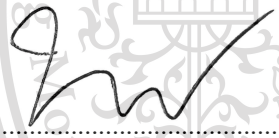
ปีการศึกษา 2563

ปริญญานิพนธ์ปีการศึกษา 2563
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสแกนช่องโหว่ด้วยราสเบอร์รี่ไพน์
VULNERABILITY SCANNING SYSTEM USING RASPBERRY PI

ผู้จัดทำ
1. นายนนทวัฒน์ ทับทิม 60010495
2. นายสุปกรณ์ สวัสดิ์พุทรา 60011097

..... **อาจารย์ที่ปรึกษา**
(ผศ.ดร.ธเนศ พัฒนธาดาพงษ์)



..... **อาจารย์ที่ปรึกษาร่วม**
(ผศ.ดร.นภัทร สระเอี่ยม)

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง “ระบบสแกนช่องโหว่” สำเร็จลุล่วงได้ด้วยความกรุณา และความช่วยเหลือจาก ผศ.ดร.ธเนศ พัฒนธาดาพงษ์ และ ผศ.ดร.นภัทร สระเอี่ยม อาจารย์ที่ปรึกษาปริญญานิพนธ์ ดร.สมปอง วิเศษพาณิชกิจ ที่ให้คำแนะนำ คำปรึกษา และแนวคิดตลอดจนการแก้ไขปัญหาข้อบกพร่องต่างๆ มาโดยตลอดเพื่อให้ปริญญานิพนธ์นี้สำเร็จโดยสมบูรณ์รวมถึงการสนับสนุนเครื่องมือ อุปกรณ์ หนังสือ และบทความต่างๆ ที่ใช้ระหว่างการทำปริญญานิพนธ์ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงสำหรับการดูแลและเอาใจใส่

ขอขอบพระคุณคณาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้อบรมสั่งสอนมอบวิชาความรู้และประสบการณ์ต่างๆ ที่สามารถนำมาใช้ในการทำปริญญานิพนธ์นี้

ขอบคุณเพื่อนๆ ทุกคนที่คอยให้คำแนะนำ คำปรึกษาและความช่วยเหลือเป็นอย่างดีมาโดยตลอดซึ่งเป็นประโยชน์ในการทำปริญญานิพนธ์นี้ตลอดจนขอกราบพระคุณบิดา มารดาและครอบครัวที่คอยเป็นกำลังใจที่ดีเสมอมา

นายนนทวัฒน์ ทับทิม
นายสุปกรณ์ สวัสดิ์พุทรา
ผู้จัดทำ

ระบบตรวจจับช่องโหว่ด้วยราสเบอร์รี่ไพน์
VULNERABILITY SCANNING SYSTEM USING RASPBERRY PI

โดย นายนนทวัฒน์ ทับทิม 60010495
นายสุปกรณ์ สวัสดิ์พุทรา 60011097

อาจารย์ที่ปรึกษา ผศ.ดร.ธเนศ พัฒนธาดาพงษ์
อาจารย์ที่ปรึกษาร่วม ผศ.ดร.นภัทร สระเอี่ยม

บทคัดย่อ

ปริญญานิพนธ์นี้ต้องการศึกษาช่องโหว่ภายในระบบเครือข่ายโดยทำการวิเคราะห์แล้วทำการรวบรวมช่องโหว่ที่พบในเครือข่ายมาแสดงผลสรุปเป็นรายงานโดยใช้เครื่องมือบนระบบปฏิบัติการ Kali Linux โดยเครื่องมือที่ใช้ในการค้นหาและรวบรวมช่องโหว่ก็จะแตกต่างกันออกไปขึ้นอยู่กับประเภทของช่องโหว่ ขั้นตอนดำเนินงานได้แก่การค้นหาและรวบรวมช่องโหว่ที่พบในเครือข่ายโดยใช้เครื่องมือจากระบบปฏิบัติการ Kali Linux จากนั้นใช้ภาษาไพธอนในการค้นหาช่องโหว่อัตโนมัติและแสดงผลผ่านทางเว็บแอปพลิเคชัน (Web Application)

ABSTRACT

This thesis would like to study about vulnerabilities in our network by analysis and gathering breach information. Bring those result to display in report form with Kali Linux' tools. Tools that we use to find and gathering vulnerability will be different depend on the type of the breach. Our procedure is to searching and gathering the breach that we found in our network with tools from Kali Linux. After that, write a Python codes to search vulnerabilities automatically and report the result on web application.

สารบัญ

สารบัญ (ต่อ)

สารบัญรูป

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

สารบัญรูป(ต่อ)

สารบัญรูป(ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อย่างที่ทราบกันดีว่าในปัจจุบันความปลอดภัยในระบบเครือข่ายคอมพิวเตอร์นั้นเป็นสิ่งที่สำคัญอย่างยิ่งโดยเฉพาะภายในองค์กร ที่ใช้งานระบบเครือข่ายนั้นมีผู้ใช้งานจำนวนมาก ดังนั้นจึงมีทั้งผู้ที่ประสงค์ดีและผู้ประสงค์ร้าย และผู้ที่ประสงค์ร้ายอาจอยู่ทั้งภายในและภายนอกระบบเครือข่ายขององค์กร ซึ่งอาจก่อให้เกิดความเสี่ยงจากการบุกรุกระบบเครือข่ายและความเสี่ยงจากภัยคุกคามต่างๆ ที่ผู้ประสงค์ร้ายพยายามที่จะบุกรุกระบบเครือข่ายโดยอาจเข้าสู่ระบบโดยมิได้รับอนุญาตเพื่อลักลอบนำข้อมูลที่สำคัญภายในองค์กรเพื่อที่จะนำข้อมูลที่ได้มาไปแสวงหาผลประโยชน์ส่วนตน ซึ่งสิ่งหนึ่งที่จะป้องกันการโจมตีจากผู้ที่ประสงค์ร้ายคือการตรวจสอบช่องโหว่ภายในระบบเครือข่าย ไม่ว่าจะเป็นจากตัวระบบปฏิบัติการ ซอฟต์แวร์ของอุปกรณ์เครือข่ายที่ไม่ได้รับการอัปเดตและอื่นๆ ดังนั้นผู้จัดทำจึงได้ทำการศึกษาการใช้งานโปรแกรมตรวจจับช่องโหว่บนระบบปฏิบัติการลินุกซ์และศึกษาช่องโหว่ด้านความปลอดภัยของระบบเครือข่ายคอมพิวเตอร์และทำการเขียนโปรแกรมในการแสดงผลช่องโหว่ที่ตรวจพบเพื่อให้ผู้ใช้งานสามารถรับรู้ถึงช่องโหว่ที่พบและหาทางป้องกันช่องโหว่ได้

1.2 วัตถุประสงค์

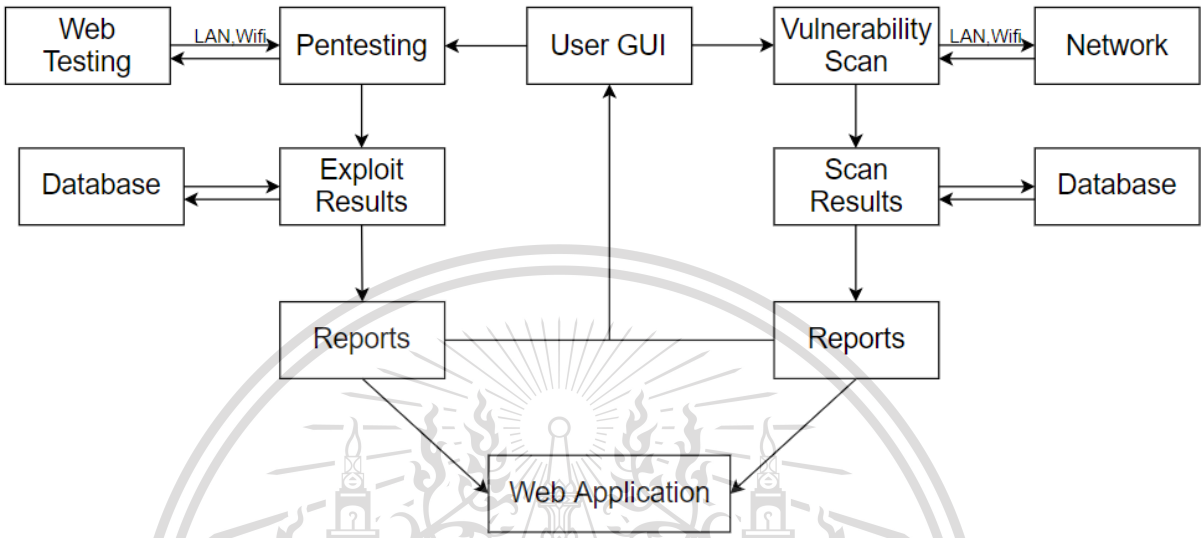
1) เพื่อศึกษาและทำความเข้าใจเกี่ยวกับการค้นหาช่องโหว่ภายในเครือข่าย
2) สามารถออกแบบและพัฒนาระบบค้นหาช่องโหว่ภายในเครือข่าย
3) สามารถนำระบบค้นหาช่องโหว่ไปใช้ในองค์กร

1.3 ขอบเขตของปริญญานิพนธ์

ระบบตรวจจับช่องโหว่มีหลักการทำงานดังรูปที่ 1.1 บล็อกไดอะแกรมการทำงานของระบบ ซึ่งมีขอบเขตของโครงงานดังนี้

1) ศึกษาเครื่องมือต่างๆของ Kali Linux ที่ใช้ในการสแกนช่องโหว่ภายในเครือข่าย
2) ออกแบบ ระบบสแกนช่องโหว่
3) พัฒนาระบบสแกนช่องโหว่เพื่อนำเอาข้อมูลที่ได้มาสรุปและรายงานผลที่เกิดขึ้น

บล็อกไดอะแกรมของโครงงานที่นำเสนอ

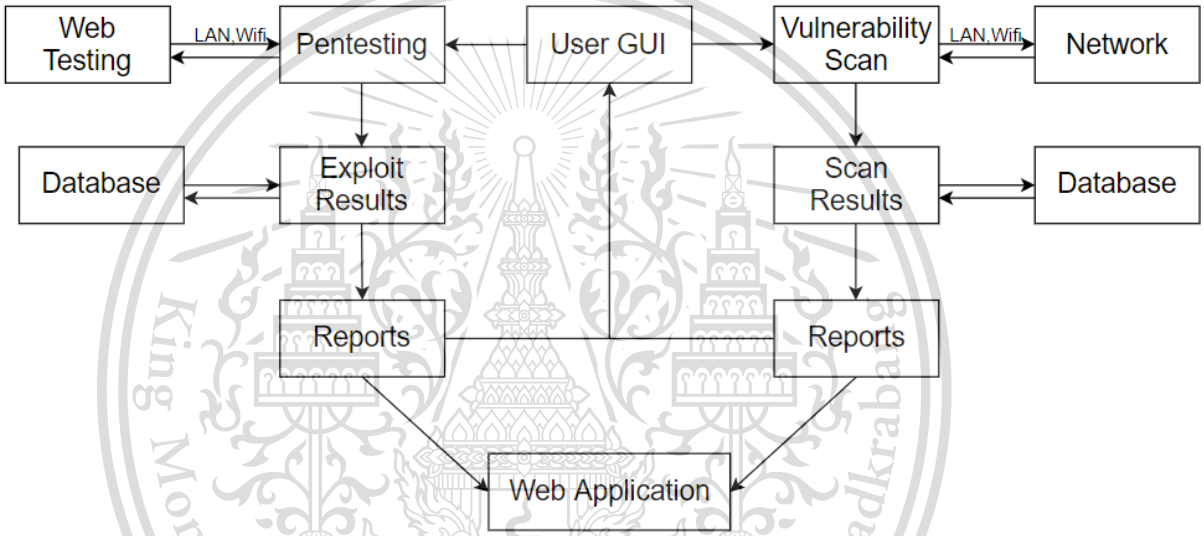


รูปที่ 1.1 บล็อกไดอะแกรมการทำงานโดยรวมของระบบสแกนช่องโหว่ด้วยราสเบอร์รี่ไพน์

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

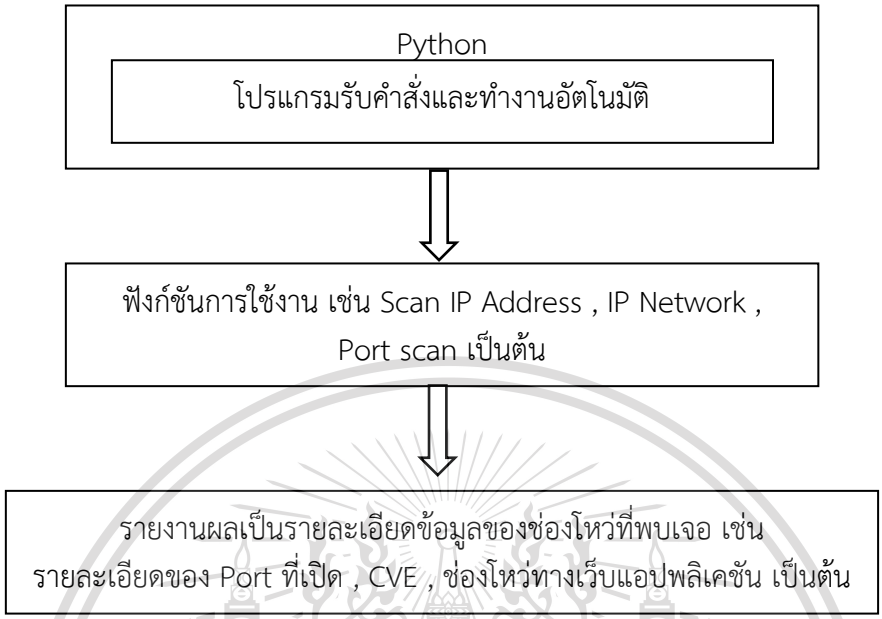
โครงงานเรื่องระบบตรวจจับช่องโหว่มีการทำงานโดยขั้นแรกจะรับคำสั่งจากผู้ใช้งานผ่าน User GUI และควบคุมการใช้เครื่องมือบนระบบปฏิบัติการ Raspbian เพื่อรวบรวมข้อมูลนำไปสร้างส่วนที่เป็นรายงานผลการตรวจจับช่องโหว่ ดังรูปที่ 2.1



รูปที่ 2.1 รายละเอียดการทำงานโดยรวมของระบบ

ซึ่งในการเขียนโปรแกรมระบบตรวจจับช่องโหว่จะมีการทำงานในแต่ละส่วนดังรูปที่ 2.2 มีรายละเอียดดังนี้

1. การตรวจจับช่องโหว่ใช้เครื่องมือจากระบบปฏิบัติการ Raspbian โดยใช้ภาษาไพธอน (Python) เพื่อรับคำสั่งและมาทำการสั่งงานเครื่องมือต่าง ๆ ที่สามารถใช้ในเทอร์มินอล (Terminal) ที่ได้ทำการติดตั้งอยู่ในระบบปฏิบัติการ Raspbian ในการตรวจจับช่องโหว่
2. เว็บแอปพลิเคชัน (Web Application) เป็นหน้าเว็บสำหรับแสดงผลให้ผู้ใช้งานได้ดูผลการรายงานของช่องโหว่ที่ตรวจพบโดยเนื้อหาที่แสดงหน้าเว็บใช้ภาษาเอชทีเอ็มแอล (Hyper Text Markup Language) เป็นหลัก



รูปที่ 2.2 บล็อกไดอะแกรมการเขียนโปรแกรมเพื่อตรวจจับช่องโหว่ในเครือข่าย

2.1 ระบบเครือข่ายท้องถิ่น

ระบบเครือข่ายท้องถิ่น (Local Area Network : LAN) คือระบบเครือข่าย แบบเชื่อมต่อคอมพิวเตอร์เข้าด้วยกันในระยะจำกัด เช่น ในอาคารเดียวกันหรือบริเวณเดียวกันที่สามารถลากสายถึงกันได้โดยตรง ส่วนมากจะใช้สายเคเบิ้ลหรือที่เรียกกันว่า สายแลน เป็นตัวกลางในการเชื่อมต่อ ความเร็วของเครือข่าย LAN ขึ้นอยู่กับตัวกลางสายส่งที่ใช้เทคนิคการส่งสัญญาณและข้อกำหนดของผู้ให้บริการ

2.1.1 อุปกรณ์ในเครือข่าย LAN

2.1.1.1 เร้าเตอร์ (Router)

เร้าเตอร์ เป็นอุปกรณ์คอมพิวเตอร์ที่ทำหน้าที่ค้นหาเส้นทาง และส่งแพ็กเก็ตข้อมูลระหว่างเครือข่ายคอมพิวเตอร์ไปยังเครือข่ายปลายทางที่ต้องการเร้าเตอร์ทำงานบนชั้นที่ 3 ตามมาตรฐานของ OSI Model แสดงดังรูปที่ 2.3

รูปที่ 2.3 เร้าเตอร์ (Router)

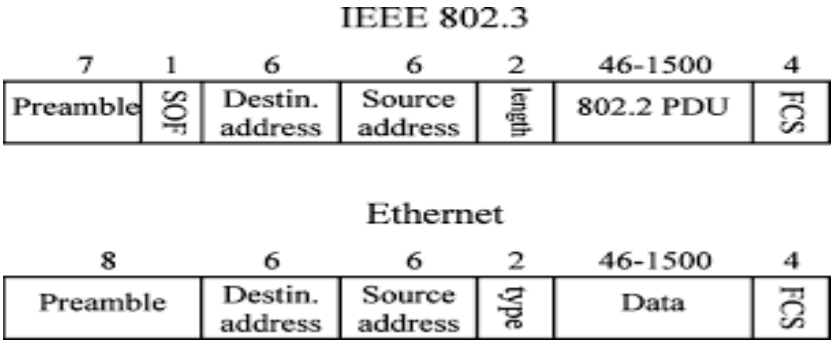
2.1.1.2 สวิตช์ (Switch)

สวิตช์เป็นอุปกรณ์เครือข่ายที่ทำหน้าที่เชื่อมต่ออุปกรณ์ต่าง ๆ เข้าด้วยกันโดยใช้หลักการสลับแพ็กเก็ตเพื่อรับประมวลผลและส่งต่อข้อมูลไปยังอุปกรณ์ปลายทาง แสดงดังรูปที่ 2.4

รูปที่ 2.4 สวิตช์ (Switch)

2.1.2 มาตรฐาน IEEE 802.3 และ Ethernet

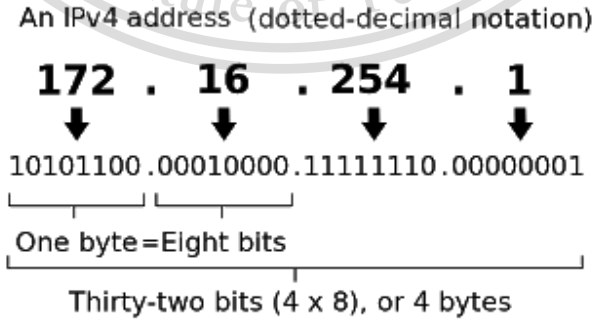
ระบบเครือข่าย Ethernet ถูกพัฒนาขึ้นในปลายทศวรรษ 1970 และในปี 1980 บริษัท Digital Equipment , Intel และ Xeror ได้ร่วมกันออกระบบ Ethernet I ซึ่งใช้งานกับสาย และต่อมาในปีก็ได้ทำการพัฒนาเป็น Ethernet II ซึ่งเป็นระบบเครือข่ายที่ถูกใช้งานมากที่สุดแบบหนึ่งจากนั้นองค์กรมาตรฐานจึงได้ออกข้อกำหนดมาตรฐาน IEEE 802.3 โดยใช้ Ethernet II เป็นรากฐานโดยมีจุดแตกต่างจากเล็กน้อย ดังรูปที่ 2.5 แต่หลักการใหญ่ๆ จะคล้ายคลึงกันคือ ใช้ Access Method และ CSMA/CD และใช้ Topology แบบ Bus หรือ Star นอกจากมาตรฐาน IEEE 802.3 ยังได้ร่างมาตรฐานการใช้สื่อในระดับกายภาพ (Physical) แบบต่าง ๆ ทำให้สามารถใช้สายเคเบิลในระดับกายภาพแบบได้หลายแบบโดยไม่ต้องเปลี่ยนในส่วนของชั้นที่ 2 ขึ้นไป เช่น 10Base5, 10BaseT โดย "10" หมายถึงความเร็ว 10 Mbps ส่วน "Base" หมายถึง Baseband และในส่วนสุดท้ายนั้นในช่วงแรก "5" หมายถึงระยะไกลสุดที่สามารถเชื่อมต่อมีหน่วยเป็นเมตรคูณร้อยในที่นี่คือ 500 เมตร แต่ต่อมาได้มีการใช้ความหมายของส่วนนี้เพิ่มเติมเป็นชนิดของสาย เช่น "T" หมายถึง ใช้สาย Twisted Pair และ "F" หมายถึง สายใยแก้วนำแสง



รูปที่ 2.5 เปรียบเทียบระหว่าง IEEE 802.3 และ Ethernet

2.1.3 หมายเลขไอพีแอดเดรส (IP Address) และ Subnet Mask

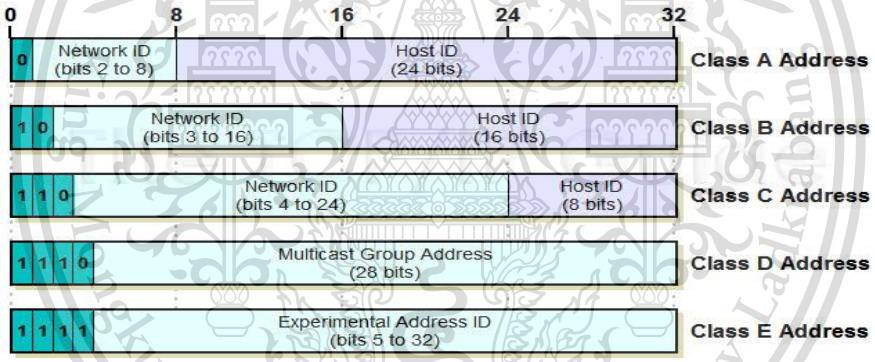
2.1.3.1 IP Address (Internet Protocol Address) คือ หมายเลข Logical ให้เครื่องคอมพิวเตอร์แต่ละเครื่องในระบบเครือข่ายที่ใช้โปรโตคอลแบบ TCP/IP ในระบบเครือข่ายจำเป็นจะต้องมีหมายเลข IP กำหนดไว้ให้กับคอมพิวเตอร์ และอุปกรณ์อื่น ๆ ที่ต้องการ IP เวลามีการโอนย้ายข้อมูลหรือสั่งงานใด ๆ จะสามารถทราบตำแหน่งของเครื่องที่ต้องการส่งข้อมูลไปจะได้ไม่ผิดพลาดเวลาส่งข้อมูลซึ่งประกอบด้วยตัวเลข 4 ชุดมีเครื่องหมายจุดขั้นระหว่างชุด เช่น 192.168.100.1 หรือ 172.16.10.1 เป็นต้นโดยหมายเลข IP Address ของเครื่องคอมพิวเตอร์แต่ละเครื่องจะมีค่าไม่ซ้ำกันตัวเลข 4 ชุดนี้บอก คือ Network ID กับ Host ID จะบอกให้รู้ว่า เครื่องคอมพิวเตอร์ของเราอยู่ในเครือข่ายไหน และเป็นเครื่องไหนในเครือข่ายโดย Network ID และ Host ID มีค่าเท่าไหร่ ก็ขึ้นอยู่กับว่า IP Address นั้น อยู่ใน Class อะไรโดยปัจจุบันใช้ IP Address เป็นเวอร์ชัน 4 หรือ IPv4 โดยแสดงดังรูปที่ 2.6



รูปที่ 2.6 รูปแบบของ IPv4

โดยการแบ่ง IP Address ตาม Class เป็นการแบ่งเพื่อให้เกิดความเป็นระเบียบเรียบร้อยและเพื่อจุดประสงค์ในการใช้งานที่ต่างกันโดยการกำหนดหมายเลขแต่ละบิตแสดงดังรูปที่ 2.7 และสามารถแบ่ง Class ได้ 5 Class ดังนี้

1. Class A เริ่มตั้งแต่ 1.0.0.1 ถึง 127.255.255.254 มี Address ใน Network ได้ทั้งหมด 16 ล้านเบอร์และมี Network ได้ทั้งหมด 128 เครือข่าย
2. Class B เริ่มตั้งแต่ 128.0.0.1 ถึง 191.255.255.254 มี Address ใน Network ได้ทั้งหมด 65,536 เบอร์และมี Network ได้ทั้งหมด 16,384 เครือข่าย
3. Class C เริ่มตั้งแต่ 192.0.0.1 ถึง 223.255.255.254 มี Address ใน Network ได้ทั้งหมด 254 เบอร์
4. Class D เริ่มตั้งแต่ 224.0.0.0 ถึง 225.255.255.255 มี Address ใช้สำหรับการทำ Multicast
5. Class E เริ่มตั้งแต่ 240.0.0.0 ถึง 254.255.255.255 ยังไม่ถูกนำมาใช้งาน



รูปที่ 2.7 การกำหนดหมายเลขบิตในแต่ละ CLASS

2.1.3.2 Subnet Mask คือ เป็นค่า Parameter ซึ่งใช้ระบุควบคู่กับ IP Address โดย Subnet Mask มีหน้าที่แบ่งแยกส่วนของ IP Address ว่าส่วนไหนเป็น Network ID และ ส่วนไหนเป็น Host ID โดยจะสามารถสังเกตได้เพราะทุกครั้งเมื่อเรากำหนดหมายเลข IP Address ให้กับเครื่องคอมพิวเตอร์ในเครือข่ายต้องกำหนด Subnet Mask ลงไปด้วยทุกครั้ง IP Address จะเพิ่มขึ้นหรือลดลงอยู่ที่ Subnet Mask ที่เรากำหนดค่าไว้ให้โดยในแต่ละ Class จะมี Subnet Mask ดังนี้

1. Class A จะมี Subnet Mask เป็น 255.0.0.0 หรือ /8
2. Class B จะมี Subnet Mask เป็น 255.255.0.0 หรือ /16
3. Class C จะมี Subnet Mask เป็น 255.255.255.0 หรือ /24

2.1.3.3 Public IP คือ หมายเลข IP ที่เราสามารถเข้าถึงได้บนเครือข่าย Internet เพราะฉะนั้นการที่เราจะใช้ Internet ในแต่ละครั้งเราจะต้องมี Public IP เพื่อที่จะติดต่อสื่อสารกับ Public IP อื่นๆ

2.1.3.4 Private IP คือ หมายเลข IP ส่วนตัวมีไว้สำหรับใช้งานภายในองค์กรเท่านั้น ไม่ว่าองค์กรนั้นจะมีขนาดใหญ่หรือเล็กเพียงใดก็ตาม จะถูกกำหนดหมายเลข IP โดยผู้ดูแลระบบและเมื่อต้องการติดต่อกับ Public IP ต้องใช้การแปลง IP หรือ NAT

2.1.4 หมายเลขพอร์ต (Port Number)

Port number คือเลขฐาน 16 บิต ตั้งแต่ 0 – 65535 หมายเลข port แต่ละหมายเลขจะถูกกำหนดโดยเฉพาะจาก OS (Operating Systems) โดยทาง Internet Assigned Numbers Authority (IANA) เป็นหน่วยงานกลางในการประสานการใช้ Port ว่า Port หมายเลขใดควรเหมาะสำหรับ Service ใด โดย Port Number สามารถแยกเป็น 2 ประเภทดังนี้

2.1.4.1 Well Known Ports เป็นพอร์ตที่มีหมายเลขตั้งแต่ 0 ถึง 1,024 เป็นหมายเลขพอร์ตที่ให้บริการทั่วไปที่รู้จักกันดี ดังตารางที่ 2.1

ตารางที่ 2.1 Well KNOW PORTS

Service	Port	Function
HTTP	80	Web traffic
HTTPS	443	Secure web traffic
FTP	20, 21	File transfer
DNS	53	Name resolution
SMTP	25	Internet mail
POP3	110	Post Office Protocol (POP) mailbox
IMAP	143	Internet Message Access Protocol (IMAP) Mailbox
Telnet	23	Remote login
SSH	22	Secure remote logn

2.1.4.2 Registered Ports เป็นพอร์ตที่มีหมายเลขตั้งแต่ 1,024 ขึ้นไปและให้บริการอื่นๆ นอกจาก Well Know Ports

2.2 ระบบปฏิบัติการ Kali Linux

ระบบปฏิบัติการ Kali Linux เป็นระบบปฏิบัติการที่ถูกสร้างขึ้นมาเพื่อใช้ทดสอบความปลอดภัยของระบบซึ่งมีเครื่องมือที่จำเป็นต่อการทดสอบความปลอดภัยของระบบโดยมีเครื่องมือในแต่ละหมวดการใช้งานดังนี้

1. Information Gathering รวบรวมข้อมูลจากเครือข่ายเป้าหมาย เพื่อหาวิธีต่าง ๆ ในการโจมตีเครือข่ายหรือระบบ

2. Vulnerability Analysis การระบุภัยคุกคามและความเสี่ยงในระบบคอมพิวเตอร์ แอปพลิเคชันและโครงข่ายโครงสร้างพื้นฐาน

3. Wireless Attacks การโจมตีโครงข่ายไร้สาย เช่น Sniffing, Evil twins

4. Web Applications ใช้ในการทดสอบความปลอดภัยของ Web Applications

5. Exploitation Tools ใช้ในการเจาะช่องโหว่

6. Forensics Tools รวบรวมหลักฐานเพื่อใช้ในการสืบสวนต่อเนื่องหลังเกิดเหตุการณ์ไม่พึงประสงค์ โดยเกี่ยวข้องกับการเก็บรักษา, ค้นหา, คัดแยก, ทำเป็นเอกสาร, และตีความข้อมูลคอมพิวเตอร์ที่เกี่ยวข้อง

7. Stress Testing ใช้เพื่อสร้างการโจมตี DoS หรือการทดสอบการใช้งานให้ทำงานเกินข้อจำกัดจากกระบวนการทำงานปรกติ

8. Sniffing & Spoofing ใช้ในการดักจับข้อมูลและตรวจสอบ traffic ของเว็บ รวมไปถึงเครื่องมือ ปลอมแปลงเครือข่าย

9. Password Attacks ใช้ในการโจมตีรหัสผ่าน 2 รูปแบบ Dictionary Attack และ Brute force Attack

10. Maintaining Access ใช้ในการรักษาสิทธิในการเชื่อมต่อและการเข้าถึงกับเครื่องที่ถูกโจมตี

11. Reverse Engineering ใช้ในการแยกแยะและระบุรายละเอียดของการโจมตีที่ว่าผู้โจมตีเข้ามาในระบบอย่างไร และขั้นตอนในการเข้าถึงในระบบ

12. Reporting Tools ใช้ในการบันทึกกระบวนการทดสอบและผลการทดสอบ

13. Hardware Hacking ใช้ในการเขียน software เพื่อติดตั้งลงใน hardware โดยเครื่องมือที่นำมาใช้ในปริญญานิพนธ์นี้อยู่ในหมวด Vulnerability Analysis

2.2.1 Nmap

เป็นเครื่องมือตัวหนึ่งที่ได้รับความนิยมเป็นอย่างสูง และถูกนำไปใช้กันอย่างแพร่หลายโดยเฉพาะผู้ที่ทำงานในด้านการดูแลระบบเครือข่ายและระบบความมั่นคงปลอดภัยโดย Nmap นั้นมีความสามารถโดยทั่วไปหลักๆ ดังนี้

2.2.1.1 Host Discovery เป็นการค้นหาอุปกรณ์ที่กำลังทำงานอยู่ในระบบเครือข่ายเป้าหมายโดยค่าเริ่มต้น Nmap มักจะทำการสแกนแบบเชิงลึกในอุปกรณ์ที่พบว่าสามารถค้นหาได้ในช่วงระหว่างการ ping scan stage ซึ่งเป็นการประหยัดเวลาและแบนด์วิดท์เมื่อเปรียบเทียบกับการสแกน IP Address ทั้งหมด(full scan) วิธีการนี้จะไม่เป็นไปตามอุดมคติในทุก ๆ กรณี ซึ่งกรณีที่เราต้องการสแกน IP (-PN) หรือต้องการค้นหา host เพียงอย่างเดียว (-sP) โดย Nmap จะเสนอตัวเลือกขั้นสูงที่หลากหลายให้แก่กรณีดังต่อไปนี้

1. List Scan (-sL) List scan เป็นการ Host discovery ที่ลดรายละเอียดลงโดยแค่จะลิสต์แต่ละ host ของเครือข่ายที่กำหนด ปราศจากการส่ง packet ใดๆ ไปที่ host เป้าหมาย โดยค่าเริ่มต้น Nmap ยังคงทำการ reverse - DNS ที่ host เพื่อให้ทราบถึงชื่อ โดย Nmap นั้นยังแสดงจำนวน IP address ทั้งหมดเมื่อทำการค้นหาเสร็จสิ้น การ list scan นั้นเป็นการตรวจสอบที่ดีเพื่อทำให้มั่นใจว่าเรามี IP address ที่ถูกต้อง ถ้า host นั้นมี domain names ที่ไม่สามารถระบุได้ มันคุ้มค่าที่จะตรวจสอบเพื่อป้องกันการสแกนเครือข่ายของบริษัทที่ผิดพลาด

2. No port scan (-sn) ตัวเลือกคำสั่งนี้บอกให้ Nmap ไม่ต้องทำการสแกนพอร์ต หลังจากทำการ Host discovery และการแสดงเพียง host ที่ตอบสนองต่อ host discovery probe เป็นที่รู้จักกันอีกอย่างว่า ping scan วิธีการนี้เป็นการสแกนที่เชิงลึกกว่า List scan

3. No Ping (-Pn) คำสั่งนี้จะข้ามกระบวนการ Nmap discovery ซึ่งมักจะจำเป็นต่อหลายๆ firewall และ host ที่ไม่มีการตอบสนองต่อการ ping การยกเลิก host discovery ด้วยคำสั่ง -Pn ทำให้ Nmap ต้องทำทุก ๆ port ที่ระบุ

2.2.1.2 Port Scanning (-p) คือกระบวนการที่ใช้ในการติดต่อไปที่พอร์ตของ TCP หรือ UDP ของเครื่องเป้าหมายและมีจุดประสงค์ในการตรวจสอบเพื่อหาบริการที่ระบบรอรับการเชื่อมต่อหรืออยู่ในสถานะที่ให้บริการโดยมีจุดประสงค์อื่น ๆ การค้นหาบริการที่ทำงานอยู่บน TCP หรือ UDP ว่ามีบริการใดทำงานอยู่ เช่น HTTP ที่ port 80 เป็นต้น ค้นหาแอปพลิเคชันที่ทำงานบนเครื่องเป้าหมาย เช่น web server โดย Nmap มีการแบ่งประเภทของพอร์ตออกเป็น 6 สถานะ โดยสถานะเหล่านี้ไม่ใช่สถานะที่แท้จริงของพอร์ตแต่เป็นการอธิบายจากมุมมองของ Nmap โดย 6 สถานะดังกล่าวมีดังนี้

1. Open การร้องขอนั้นได้รับการยอมรับอย่างรวดเร็วจาก TCP connection, UDP datagrams หรือ SCTP ในพอร์ตนี้การค้นหาพอร์ตเหล่านี้ถือเป็นเป้าหมายหลักของการสแกนพอร์ต open พอร์ตเหล่านี้เป็นที่น่าสนใจเพราะมันแสดงบริการที่ใช้บนเครือข่าย

2. Closed พอร์ตนั้นสามารถทำการเข้าถึงได้ (สามารถได้รับและตอบสนองต่อ Nmap probe packet) แต่ไม่มีบริการ

3. Filtered เป็นพอร์ตที่ Nmap ไม่สามารถตัดสินได้ว่า พอร์ตนั้นเปิดอยู่ เพราะ ตัวกรองแพ็คเกจ นั้นป้องกัน probes จากการเข้าถึงพอร์ต

4. Unfiltered สถานะ unfiltered นั้นหมายความว่าพอร์ตนั้นสามารถเข้าถึงได้ แต่ Nmap นั้นไม่สามารถตัดสินได้ว่าพอร์ตนั้น เปิด หรือ ปิด

5. Open|filtered ในการที่ Nmap จะบอกสถานะนี้เมื่อมันไม่สามารถตัดสินได้ว่าพอร์ตนั้น open หรือ filtered มันจะเกิดขึ้นในประเภทของการสแกนซึ่ง open port นั้นไม่ตอบสนอง การขาดซึ่งการตอบสนองสามารถหมายความได้ว่า packet filter กรอง probe หรือ มีการกรองการตอบสนองต่าง ๆ ที่ออกมา

6. Closed|filtered สถานะนี้จะถูกใช้เมื่อ Nmap ไม่สามารถตัดสินได้ว่าพอร์ตนั้น closed หรือ filtered

ซึ่งเทคนิคการค้นหาพอร์ต (Port Scanning Techniques) มีทั้งหมด 10 วิธีดังนี้

1. TCP SYN Stealth (-sS) เป็นวิธีที่เป็นที่นิยมเพราะเป็นวิธีที่ได้ผลเร็วในการค้นหาพอร์ตและเป็นที่นิยมในโปรโตคอล TCP

2. TCP Connect (-sT) มักจะถูกใช้สำหรับผู้ใช้งาน UNIX และเป้าหมายที่เป็น IPv6 เพราะว่า SYN Scan ไม่สามารถทำงานในกรณีนี้ได้

3. UDP (-sU) เป็นการค้นหาพอร์ต UDP โดยที่พอร์ต UDP ยังคงมีช่องโหว่ด้านความปลอดภัยมากพอสมควร

4. TCP FIN. Xmas และ Null (-sF, -sX, -sN) การค้นหาเหล่านี้มีไว้เพื่อวัตถุประสงค์เฉพาะในการเข้าไปดูด้านหลัง Firewall เพื่อที่จะดูระบบด้านหลัง แต่เป็นที่น่าเสียดายที่วิธีการเหล่านี้ขึ้นอยู่กับพฤติกรรมของบางระบบ

5. TCP ACK (-sA) ปกติ ACK scan ใช้ในการวางแผนการตั้งค่ากฎของ firewall โดยเฉพาะอย่างยิ่งมันจะช่วยให้เข้าใจว่ากฎของ Firewall เป็น stateful หรือไม่

6. TCP Window (-sW) Window Scan เหมือนกับ ACK Scan ต่างกันที่มันจะสามารถตรวจจับพอร์ตที่เปิดกับปิดได้

7. TCP Malmon (-sM) เป็นการค้นหาที่หลบเลี่ยง Firewall ที่ไม่ขัดเจนมีลักษณะคล้ายกับ FIN Scan แต่ยังรวม ACK Flag ไปด้วยและยังอนุญาตให้ได้รับแพคเก็ตมากขึ้นที่โดน Firewall กรอง โดยมีข้อเสียคือสามารถใช้งานได้กับระบบน้อยกว่า FIN Scan

8. TCP Idle (-sI <zombie host>) Idle Scan เป็น การค้นหาแบบลักลอบที่สุดในทั้งหมด และบางครั้งใช้ประโยชน์จากหมายเลข IP ที่น่าเชื่อถือแต่เป็นที่น่าเสียดายการค้นหาแบบนี้ช้าและซับซ้อน

9. IP protocol (-sO) Protocol Scan ตัดสินว่า IP protocol (TCP, ICMP, IGMP และอื่นๆ) อันไหนจะรองรับโดยเครื่องเป้าหมาย ซึ่งไม่ใช่เทคนิคการค้นหาพอร์ต เนื่องจากทำการวนหมายเลข IP protocol แทนที่จะเป็นหมายเลขพอร์ต TCP หรือ UDP โดยการใช้งานยังคงใช้ตัวเลือก -p เลือกค้นหาหมายเลขโปรโตคอล รายงานผลด้วยรูปแบบตารางพอร์ตปกติ

10. TCP FTP bounce (-b <FTP bounce proxy>) เป็นการค้นหาที่เลิกใช้งานเพราะผู้ให้บริการ FTP ส่วนใหญ่ลงตัวเสริมเพื่อป้องกันเรียบร้อยแล้วแต่ก็เป็นทางที่ดีที่จะสอดส่องผ่านข้อจำกัดของ firewalls ตอนทำงานอยู่

2.2.1.3 Remote OS Detection เป็นการตรวจสอบระบบปฏิบัติการ โดยการใช้ TCP/IP stack fingerprinting โดย Nmap ส่งชุดแพคเก็ตของ TCP และ UDP ไปยัง host ระยะไกลวิเคราะห์ทุกบิตที่ทำการตอบสนอง หลังจากการทดสอบ TCP ISN sampling, IP ID sampling, initial window size check หลังจาก Nmap เปรียบเทียบผลลัพธ์กับฐานข้อมูล nmap-os-db และแสดงผลลัพธ์เมื่อมีค่าที่ตรงกัน ในแต่ละ fingerprint จะประกอบด้วย ชื่อของระบบปฏิบัติการ ชื่อผู้ผลิต ชื่อรุ่น และชื่ออุปกรณ์ fingerprint ส่วนใหญ่มักจะประกอบด้วย Common Platform Enumeration (CPE) โดยมีการตรวจสอบ 3 รูปแบบดังนี้

1. Enable OS detection (-O) ทำให้สามารถตรวจสอบ OS, อีกทางเลือกหนึ่งคือใช้ -A เพื่อสามารถตรวจสอบ OS ร่วมกับกระบวนการอื่น ๆ

2. -osscan-limit เป็นการตรวจสอบ OS ที่มีประสิทธิภาพเพิ่มขึ้นมากถ้าหากพบหนึ่ง open และ หนึ่ง closed TCP port คำสั่งนี้จะทำให้ Nmap ไม่สแกน OS ที่ไม่ตรงกับคุณสมบัติข้างต้น ทำให้ประหยัดเวลาโดยเฉพาะในคำสั่งค้นหา -Pn

3. --osscan-guess เมื่อ Nmap ไม่สามารถทำการตรวจพบกับ OS ที่ตรงกับฐานข้อมูล ในบางครั้ง Nmap จะเสนอค่าที่ใกล้เคียงมากที่สุดซึ่งต้องมีความใกล้เคียงอย่างมากเพื่อให้แสดงค่าในค่าเริ่มต้น โดยคำสั่งนี้จะทำให้ Nmap เดาโดยอิงจากค่าความใกล้เคียงที่เทียบจากฐานข้อมูล Nmap จะมีการบอกเมื่อมีการแสดง OS ที่มีข้อมูลไม่ตรงกับฐานข้อมูลและแสดงระดับของความถูกต้องในรูปของเปอร์เซ็นต์

2.2.1.4 Script Scanning (--script) เป็นการใช้สคริปต์เพื่อเพิ่มความสามารถของ Nmap ในการทำงานด้านอื่นๆ โดยทั่วไปจะเกี่ยวข้องกับการตรวจสอบความมั่นคงปลอดภัยของระบบ ความสามารถของ Nmap ในการเรียกใช้สคริปต์นี้มีชื่อว่า Nmap Scripting Engine

2.2.1.5 Detecting and Subverting Firewalls and Intrusion Detection Systems

Firewall ทำให้การ mapping เครือข่ายนั้นยากมากขึ้น อย่างไรก็ตาม Nmap นั้นมีหลายความสามารถที่จะช่วยให้เข้าใจความซับซ้อนของเครือข่าย และเพื่อตรวจสอบว่า filter นั้นทำงานได้อย่างเจตนาไว้

Nmap เป็นเครื่องมือที่รองรับกลไกสำหรับการ Bypass ที่ช่วยป้องกันการใช้งานอย่างหละหลวม ซึ่งวิธีการที่ดีที่สุดในการทำความเข้าใจความปลอดภัยในเครือข่าย คือ การที่เราใช้แนวคิดของผู้โจมตีเป็นหลักรวมไปถึงการใช้ FTP bounce scan, idle scan, การโจมตี fragment หรือพยายามเจาะเข้าไปใน proxies ของเราเอง นอกจากนี้ความเข้มงวดในการใช้งานเครือข่ายของบริษัทต่าง ๆ นั้นถูกเพิ่มการตรวจสอบการใช้งานเครือข่ายด้วย Intrusion Detection Systems (IDS) ซึ่งผลิตภัณฑ์เหล่านี้ถูกเปลี่ยนให้เป็น intrusion prevention systems (IPS) ซึ่งคอยทำหน้าที่บล็อกทราฟฟิกที่ถือว่าเป็นอันตราย

ขั้นแรกในการ Bypass firewall คือการเข้าใจในกระบวนการการ bypass firewall Nmap นั้นถูกแยกระหว่าง port ที่สามารถเข้าถึงได้แต่ถูกปิดไม่ให้เข้า และ port ที่ถูก filtered โดยเทคนิคที่มีประสิทธิภาพที่จะใช้เริ่มต้นคือการสแกน port SYN จากนั้นไปต่อด้วยวิธีการที่แปลกใหม่มากขึ้น เช่น ACK scan และ IP ID ตามลำดับเพื่อความเข้าใจในเครือข่ายที่มากขึ้น

การ bypass rules จะเป็นเป้าหมายหลักเมื่อมีการวางแผน Firewall rules ที่เป็นประโยชน์ และเมื่อNmap มีการใช้งานอย่างหลากหลายโดยแม้ว่าเทคนิคเหล่านี้ จะมีประสิทธิภาพเฉพาะแค่กับเครือข่ายที่มีการตั้งค่าที่หละหลวม แต่กลับกลายเป็นเรื่องธรรมดาทั่วไป การใช้วิธีการเดิม ๆ ในการทำงานมีอัตราการประสบความสำเร็จต่ำ ดังนั้นควรใช้วิธีการที่หลากหลาย เนื่องจากผู้โจมตีต้องการเพียงแค่การตั้งค่าที่ผิดพลาดเพียงอันเดียวเพื่อบรรลุเป้าหมาย ในขณะที่คนป้องกันเครือข่ายจะต้องปิดช่องโหว่เหล่านี้ทั้งหมดไม่ว่าจะเป็นจากการใช้ SYN และ ACK scan การใช้งานของ port การโจมตีด้วย IPv6, IP ID Idle Scanning, Multiple Ping Probes, MAC Address Spoofing , FTP Bounce Scans เป็นต้น

Intrusion detection system (IDS) คือซอฟต์แวร์ (software) หรือ ฮาร์ดแวร์ (hardware) ที่ได้รับการออกแบบมาเพื่อให้ตรวจสอบการเชื่อมต่อที่ไม่พึงประสงค์หรือความพยายามที่จะเข้ามาทำอันตรายต่อเครือข่าย โดยผ่านระบบต่าง ๆ เช่น Internet, Lan เป็นต้น โดยการโจมตีนั้นอาจจะเกิดจาก cracker, Worm หรือ Malware ต่าง ๆ และข้อจำกัดของ Intrusion detection system (IDS) นั้นก็คือไม่สามารถที่จะตรวจสอบ Packet ที่เข้ารหัสได้ องค์ประกอบของ Ids นั้นมีหลายหลายส่วนแต่ส่วนประกอบของ Intrusion Detection System (IDS) ที่สำคัญนั้นมีอยู่สามส่วนได้แก่ 1) Sensor คือส่วนที่จะสร้างเหตุการณ์ที่เกี่ยวกับความปลอดภัยขึ้นมา 2) Console คือส่วนที่จะตรวจจับเหตุการณ์ต่าง, แจ้งเตือน รวมไปถึงการควบคุม Sensor 3) Engine เป็นส่วน ที่จะบันทึกเหตุการณ์จาก sensor ลงใน Database และจะแจ้งเตือนตามกฎที่ได้ตั้งเอาไว้ใน IDS คำศัพท์เฉพาะที่เกี่ยวกับเรื่อง IDS จะแสดงดังตารางที่ 2.2

ตารางที่ 2.2 คำศัพท์เฉพาะที่เกี่ยวกับ IDS

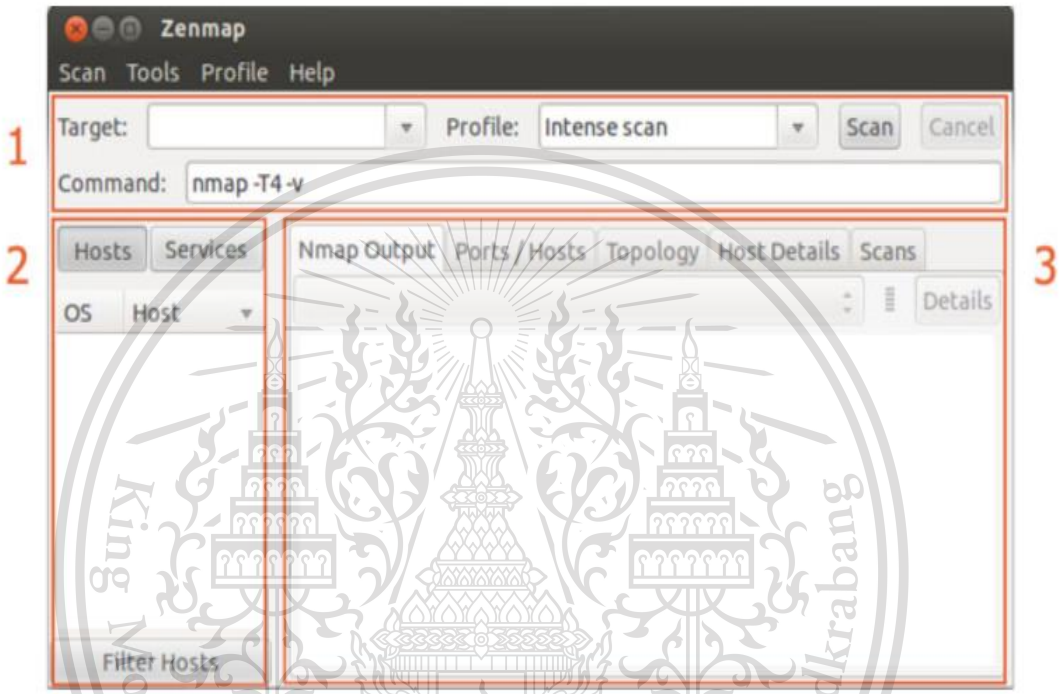
คำศัพท์	คำอธิบาย
Alert/Alarm	การแจ้งเตือนเมื่อระบบถูกโจมตี
True attack stimulus	เหตุการณ์ที่กระตุ้นให้ IDS เกิดการแจ้งเตือนเมื่อเกิดการโจมตีขึ้นจริง
False attack stimulus	เหตุการณ์ที่กระตุ้นให้เกิดการแจ้งเตือนเมื่อไม่มีการโจมตีขึ้นจริง
False (False Positive)	การแจ้งเตือนเมื่อไม่เกิดการโจมตีขึ้นจริง
False negative	การที่ไม่แจ้งเตือนเมื่อเกิดการโจมตีขึ้นจริง
Noise	สิ่งรบกวนที่สามารถทำให้เกิดการแจ้งเตือนที่ผิดพลาดขึ้นได้
Alarm filtering	การดำเนินการแยกการแจ้งเตือนที่ผิดพลาดออกจากการโจมตีจริง เพื่อให้การแจ้งเตือนมีความแม่นยำมากยิ่งขึ้น

ซึ่งวิธีการในการตรวจสอบการบุกรุกอาจสามารถที่จะแบ่งได้ออกเป็น 2 วิธีการได้แก่ Misuse Based Detection และ Anomaly Based Detection

1. Misuse Based Detection เป็นระบบการตรวจสอบโดยการใช้ signature ของข้อมูลจึงเป็นที่มาของอีกชื่อหนึ่งนั่นคือ signature based หรือ knowledge based detection โดยปกติแล้ว การตรวจสอบด้วยระบบนี้จะสามารถตรวจสอบได้เฉพาะการโจมตีที่ทราบอยู่แล้วในระบบฐานข้อมูล โดยการเซ็ตกฎหรือตัวกรองในระบบตรวจสอบ หากเป็นการโจมตีหรือสิ่งแปลกปลอมใหม่ๆ ที่นอกเหนือจากระบบฐานข้อมูลการโจมตีแล้ว ระบบ misuse นี้จะตรวจจับไม่ได้ ระบบ Misuse Based นี้โดยทั่วไปแล้วจะทำงานคล้ายกับระบบของโปรแกรมป้องกันไวรัสซึ่งทำการเปรียบเทียบ signature ของข้อมูลที่วิ่งไปมาในระบบกับฐานข้อมูลขนาดใหญ่ที่มีอยู่แล้วซึ่งหากว่าเหมือนกับในฐานข้อมูลก็แสดงว่าข้อมูลดังกล่าวอาจจะเป็นการบุกรุกหรือประสงค์ร้ายนั่นเอง ซึ่งอาจจะมีจุดอ่อนตรงที่ไม่สามารถตรวจสอบได้หากวิธีในการบุกรุกนั้นเป็นวิธีใหม่ๆ

2. Anomaly Based Detection การทำงานของระบบนี้จะเป็นการตรวจสอบ pattern ของข้อมูลหรือจะเรียกว่าพฤติกรรมต่างของข้อมูลที่วิ่งอยู่ในระบบเพื่อเรียนรู้ว่าอะไรคือสิ่งปกติและผิดปกติภายในระบบโดยที่ระบบจะมีกระบวนการเรียนรู้ด้วยตัวเอง เหมือนดังเช่นกับระบบ spam filter โดยปกติแล้วระบบนี้จะถูกตั้งค่าโดยผู้ดูแลระบบเครือข่ายโดยที่ผู้ดูแลอาจจะกำหนดเส้นแบ่งว่าพฤติกรรมไหนถือว่าเป็นพฤติกรรมที่ปกติโดยอาจจะพิจารณาจาก traffic, พฤติกรรม, protocol หรือขนาดของข้อมูลเป็นต้น ดังนั้นจะทราบได้ทันทีว่าพฤติกรรมไหนเป็นพฤติกรรมที่เข้าข่ายการโจมตีระบบนั่นเอง

2.2.1.8 Zenmap เป็น Nmap สำหรับผู้ใช้ในระดับเริ่มต้นที่ไม่ถนัดจะใช้โปรแกรม Nmap ผ่านทาง Command-line สามารถเลือกใช้โปรแกรม Zenmap ซึ่งเป็น GUI front-end ของโปรแกรม Nmap ที่มาพร้อมกับตัวติดตั้งแทนได้ โดยเมื่อเปิดโปรแกรม Zenmap ขึ้นมาจะพบกับหน้าต่าง ดังรูปที่ 2.8



รูปที่ 2.8 หน้าต่างหลักของโปรแกรม Zenmap

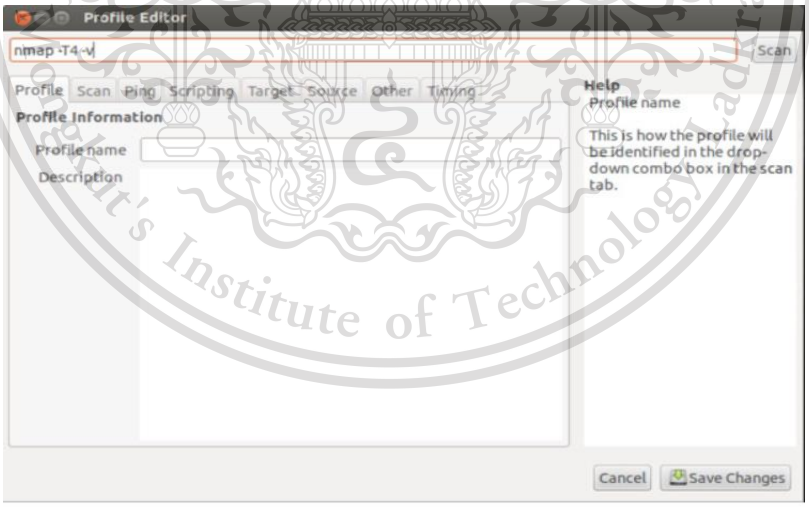
จากรูปที่ 2.8 หน้าต่างหลักของโปรแกรม Zenmap แบ่งออกเป็น 3 ส่วนดังนี้

ส่วนที่ 1 เกี่ยวข้องกับคำสั่ง ประกอบด้วย 1.) Target หมายถึง ระบบเป้าหมายที่จะทำการตรวจสอบ ในช่องนี้สามารถระบุเป็น IP address, Hostname หรือ Domain name 2.) Profile หมายถึง ลักษณะการสแกนในรูปแบบต่าง ๆ เช่น Ping scan หรือการสแกนเฉพาะพอร์ต TCP โดยโปรแกรม Zenmap จะมีโปรไฟล์มาให้เลือกใช้อยู่แล้วส่วนหนึ่ง 3.) Command หมายถึง คำสั่งที่จะใช้ในการประมวลผล ซึ่งจะปรากฏหลังจากที่ระบุค่าในช่อง Target และเลือกโปรไฟล์แล้ว ทั้งนี้ผู้ใช้สามารถระบุคำสั่งลงในช่องนี้ได้โดยตรง โดยไม่ต้องระบุค่าในช่อง Target และเลือกโปรไฟล์ได้

ส่วนที่ 2 เป็นหน้าต่างสำหรับกรองผลการสแกนตามอุปกรณ์หรือบริการที่ตรวจพบ เมื่อเลือกรายการใด ๆ ในหน้าต่างนี้จะทำให้หน้าต่างทางด้านขวาแสดงผลลัพธ์ที่สัมพันธ์กับรายการที่เลือกไว้ แบ่งออกเป็น 2 ส่วนย่อยคือ 1.) Hosts คือ รายการอุปกรณ์ทั้งหมดที่ตรวจพบว่ากำลังทำงานอยู่ในระบบเครือข่าย โดยแสดงข้อมูลของระบบปฏิบัติการ

ที่ใช้, Hostname (ถ้ามี) และ IP address ของแต่ละเครื่อง 2.) Services คือ บริการที่ตรวจพบว่ากำลังเปิดใช้งานอยู่บนระบบ

ส่วนที่ 3 คือหน้าต่างแสดงผลการสแกนระบบเป้าหมาย แบ่งออกเป็น 5 แท็บได้แก่ 1.) Nmap Output คือ แสดงผลลัพธ์การสแกนที่ได้ทั้งหมด ซึ่งจะมีหน้าตาเหมือนกับผลลัพธ์ที่ได้จากการเรียกใช้ Nmap ผ่านทาง Command line 2.) Ports / Hosts คือ รายละเอียดของบริการที่เปิดใช้งานอยู่บนระบบเป้าหมาย ประกอบไปด้วยหมายเลขและสถานะของพอร์ต โพรโทคอล ชื่อและเวอร์ชันของบริการ 3.) Topology คือ รูปแบบโครงสร้างการเชื่อมต่อของอุปกรณ์ที่ตรวจพบภายในเครือข่าย ซึ่งสร้างขึ้นจากผลลัพธ์ที่ได้จากการสแกน 4.) Host Details คือ รายละเอียดของระบบเป้าหมายที่ตรวจพบ เช่น IP address, MAC address และ Hostname 5.) Scans คือ รายการคำสั่งที่เคยเรียกใช้ในการใช้งานทั่วไปนั้น ผู้ใช้เพียงระบุค่าในช่อง Target จากนั้นเลือกโปรไฟล์แล้วคลิกปุ่ม Scan ก็ถือว่าเป็นอันเสร็จสิ้น แต่ในกรณีที่ผู้ใช้ต้องการสแกนในรูปแบบอื่น ๆ นอกเหนือจากโปรไฟล์ที่มีให้เลือก ผู้ใช้สามารถเพิ่มโปรไฟล์ได้เองโดยการเลือกเมนู Profile -> New Profile or Command จะพบกับหน้าต่าง Profile Editor ดังรูปที่ 2.9 โดยคำสั่งพื้นฐานของ Nmap จะอยู่ในแท็บ Scan และ Ping ผู้ใช้เพียงระบุชื่อโปรไฟล์ในแท็บ Profile จากนั้นเลือกคำสั่งที่ต้องการในแท็บอื่น ๆ แล้วคลิกปุ่ม Save Changes ก็สามารถนำโปรไฟล์ดังกล่าวไปใช้งานได้



รูปที่ 2.9 หน้าต่าง Profile Editor

2.2.1.7 Nmap Output Format เป็นการกำหนดผลลัพธ์ที่ได้จาก Nmap โดยมีรูปแบบผลลัพธ์ที่เป็นพื้นฐานโดยมีทั้งหมด 5 รูปแบบดังนี้

1. Interactive Output เป็นผลลัพธ์ที่ Nmap ส่งออกมาเป็น Output Stream โดยไม่จำเป็นต้องไปตั้งค่าใด ๆ ดังนั้นจึงไม่มีตัวเลือกใด ๆ บน Command-Line โดยที่ Interactive Mode ให้ความสนใจกับการอ่านผลลัพธ์ของผู้ใช้งานโดยตรงและมีลักษณะที่เป็นตารางของพอร์ตที่ดูน่าสนใจดังรูปที่ 2.10

```
# nmap -T4 -A -p- scanme.nmap.org

Starting Nmap ( http://nmap.org )
Interesting ports on scanme.nmap.org (64.13.134.52):
Not shown: 65529 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 4.3 (protocol 2.0)
25/tcp  closed smtp
53/tcp  open   domain  ISC BIND 9.3.4
70/tcp  closed gopher
80/tcp  open   http    Apache httpd 2.2.2 ((Fedora))
|_ HTML title: Go ahead and ScanMe!
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.21, Linux 2.6.23

TRACEROUTE (using port 22/tcp)
HOP RTT   ADDRESS
1   16.92 nodem-msfc-vl245-act-security-gw-1-113.ucsd.edu (132.239.1.113)
[... nine similar lines cut ...]
11  21.97 scanme.nmap.org (64.13.134.52)

OS and Service detection performed. Please report any incorrect results↵
 at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 168.10 seconds
```

รูปที่ 2.10 Interactive Output ของ Nmap

2. Normal Output (-oN) มีความคล้ายคลึงกับ Interactive Output ซึ่งมีความแตกต่างจาก Interactive Output ในหลายๆ ทางเช่น Interactive Output จะรวมข้อความเช่นการประมาณเวลาการค้นหาที่เสร็จสมบูรณ์และการแจ้งเตือนพอร์ตที่เปิด โดยที่ Normal Output จะละเว้นสิ่งที่ไม่จำเป็นเมื่อการค้นหาเสร็จสิ้นและโดยที่ตารางพอร์ตหน้าสุดท้ายจะถูกรายงานออกมารวมไปถึงเวลาและวันที่เริ่มใช้คำสั่งจะถูกรายงานมาในบรรทัดแรก ดังรูปที่ 2.11

```
# nmap -T4 -A -p- -oN - scanme.nmap.org

# Nmap 4.68 scan initiated Tue Jul 15 07:27:26 2008 as: nmap -T4 -A -p- ↵
-oN - scanme.nmap.org
Interesting ports on scanme.nmap.org (64.13.134.52):
Not shown: 65529 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 4.3 (protocol 2.0)
25/tcp  closed smtp
53/tcp  open   domain  ISC BIND 9.3.4
70/tcp  closed gopher
80/tcp  open   http    Apache httpd 2.2.2 ((Fedora))
|_ HTML title: Go ahead and ScanMe!
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.21, Linux 2.6.23

TRACEROUTE (using port 22/tcp)
HOP RTT   ADDRESS
1   2.98  nodem-msfc-v1245-act-security-gw-1-113.ucsd.edu (132.239.1.113)
[... nine similar lines cut ...]
11  13.34 scanme.nmap.org (64.13.134.52)

OS and Service detection performed. Please report any incorrect results ↵
at http://nmap.org/submit/ .
# Nmap done at Tue Jul 15 07:29:45 2008 -- 1 IP address (1 host up) ↵
scanned in 138.938 seconds
```

รูปที่ 2.11 Normal Output ของ Nmap

3. XML Output (-oX) XML เป็นรูปแบบที่ง่ายต่อการแจกแจงโดยซอฟต์แวร์ การแจกแจง XML สามารถใช้ได้กับภาษาคอมพิวเตอร์ส่วนใหญ่ได้เช่น C/C++, Perl, Python และ Java ซึ่งผลลัพธ์ที่เป็น XML สามารถกลายเป็นรูปแบบอื่นได้เช่น HTML ซึ่งผลลัพธ์แบบ XML แสดงดังรูปที่ 2.12

```
# nmap -T4 -A -p- -oX - scanme.nmap.org
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/usr/share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 4.68 scan initiated Tue Jul 15 07:27:26 2008 as:
         nmap -T4 -A -p- -oX - scanme.nmap.org -->
<nmaprun scanner="nmap" args="nmap -T4 -A -p- -oX - scanme.nmap.org"
         start="1216106846" startstr="Tue Jul 15 07:27:26 2008"
         version="4.68" xmloutputversion="1.02">
  <scaninfo type="syn" protocol="tcp" numservices="65535" services="1-65535" />
  <verbose level="0" /> <debugging level="0" />
   <host starttime="1216106846" endtime="1216106985">
    <status state="up" reason="reset" />
    <address addr="64.13.134.52" addrtype="ipv4" />
    <hostnames><hostname name="scanme.nmap.org" type="PTR" /></hostnames>
    <ports><extraports state="filtered" count="65529">
        <extrareasons reason="no-responses" count="65529" /></extraports>
      <port protocol="tcp" portid="22">
        <state state="open" reason="syn-ack" reason_ttl="52" />
        <service name="ssh" product="OpenSSH" version="4.3"
                 extrainfo="protocol 2.0" method="probed" conf="10" /> </port>
      <!-- Several port elements removed for brevity -->
      <port protocol="tcp" portid="80">
        <state state="open" reason="syn-ack" reason_ttl="52" />
        <service name="http" product="Apache httpd" version="2.2.2"
                 extrainfo="(Fedora)" method="probed" conf="10" />
        <script id="HTML title" output="Go ahead and ScanMe!" /> </port>
      <port protocol="tcp" portid="113">
        <state state="closed" reason="reset" reason_ttl="52" />
        <service name="auth" method="table" conf="3" /> </port> </ports>
    <os>
```

รูปที่ 2.12 XML Output ของ Nmap

4. Grepable Output (-oG) รูปแบบนี้เป็นรูปแบบที่ง่ายที่จะจัดการบน command line ด้วยเครื่องมือของ UNIX เช่น grep, awk, cut และ diff แต่ละ host จะถูกเขียนลงบน 1 บรรทัดแล้วถูกจัดการด้วย tab, slash และ เครื่องหมาย comma ดังรูปที่ 2.13 ใช้ในการกำหนดพื้นที่ของผลลัพธ์ซึ่งมีประโยชน์สำหรับผลลัพธ์แบบ grokking แต่ในรูปแบบ XML เป็นที่ต้องการสำหรับงานที่สำคัญยิ่งขึ้นเนื่องจากมีเสถียรภาพและข้อมูลมากกว่า

```
# nmap -oG - -T4 -A -v scanme.nmap.org
# Nmap 4.68 scan initiated [time] as: nmap -oG - -T4 -A -v scanme.nmap.org
# Ports scanned: TCP(1715;1-1027,1029-1033,...,65301) UDP(0;) PROTOCOLS(0;)
Host: 64.13.134.52 (scanme.nmap.org)    Ports: 22/open/tcp//ssh//OpenSSH 4.3↵
(protocol 2.0)/, 25/closed/tcp//smtp///, 53/open/tcp//domain//ISC BIND ↵
9.3.4/,70/closed/tcp//gopher///, 80/open/tcp//http//Apache httpd 2.2.2 ↵
((Fedora))/,113/closed/tcp//auth/// Ignored State: filtered (1709)  OS: ↵
Linux 2.6.20-1 (Fedora Core 5) Seq Index: 203  IP ID Seq: All zeros
# Nmap done at [time] -- 1 IP address (1 host up) scanned in 34.96 seconds
```

รูปที่ 2.13 Grepable Output ของ Nmap

2.2.1.8 Nmap Data File เครื่องมือ Nmap ได้พึ่งพา 6 ไฟล์ข้อมูลสำหรับการทำ Port Scanning และการดำเนินการอื่น ๆ โดยไฟล์ข้อมูลที่ Nmap นำมาใช้ประกอบไปด้วย

1. Well Known Port List (nmap-service) ไฟล์ nmap-service เป็นรายการของชื่อ Port ที่ตรงกับหมายเลขและโปรโตคอลโดยแต่ละหมายเลขจะแสดงถึงพอร์ตที่พบ โดยในบรรทัดส่วนใหญ่จะมีคอมเม้นที่ตรง ในบางครั้งผู้ใช้งานสามารถใช้คำสั่ง grep เพื่อดูคอมเม้นในไฟล์นั้นได้เมื่อ Nmap รายงานบริการของพอร์ตที่เปิดแสดงดังรูปที่ 2.14 แสดงให้เห็นถึงรูปแบบที่ดูง่ายมี 3 คลอลัมน์ที่แยกออกจากกันโดยคลอลัมน์แรกเป็นชื่อบริการ คลอลัมน์ที่ 2 เป็นหมายเลขพอร์ตและโปรโตคอลส่วนคลอลัมน์ที่ 3 เป็นความถี่ที่พบเจอพอร์ตนี้บ่อยมากน้อยเพียงใด โดยในบรรทัดส่วนมากจะต่อด้วยเครื่องหมาย ‘#’ และตามด้วยคอมเม้นแต่ในบางบรรทัดก็จะมีคอมเม้นแต่ไม่มีเครื่องหมาย ‘#’

```
qotd            17/tcp    0.002346  # Quote of the Day
qotd            17/udp    0.009209  # Quote of the Day
msp             18/udp    0.000610  # Message Send Protocol
chargen         19/tcp    0.002559  # ttytst source Character Generator
chargen         19/udp    0.015865  # ttytst source Character Generator
ftp-data        20/tcp    0.001079  # File Transfer [Default Data]
ftp-data        20/udp    0.001878  # File Transfer [Default Data]
ftp             21/tcp    0.197667  # File Transfer [Control]
ftp             21/udp    0.004844  # File Transfer [Control]
ssh             22/tcp    0.182286  # Secure Shell Login
ssh             22/udp    0.003905  # Secure Shell Login
telnet          23/tcp    0.221265
telnet          23/udp    0.006211
priv-mail       24/tcp    0.001154  # any private mail system
priv-mail       24/udp    0.000329    any private mail system
smtp            25/tcp    0.131314  # Simple Mail Transfer
smtp            25/udp    0.001285  # Simple Mail Transfer
```

รูปที่ 2.14 nmap-service

2. Version Scanning DB (nmap-service-probes) เป็นไฟล์ที่ Nmap ใช้ตรวจจับบริการและเวอร์ชันของพอร์ต (ตัวเลือก -sV หรือ -A) ซึ่ง nmap-service-probes จะมีความซับซ้อนกว่า nmap-service ซึ่ง nmap-service-probes จะแสดงดังรูปที่ 2.15

```
##############################NEXT PROBE##############################
# DNS Server status request: http://www.rfc-editor.org/rfc/rfc1035.txt
Probe UDP DNSStatusRequest q|\0\0\x10\0\0\0\0\0\0\0\0|
ports 53,135
match domain m|^\0\0\x90\x04\0\0\0\0\0\0\0|
# This one below came from 2 tested Windows XP boxes
match msrpc m|^\x04\x06\0\0\x10\0\0\0\0\0\0\0|
[...]
##############################NEXT PROBE##############################
Probe UDP Help q|help\r\n\r\n|
ports 7,13,37
match chargen m|@ABCDEFGHIJKLMNOPQRSTUVWXYZ|
match echo m|^help\r\n\r\n$|
match time m|^[\xc0-\xc5]...$|
```

รูปที่ 2.15 nmap-service-probes

3. หมายเลข SunRPC (nmap-rpc) มีลักษณะคล้ายกับ nmap-service ที่มีการจัดวางที่ดูง่ายมีหมายเลขและชื่อแต่จะตัดบางส่วนออกทำให้ดูยากมากกว่า nmap-service เช่นหัวคลอลัมน์ แสดงดังรูปที่ 2.16

```
rpcbind         100000  portmap sunrpc rpcbind
rstatd          100001  rstat rup perfmeter rstat_svc
rusersd         100002  rusers
nfs             100003  nfsprog nfsd
ypserv          100004  ypprog
mountd          100005  mount showmount
rpc.operd       100080  opermsg         # Sun Online-Backup
# DMFE/DAWS (Defense Automated Warning System)
#
Gqsrv           200034  gqsrv
Ppt             200035  ppt
Pmt             200036  pmt
```

รูปที่ 2.16 nmap-rpc

4. Nmap OS Detection DB (nmap-os-db) เป็นไฟล์ข้อมูลที่รวมตัวอย่างจากการที่ได้รับการตอบสนองจากระบบปฏิบัติการไว้เป็นหลักร้อยจากการตรวจจับระบบปฏิบัติการของ Nmap แสดงดังรูปที่ 2.17 ซึ่งจะแบ่งออกเป็น fingerprint หลายๆ อันโดยแต่ละ fingerprint จะรวมชื่อของระบบปฏิบัติการและจะสร้าง classification เพื่อให้ตอบสนองต่อข้อมูล

```
Fingerprint Linux 2.6.11 - 2.6.20
Class Linux | Linux | 2.6.X | general purpose
SEQ(SP=B9-CF%GCD=<7%ISR=C4-D7%TI=Z%II=I%TS=7)
OPS(O1=M5B4ST11NW1%O2=M5B4ST11NW1%O3=M5B4NNT11NW1%O4=M5B4ST11NW1%↵
    O5=M5B4ST11NW1%O6=M5B4ST11)
WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)
ECN(R=Y%DF=Y%T=40%TG=40%W=16D0%O=M5B4NNSNW1%CC=N%Q=)
T1(R=Y%DF=Y%T=40%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=40%TG=40%W=16A0%S=O%A=S+%F=AS%O=M5B4ST11NW1%RD=0%Q=)
T4(R=Y%DF=Y%T=40%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=40%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=40%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=40%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=40%TG=40%TOS=C0%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUL=G%RUD=G)
IE(DFI=N%T=40%TG=40%TOSI=S%CD=S%SI=S%DLI=S)
```

รูปที่ 2.17 nmap-os-db

5. MAC Address Vendor Prefixes (nmap-mac-prefixes) เป็นไฟล์ที่เชื่อมโยง MAC Address Prefixes กับชื่อของผู้ผลิตแสดงดังรูปที่ 2.18 โดยที่ MAC Address ถูกกำหนดโดย IEEE และเรียกรูปแบบนี้ว่า Organizationally Unique Identifier (OUI) โดยที่ผู้ผลิตอุปกรณ์ ethernet จะต้องกำหนด MAC Address มีรูปแบบโดยแบ่งเป็น 2 ส่วนคือ 3 ไบต์แรกจะเป็นรหัสของผู้ผลิตและ 3 ไบต์หลังผู้ผลิตสามารถกำหนดได้เองยกตัวอย่าง MAC Address 00:60:1D:38:32:90 โดยที่ 3 ไบต์แรกคือ 00601D เป็นของผู้ผลิตที่ชื่อว่า Lucent Technologies

```
006017 Tokimec
006018 Stellar ONE
006019 Roche Diagnostics
00601A Keithley Instruments
00601B Mesa Electronics
00601C Telxon
00601D Lucent Technologies
00601E Softlab
00601F Stallion Technologies
006020 Pivotal Networking
006021 DSC
006022 Vicom Systems
006023 Pericom Semiconductor
006024 Gradient Technologies
006025 Active Imaging PLC
006026 Viking Components
```

รูปที่ 2.18 nmap-mac-prefixes

6. IP Protocol Number List (nmap-protocols) เป็นไฟล์ที่เชื่อมโยง 1 ไบต์ที่อยู่ในส่วนหัวของโปรโตคอล IP ให้ตรงกับชื่อของโปรโตคอลแสดงดังรูปที่ 2.19

```
hopopt              HOPOPT      # IPv6 Hop-by-Hop Option
icmp          1     ICMP        # Internet Control Message
igmp                IGMP        # Internet Group Management
ggp                 GGP         # Gateway-to-Gateway
ip                  IP          # IP in IP (encapsulation)
st                  ST          # Stream
tcp           6     TCP         # Transmission Control
cbt           7     CBT         # CBT
egp                 EGP         # Exterior Gateway Protocol
  ...
chaos        16     CHAOS       # Chaos
udp          17     UDP         # User Datagram
```

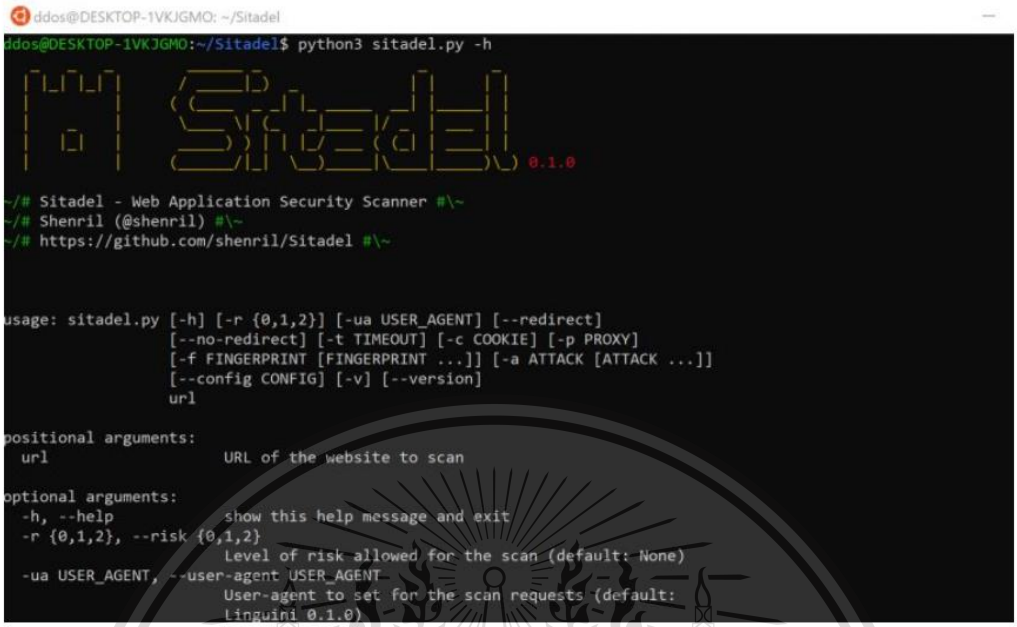
รูปที่ 2.19 nmap-protocols

7. File Related to Scripting หมายถึงสคริปต์ที่ถูกใช้โดย Nmap Scripting Engine ซึ่งจะอยู่ในรูปแบบไฟล์ที่ถูกเก็บในไดเรกทอรี่ย่อยในไดเรกทอรี่หนึ่งอีกทีโดยที่แต่ละสคริปต์จะลงท้ายด้วย. nse โดยทุกไฟล์ในสคริปต์ไดเรกทอรี่เป็นสคริปต์ที่ทำงานได้ยกเว้น script.db ไฟล์นี้เป็น plain text ที่เก็บข้อมูลว่าแต่ละสคริปต์อยู่ในประเภทไหนโดยไฟล์ script.db ไม่ควรไปแก้ไขกับมันโดยตรงให้ใช้คำสั่ง –script-updatedb แทน

8. Using Customized Data Files ในไฟล์ Nmap ทุกไฟล์อาจถูกแทนที่ด้วยเวอร์ชันตามผู้ใช้งานต้องการแต่ไม่สามารถที่จะเปลี่ยนแปลงหรือผสานไฟล์ดั้งเดิมเข้าด้วยกันได้โดยเมื่อ Nmap มองหาไฟล์แต่ละไฟล์จะทำการค้นหาจากชื่อในหลายๆ ไดเรกทอรีและเลือกมาไฟล์แรกที่พบ

2.2.2 Sitadel

Sitadel คือ โมดูลที่เป็นส่วนหนึ่งของ WAScan (Web Application Security Scanner)
สำหรับใช้งานในการตรวจสอบ Web Application และมีฟีจเจอร์ที่หลากหลายในการใช้งานโดยหน้าต่างแสดงหลักในการใช้งานเป็นดังรูปที่ 2.20



รูปที่ 2.20 หน้าต่างหลักของ Sitadel

2.2.2.1 WAScan (Web Application Scanner)

คือ เครื่องมือสแกนช่องโหว่ของแอปพลิเคชันบนเว็บแบบโอเพนซอร์ส ได้รับการออกแบบมาเพื่อค้นหาช่องโหว่ต่างๆโดยใช้วิธีแบบ "black-box" ซึ่งหมายความว่าจะไม่ศึกษาซอร์สโค้ดของเว็บแอปพลิเคชัน แต่จะทำงานเหมือนฟัซเซอร์สแกนหน้าเว็บแอปพลิเคชันที่ใช้งานการแยกลิงก์และแบบฟอร์มและ โจมตีสคริปต์ส่งเพย์โหลดและค้นหาข้อความแสดงข้อผิดพลาด

2.2.2.2 SQL Injection

การโจมตีแบบ SQL Injection นั้นเป็นไม่เพียงแต่เป็นช่องโหว่ที่พบมากที่สุดแต่ ยังเป็นช่องโหว่ที่อันตรายและร้ายแรงมากที่สุดอีกด้วย ทั้งนี้การโจมตีแบบ SQL Injection นั้นผู้โจมตี จะทำการแทรก SQL query หรือ SQL statement ไปยังเว็บแอพพลิเคชั่นโดยผ่านทางช่อง input ถ้าผู้โจมตีสามารถแทรก SQL query ได้สำเร็จ ผู้โจมตีจะสามารถอ่านข้อมูลที่สำคัญของฐานข้อมูล สามารถเปลี่ยนแปลงฐานข้อมูลโดยการเพิ่ม, ปรับปรุง หรือแม้กระทั่งลบข้อมูลจากฐานข้อมูลได้และ ยังสามารถประมวลผลโดยใช้สิทธิ์ของผู้มีสิทธิ์เข้าถึงจากฐานข้อมูลนั้น เช่น การปิด Database Management System (DBMS) สามารถกู้ข้อมูลจากไฟล์ที่ปรากฏบนไฟล์ในระบบ DBMS และใน บางกรณีสามารถออกคำสั่งกับระบบปฏิบัติการได้

2.2.2.3 Brute Force Attack

เป็นการโจมตีโดยการคาดเดาชื่อผู้ใช้งาน, รหัสผ่าน เพื่อเข้าสู่ระบบโดยไม่ได้รับอนุญาต เป็นการโจมตีที่ง่ายและมีอัตราความสำเร็จที่สูง ผู้โจมตีจะใช้เครื่องมือเช่น Applications หรือการวางไฟล์ Scripts เพื่อโจมตีเซิร์ฟเวอร์ของเป้าหมาย ผู้โจมตีจะพยายามเข้าถึง Applications เซิร์ฟเวอร์เป้าหมายโดยค้นหารหัส Searching ที่ถูกต้อง แรงจูงใจของผู้โจมตีนั้นเพื่อหวังขโมยข้อมูลขัดขวางการใช้งาน ทำให้ติด Malware และการใช้งานในส่วนต่างๆ บนระบบที่ใช้งานอยู่ ในขณะที่ผู้โจมตีบางคนยังใช้วิธีการโจมตีด้วยตนเองแต่ปัจจุบันนี้การโจมตีด้วย Brute Force Attack เกือบทั้งหมดดำเนินการด้วย Bot เมื่อโจมตีสำเร็จ จะมีระบบแจ้งเตือนไปที่ผู้โจมตี

2.3 ระบบปฏิบัติการ Raspbian

Raspbian เป็นระบบปฏิบัติการสำหรับติดตั้งใช้งานบนบอร์ดขนาดเล็กนาม Raspberry Pi พัฒนามาจากระบบ Debian Linux เหมาะสำหรับนำมาใช้ทำแล็ป และงานวิจัยเกี่ยวกับระบบคอมพิวเตอร์แบบฝังตัว (Embedded System) โดยที่ Raspbian มีแพ็กแกจให้ใช้งานกว่า 35,000 แพ็กเกจ กล่าวได้ว่าสามารถติดตั้งแพ็กเกจที่ใช้งานใน Debian Linux และ Ubuntu Linux ได้เกือบทุกแพ็กเกจ

2.4 การจัดการ และค้นหาช่องโหว่

การจัดการ และค้นหาช่องโหว่ ประกอบด้วย

2.4.1 การจัดการและค้นหาช่องโหว่

ช่องโหว่ คือ จุดอ่อนในแอปพลิเคชันซึ่งอาจเป็นข้อบกพร่องด้านการออกแบบหรือข้อผิดพลาดในการใช้งานซึ่งทำให้ผู้โจมตีสามารถก่อให้เกิดอันตรายต่อผู้มีส่วนได้เสียของแอปพลิเคชัน ผู้มีส่วนได้เสีย ได้แก่ เจ้าของแอปพลิเคชัน ผู้ใช้แอปพลิเคชันและผู้เกี่ยวข้องอื่น ๆ ที่พึ่งพาแอปพลิเคชันความเสี่ยงของช่องโหว่ในระบบคอมพิวเตอร์ทั้งซอฟต์แวร์หรือฮาร์ดแวร์เป็นสิ่งที่มีผลกระทบต่อความมั่นคงปลอดภัยของระบบสารสนเทศในองค์กรหรือบริษัทการค้นพบช่องโหว่ใหม่มักจะนำไปสู่การสร้างโปรแกรมเจาะระบบ, ไวรัส, หรือมัลแวร์จากผู้บุกรุก

การจัดการความเสี่ยงเป็นกระบวนการสำคัญในการทำงานด้านความปลอดภัยของแต่ละองค์กร ซึ่งจะเริ่มต้นตั้งแต่การวางแผนจนไปถึงการรายงานผล เพื่อจัดการช่องโหว่ทั้งหมดที่เกิดขึ้นได้อย่างเป็นระบบ กระบวนการนี้จะต้องดำเนินการอย่างต่อเนื่องเพื่ออัพเดทและติดตามการเปลี่ยนแปลงใหม่ๆที่เกิดขึ้น ทั้งของฝั่งเครื่องมือที่ใช้ และฝั่งภัยคุกคามที่มีการพัฒนาขึ้น ซอฟต์แวร์ที่ใช้ในการจัดการช่องโหว่จะสามารถช่วยให้กระบวนการนี้เกิดขึ้นได้อย่างเป็นอัตโนมัติ โดยจะสามารถใช้เครื่องมือที่ทันสมัยเหล่านี้ในการสแกนหาช่องโหว่ของเครือข่าย รวมไปถึงในระบบ

ต่าง ๆที่มีส่วนเกี่ยวข้องกับองค์กรอีกด้วย สำหรับ “Vulnerability Management” จะสามารถแบ่งออกได้เป็น 4 ขั้นตอนดังนี้

1. ระบุช่องโหว่ หลักการสำคัญของทางออกในการจัดการช่องโหว่ที่เกิดขึ้นในองค์กร ก็คือการเริ่มต้นในการหาช่องโหว่ จากเครื่องมือที่ใช้ในการสแกนซึ่งจะประกอบด้วยกัน 3 ขั้นตอนดังนี้

–เครื่องมือสแกนเนอร์ทำการสแกนเข้าไปที่ระบบ

–ทำการเข้าถึงเครือข่ายด้วยการส่งข้อมูล TCP/UDP เพื่อระบุพอร์ตเปิดให้สแกนเนอร์เข้าไปทำการตรวจหาช่องโหว่

–รวบรวมข้อมูลจากระบบโดยละเอียดและเชื่อมโยงข้อมูลทั้งหมดเพื่อวางแผนแก้ไขช่องโหว่ที่เกิดขึ้น

การสแกนช่องโหว่นี้สามารถระบุความหลากหลายของระบบที่ใช้งานบนเครือข่าย โดยระบบจะตรวจสอบส่วนต่างๆ ภายในเครือข่ายและการใช้งาน ได้แก่ ระบบปฏิบัติการ, ซอฟต์แวร์ที่ติดตั้ง, บัญชีผู้ใช้, โครงสร้างระบบไฟล์, และการกำหนดค่าระบบและอื่นๆ โดยข้อมูลที่ได้ทั้งหมดจะสามารถเชื่อมโยงช่องโหว่ที่ระบบสแกนรู้จัก ก่อนจะทำการวิเคราะห์ช่องโหว่จากฐานข้อมูลเพื่อรายงานผลโดยการสแกนหาช่องโหว่มี 2 แบบคือ

-การค้นหาช่องโหว่โฮสต์จะค้นหาช่องโหว่ระดับระบบเช่นการอนุญาตไฟล์ที่ไม่ปลอดภัย,บั๊กของแอปพลิเคชันต้องใช้เครื่องมือพิเศษสำหรับระบบปฏิบัติการและซอฟต์แวร์ที่ใช้นอกเหนือจากการเข้าถึงระดับผู้ดูแลระบบแต่ละระบบที่ควรทดสอบการค้นหาช่องโหว่โฮสต์มักจะมีค่าใช้จ่ายสูงมากในระยะเวลาจึงใช้ในการประเมินระบบวิกฤติเท่านั้น เครื่องมือเช่น COPS และ Tiger เป็นที่นิยมในการค้นหาช่องโหว่โฮสต์

-ในการค้นหาช่องโหว่เครือข่ายใดเครือข่ายหนึ่งสำหรับค้นหาช่องโหว่ที่รู้จักจะหาตำแหน่งระบบทั้งหมดบนและกำหนดบริการของเครือข่ายที่ใช้งานอยู่จากนั้นวิเคราะห์บริการเหล่านั้นเพื่อหาช่องโหว่ที่อาจเกิดขึ้นกระบวนการนี้ไม่ต้องการการเปลี่ยนแปลงการกำหนดค่าใด ๆ บนระบบที่ถูกค้นหาช่องโหว่ [12]

2. การประเมินความเสี่ยงช่องโหว่

หลังจากตรวจพบช่องโหว่ และความเสี่ยงที่อาจจะเข้ามาคุกคามองค์กรแล้ว ขั้นตอนสำคัญต่อมาก็คือ “การประเมินความเสี่ยงที่เกิด” และจัดการอย่างเหมาะสม อีกทั้งยังต้องสอดคล้องกับกลยุทธ์การบริหารความเสี่ยงขององค์กร ซึ่งส่วนมากแต่ละองค์กรจะเลือกประเมินการแก้ปัญหาโดยการใช้ Common Vulnerability Scoring System (CVSS) เพื่อให้คะแนนการแก้ปัญหาของ Solution แต่ละแบบ ซึ่งจะส่งผลให้การแก้ปัญหาเป็นไปอย่างถูกจุดมากที่สุด

3. การจัดการแก้ไขปัญหาช่องโหว่ที่เกิดขึ้น ประกอบไปด้วย

-Remediation คือการแก้ไขที่จะช่วยจัดการช่องโหว่ที่เกิดขึ้นได้อย่างสมบูรณ์ เพื่อไม่ให้สามารถเข้ามาสร้างผลกระทบต่อธุรกิจของคุณได้ และเลือกที่เหมาะสมกับองค์กรที่ต้องการจัดการปัญหานี้ให้หมดไปอย่างจริงจัง

-Mitigation เป็นการลดโอกาสและผลกระทบของภัยคุกคามหรือช่องโหว่ที่เกิดขึ้น ในบางครั้งจะต้องแก้ไขด้วยการอัพเดทที่เหมาะสมกับการจัดการความเสี่ยงนี้ ตัวเลือกนี้เหมาะสมกับการแก้ไขปัญหาเบื้องต้นในระหว่างที่ทีมดูแลความปลอดภัยกำลังหาวิธีจัดการความเสี่ยงอย่างจริงจัง

-Acceptance เป็นการยอมรับปัญหาที่เกิดขึ้น โดยไม่แก้ไขหรือหาวิธีลดความเสี่ยงใด ๆ เพราะหากความเสี่ยงที่เกิดขึ้นไม่รุนแรง หรือไม่เหมาะสมกับค่าใช้จ่ายในการแก้ปัญหา วิธีการนี้ก็เหมาะสมมากเลยทีเดียว

4. การรายงานผลช่องโหว่

การประเมินผลของการแก้ไขปัญหาความเสี่ยงที่เกิดขึ้น พร้อมทั้งบันทึกรายงานผลเอาไว้จะสามารถช่วยให้การแก้ปัญหาครั้งต่อไปเกิดขึ้นได้อย่างรวดเร็วขึ้น หากเป็นช่องโหว่หรือความเสี่ยงที่เกิดขึ้นในลักษณะเดิมหรือใกล้เคียงกัน

2.4.2 รูปแบบของช่องโหว่

2.4.2.1 CVE

CVE (Common Vulnerabilities and Exposures) ซึ่งเป็นชื่อทางการของช่องโหว่โดยมีรูปแบบเป็น CVE-YYYY-NNNN โดยที่ YYYY เป็นปี ค.ศ. ที่ค้นพบช่องโหว่ ส่วน NNNN แสดงลำดับในการค้นพบดังนั้นจะมีชื่อได้ 10,000 ชื่อ เช่น CVE-2013–5576 แสดงว่าเป็นช่องโหว่ เกิดขึ้นปี 2013 และเป็นลำดับที่ 5576 ของปีนั้น (แต่ ตั้งแต่ 1 ม.ค. 2014 จะมีการเปลี่ยนแปลงรูปแบบ ตัวเลขตั้งแต่ 0–9999 จะยังใช้ NNNN หรือ 4 Digit เหมือนเดิมแต่เมื่อมากกว่านั้นก็สามารถขยายไปได้อีก เช่น CVE-0001 แต่เมื่อเกิน 10,000 ก็จะเป็น CVE-10001 หรือใช้ NNNNN เป็น 5 Digit) ซึ่ง CVE จะแสดงตัวเลขอ้างอิงสำหรับการตรวจสอบกับระบบรักษาความปลอดภัยต่าง ๆ ถ้ามีตัวเลข CVE แล้วแสดงว่าปัญหาดังกล่าวมีการยืนยันว่าเป็นปัญหา และสามารถแก้ไขปัญหาแล้ว โดยแต่ละ CVE จะมี CVSS Score, ความรุนแรงที่ไม่เท่ากันออกไปโดยแบ่งเป็น 2 เวอร์ชัน ตามตารางที่ 2.3

ตารางที่ 2.3 เปรียบเทียบความรุนแรงกับ CVSS SCORE ทั้ง 2 เวอร์ชัน

CVSS Score version 2		CVSS Score version 3	
Severity	Base Score Range	Severity	Base Score Range
		None	0.0
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

โดยแต่ละ CVE จะมี Vector ที่จะระบุรูปแบบของ CVE ในแต่ละ CVE ซึ่งจะมี 2 เวอร์ชันโดย CVSS Version 2 มี vector เช่น (AV:N/AC:L/Au:N/C:N/I:N/A:P) ประกอบไปด้วย Exploitability Metrics ซึ่งมีรายละเอียดแสดงดังตารางที่ 2.4 และ Impact Metrics ซึ่งมีรายละเอียดแสดงดังตารางที่ 2.5 และ CVSS Version 3 มี vector เช่น (AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H) ประกอบไปด้วย Exploitability Metrics ซึ่งมีรายละเอียดแสดงดังตารางที่ 2.6 และ Impact Metrics ซึ่งมีรายละเอียดแสดงดังตารางที่ 2.4 ถึงตารางที่ 2.7

ตารางที่ 2.4 รายละเอียด EXPLOITABILITY METRICS ของ CVSS Version 2

Exploitability Metrics	
Access Vector (AV) หมายถึงวิธีการเข้ามาใช้ประโยชน์จากช่องโหว่	Local (AV:L) แสดงถึงผู้โจมตีต้องทำการเข้าถึงทางกายภาพของระบบที่มีช่องโหว่
	Adjacent Network (AV:A) แสดงถึงผู้โจมตีต้องเข้าถึง broadcast หรือ collision domain ของระบบที่มีช่องโหว่ เช่น ARP spoofing , การโจมตีด้วยบลูทูธ
	Network (AV:N) แสดงถึงผู้โจมตีเข้าถึงการทำงานชั้นที่ 3 ของ OSI Model หรือสูงกว่านั้นโดยมักเรียกประเภทนี้ว่า Remotely Exploitable เช่น Remote buffer overflow ในเครือข่ายที่ให้บริการ

Access Complexity (AC) หมายถึงความยากง่ายที่จะเข้ามาใช้ประโยชน์ช่องโหว่ที่ค้นพบ	Low (AC:L) แสดงถึง ไม่มีเงื่อนไขพิเศษสำหรับการเข้ามาใช้งานช่องโหว่
	Medium (AC:M) แสดงถึงการมีเงื่อนไขพิเศษสำหรับการโจมตีเช่นการจำกัดการโจมตีหรือความต้องการสำหรับระบบที่มีช่องโหว่ที่ทำงานด้วยการตั้งค่าที่ไม่ใช่ค่าเริ่มต้น
	High (AC:H) แสดงถึงการมีเงื่อนไขพิเศษเช่นการทำ Social Engineering
Exploitability Metrics	
Authentication (Au) หมายถึงจำนวนครั้งที่ผู้ที่โจมตีต้องทำการยืนยันตัวตนเพื่อที่จะใช้ประโยชน์จากเป้าหมาย	Multiple (Au:M) แสดงถึงการที่ผู้โจมตีจะเข้ามาใช้งานช่องโหว่ต้องทำการยืนยันตัวตน 2 ครั้งหรือมากกว่านั้น
	Single (Au:S) แสดงถึงการที่ผู้โจมตีจะเข้ามาใช้งานช่องโหว่ต้องทำการยืนยันตัวตน 1 ครั้ง
	None (Au:N) แสดงถึงการการที่ผู้โจมตีจะเข้ามาใช้งานช่องโหว่ไม่ต้องทำการยืนยันตัวตน

ตารางที่ 2.5 รายละเอียด IMPACT Metrics ของ CVSS Version 2

Impact Metrics	
Confidentiality Impact (C) หมายถึงผลกระทบต่อความน่าเชื่อถือของข้อมูลที่ถูกประมวลผลโดยระบบ	None (C:N) หมายถึง ไม่มีผลกระทบต่อความน่าเชื่อถือของระบบ
	Partial (C:P) หมายถึง มีผลกระทบต่อความน่าเชื่อถือของระบบแต่ขอบเขตของการสูญเสียนั้นจำกัด
	Complete (C:C) หมายถึง มีผลกระทบต่อความน่าเชื่อถือของระบบและข้อมูลที่ถูกเปิดเผยมีผลกระทบโดยตรงและร้ายแรง

	Partial (I:P) แสดงถึงมีการแก้ไขบางส่วนของข้อมูลในระบบแต่ขอบเขตการปรับเปลี่ยนอยู่ในวงจำกัด
Impact Metrics	
Availability Impact (A) หมายถึงผลกระทบต่อความพร้อมใช้งานของระบบเป้าหมาย การโจมตีที่ Bandwidth ของเครือข่ายรอบของการประมวลผลหน่วยความจำหรือทรัพยากรอื่นๆ มีผลต่อความพร้อมใช้งานของระบบ	None (A:N) แสดงถึงไม่มีผลต่อความพร้อมในการทำงานของระบบ
	Partial (A:P) แสดงถึงมีการลดประสิทธิภาพหรือลดการทำงาน
	Complete (A:C) แสดงถึงมีความสูญเสียความพร้อมในการใช้งานของระบบที่ถูกโจมตี

ตารางที่ 2.6 รายละเอียด **EXPLOITABILITY METRICTS ของ CVSS Version 3**

Exploitability Metrics	
Attack Vector (AV) หมายถึงวิธีการเข้ามาใช้ประโยชน์จากช่องโหว่	Local (AV:L) แสดงถึงองค์ประกอบช่องโหว่ที่ไม่ใช่ขอบเขตของ Network Stack และเส้นทางของผู้โจมตีคือผ่านทางความสามารถในการเขียน/อ่าน/การดำเนินการ ในบางกรณีอาจพึ่งพาการโต้ตอบของผู้ใช้งานเพื่อดำเนินการไฟล์ที่เป็นอันตราย
	Adjacent Network (AV:A) แสดงถึงผู้โจมตีต้องเข้าถึง broadcast หรือ collision domain ของระบบที่มีช่องโหว่ เช่น ARP spoofing
	Network (AV:N) แสดงถึงผู้โจมตีเข้าถึงการทำงานชั้นที่ 3 ของ OSI Model หรือสูงกว่านั้นโดยมักเรียกประเภทนี้ว่า Remotely Exploitable
	Physical (AV:P) แสดงถึง การใช้ประโยชน์จากช่องโหว่โดยการเข้าถึงทางกายภาพโดยผู้โจมตีจะทำการสัมผัสหรือจัดการองค์ประกอบช่องโหว่ เช่น การแนบอุปกรณ์ต่อพ่วงเข้าสู่ระบบ

Exploitability Metrics	
Attack Complexity (AC) หมายถึง ระดับความซับซ้อนที่ผู้โจมตีสามารถมาใช้ประโยชน์จากช่องโหว่	High (AC:H) แสดงถึง การเข้ามาใช้ประโยชน์สำเร็จขึ้นอยู่กับเงื่อนไขที่เหนือกว่าการควบคุมของผู้โจมตีซึ่งการโจมตีที่สำเร็จผู้โจมตีต้องมีการลงทุนและเตรียมตัวในการดำเนินการ
	Low (AC:L) แสดงถึง ไม่มีเงื่อนไขพิเศษสำหรับการเข้ามาใช้งานช่องโหว่
Privileges Required (PR) หมายถึงการยืนยันตัวตนก่อนที่ผู้โจมตีจะเข้ามา	High (PR:H) แสดงถึงมีการยืนยันตัวตนในระดับผู้จัดการระบบ
	Low (PR:L) แสดงถึงมีการยืนยันตัวตนระดับผู้ใช้งาน
	None (PR:N) แสดงถึงไม่มีการยืนยันตัวตนของผู้โจมตี
User Interaction (UI) หมายถึงผู้โจมตีต้องการบางอย่างจากการที่ผู้ใช้งานมีการโต้ตอบ	None (UI:N) แสดงถึงผู้โจมตีสามารถโจมตีได้โดยไม่ต้องการการโต้ตอบจากผู้ใช้งาน
	Required (UI:R) แสดงถึงผู้โจมตีสามารถโจมตีได้โดยต้องการการโต้ตอบจากผู้ใช้งานเพื่อที่จะให้การโจมตีสำเร็จ
Scope (S) หมายถึงขอบเขตที่ผู้โจมตีทำให้มีผลต่อองค์ประกอบที่เกี่ยวข้องกับสิทธิ์	Unchanged (S:U) แสดงถึงการใช้ประโยชน์จากช่องโหว่สามารถมีผลต่อการจัดการด้วยสิทธิ์เดียวกันในกรณีนี้องค์ประกอบช่องโหว่และองค์ประกอบการโจมตีเป็นสิ่งๆ เดียวกัน
	Changed (S:C) แสดงถึงการใช้ประโยชน์จากช่องโหว่สามารถมีผลที่เหนือกว่าการยืนยันสิทธิ์ในกรณีนี้องค์ประกอบช่องโหว่และองค์ประกอบการโจมตีเป็นคนละสิ่งกัน

ตารางที่ 2.7 รายละเอียด IMPACT METRICS ของ CVSS Version 3

Impact Metrics	
Confidentiality Impact (C) หมายถึงการโจมตีที่มีผลกระทบต่อความน่าเชื่อถือของข้อมูลที่ถูกประมวลผลโดยระบบ	None (C:N) แสดงถึง ไม่มีผลกระทบต่อความน่าเชื่อถือของระบบ
	Low (C:L) แสดงถึง มีบางข้อมูลผลกระทบต่อความน่าเชื่อถือของระบบแต่ขอบเขตของการสูญเสียนั้นจำกัด
	High (C:H) แสดงถึง ทุกข้อมูลมีผลกระทบต่อความน่าเชื่อถือของระบบและเมื่อข้อมูลถูกเปิดเผยมีผลกระทบโดยตรงและร้ายแรง
Integrity Impact (I) หมายถึงผลกระทบต่อความมั่นคงของระบบที่ถูกโจมตี	None (I:N) แสดงถึง ไม่กระทบถึงความมั่นคงของระบบ
	Low (I:L) แสดงถึงข้อมูลบางส่วนโดนแก้ไขแต่ผู้โจมตีแต่ผู้โจมตีไม่สามารถใช้ประโยชน์จากการที่ข้อมูลบางส่วนถูกแก้ไข
	High (I:H) แสดงถึงมีผลการทบทั้งหมดต่อความมั่นคงของระบบหรือการป้องกันได้สูญหายไปอย่างสิ้นเชิง
Availability Impact (A) หมายถึงผลกระทบต่อความพร้อมในการทำงานของระบบที่ถูกโจมตี	None (A:N) แสดงถึงไม่มีผลต่อความพร้อมในการทำงานของระบบ
	Low (A:L) แสดงถึงมีการลดประสิทธิภาพการทำงานหรือขัดขวางการตอบสนองในการทำงาน
	High (A:H) แสดงถึงมีความสูญเสียความพร้อมในการใช้งานของระบบที่ถูกโจมตี

โดยเมื่อทราบ Vector เวอร์ชัน 2 ของแต่ละ CVE แล้วสามารถคำนวณหาค่า CVSS Score ได้โดยที่ CVSS Score เวอร์ชัน 2 มีวิธีการคำนวณดังสมการที่ 2.1, 2.2, 2.3 และ 2.4 โดยที่สามารถดูค่าของตัวแปรได้ดังตารางที่ 2.8

$$CVSS\ Score\ = [(0.6 \times Impact) + (0.4 \times Exploit) - 1.5] \times f(Impact) \quad (2.1)$$

$$Impact = 10.41\ \times [1 - (1 - ConfImpact)] \times (1 - IntegImpact) \times (1 - AvailImpact) \quad (2.2)$$

$$Exploit = 20\ \times AcceessComplexity \times Authentication \times AccessVector \quad (2.3)$$

$$f(Impact)\ = \begin{cases} 0 & if\ Impact = 0 \\ 1.176 & otherwise \end{cases} \quad (2.4)$$

ตารางที่ 2.8 ค่าของตัวแปรที่ใช้ในการคำนวณ CVSS Score เวอร์ชั่น 2

Variable (Metric)	Case	Value
AccessComplexity (Access Complexity)	High	0.35
	Medium	0.61
	Low	0.71
Authentication (Authentication)	None	0.704
	Single	0.56
	Multiple	0.45
AccessVector (Access Vector)	Local	0.395
	Adjacent Network	0.646
	Network	1
ConfImpact (Confidentiality Impact)	None	0
	Partial	0.275
	Complete	0.66
Variable (Metric)	Case	Value
IntegImpact (Integrity Impact)	None	0
	Partial	0.275
	Complete	0.66
AvailImpact	None	0

(Availability Impact)	Partial	0.275
	Complete	0.66

และเมื่อทราบ Vector เวอร์ชั่น 3 ของแต่ละ CVE แล้วสามารถคำนวณหาค่า CVSS Score เวอร์ชั่น 3 มีวิธีการคำนวณดังสมการที่ 2.5 , 2.6 , 2.7 และ 2.8 โดยที่สามารถดูค่าของตัวแปรได้ดังตารางที่ 2.9

$$CVSS\ Score = \begin{cases} 0 & if\ Impact \leq 0 \\ \left\lceil \min[(Impact + Exploit), 10] \right\rceil & if\ Scope\ Unchanged \\ \left\lceil \min[1.08(Impact + Exploit), 10] \right\rceil & if\ Scope\ Change \end{cases} \quad (2.5)$$

$$Impact = \begin{cases} 6.42 \times (ISC) & if\ Scope\ Unchanged \\ [7.52 \times (ISC - 0.029)] - [3.25 \times (ISC - 0.02)^{15}] & otherwise \end{cases} \quad (2.6)$$

$$ISC = 1 - [(1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImapct)] \quad (2.7)$$

$$Exploit = 8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction \quad (2.8)$$

ตารางที่ 2.9 ค่าของตัวแปรที่ใช้ในการคำนวณ CVSS Score เวอร์ชัน 3

Variable (Metric)	Case	Value
AttackVector (Attack Vector)	Network	0.82
	Adjacent	0.62
	Local	0.55
	Physical	0.2
AttackComplexity (Attack Complexity)	Low	0.77
	High	0.44

PrivilegeRequired (Privilege Required)	None	0.85
	Low	0.62
	High	0.27
UserInteraction (User Interaction)	None	0.85
	Required	0.62
ConfImpact (Confidentiality Impact)	None	0
	Low	0.22
	High	0.56
IntegImpact (Integrity Impact)	None	0
	Low	0.22
	High	0.56
AvailImpact (Availability Impact)	None	0
	Low	0.22
	High	0.56

2.4.2.2 OWASP TOP 10

OWASP TOP 10 เป็นช่องโหว่ที่บอกความเสี่ยงด้านความปลอดภัยของเว็บแอปพลิเคชัน 10 อันดับแรกที่กำหนดโดย Open Web Application Security Project (OWASP) ซึ่งเป็นองค์การไม่แสวงหาผลกำไรในประเทศสหรัฐอเมริกาที่มีจุดประสงค์เป็นศูนย์รวมในการร่วมมือจากนักพัฒนาเว็บแอปพลิเคชันทั่วโลก โดย OWASP TOP 10 ในปี 2017 ประกอบไปด้วยช่องโหว่ที่มีความร้ายแรงและพบได้บ่อยเรียงตามลำดับได้ดังต่อไปนี้ [17]

1) Injection ยกตัวอย่างเช่น SQL, NoSQL, OS และ LDAP injection เกิดขึ้นเมื่อข้อมูลที่ไม่น่าเชื่อถือถูกส่งเข้ามาในส่วนของคำสั่งหรือ query ผู้โจมตีที่ไม่ประสงค์ดีต่อข้อมูลสามารถใช้งานเพื่อดำเนินการคำสั่ง หรือเข้าถึงข้อมูลโดยปราศจากการได้รับอนุญาต

2) Broken Authentication คือ ช่องโหว่เกิดขึ้นจากการทำงานระบบการพิสูจน์ตัวตน (Authentication) ปลอดภัยไม่เพียงพอ เช่น การตั้งชื่อผู้ใช้งานและรหัสผ่านที่ง่ายต่อการคาดเดาหรือการจัดการสิทธิในการเข้าถึงที่ไม่มีประสิทธิภาพมากพอ (Session Management) ทำให้ผู้ไม่ประสงค์ดีสามารถเข้าสู่ระบบได้สำเร็จ

3) Sensitive Data Exposure คือ ช่องโหว่ที่เกิดขึ้นจากระบบขาดการป้องกันเพิ่มเติมเช่นเข้ารหัสลับข้อมูลสำคัญระหว่างการรับส่งข้อมูลหรือการเข้ารหัสด้วยวิธีการที่สามารถถอดรหัสได้ง่าย ทำให้ผู้ไม่ประสงค์ดีสามารถเข้าถึงข้อมูลสำคัญในระบบได้

4) XML External Entities (XXE) คือ การใช้งานภาษา XML เวอร์ชันเก่าหรือการตั้งค่าที่ไม่เหมาะสมทำให้ในการรับข้อมูลจาก Entity ภายนอกผ่านมาตรฐานการอ้างอิงรูปแบบ URI ทำให้เกิดช่องโหว่ได้หลากหลายรูปแบบ เช่น ไฟล์ข้อมูลภายในถูกเปิดเผย การค้นหาพอร์ตภายใน การดำเนินการคำสั่งแปลกปลอมจากระยะไกล (Remote code Execution) หรือการทำ DoS (Denial of Service) เป็นต้น

5) Broken Access Control คือ ช่องโหว่ที่เกิดขึ้นจากระบบขาดการตรวจสอบฟังก์ชัน หรือสิทธิ์ต่าง ๆ อย่างเหมาะสม ทำให้การเชื่อมต่อเข้าสู่ระบบ ผู้ไม่ประสงค์ดีสามารถทำการ เชื่อมต่อเข้าสู่ระบบ โดยการข้ามขั้นตอนการพิสูจน์ตัวตนได้สำเร็จ

6) Security Misconfiguration คือ ช่องโหว่ที่เกิดขึ้นจากระบบมีตั้งค่าที่ไม่เหมาะสม เช่น

การใช้ชื่อผู้ใช้งานและรหัสผ่านแบบค่าเริ่มต้น (Default) การตั้งค่าเส้นทาง (Path) การเข้าถึงแบบค่าเริ่มต้น การคงอยู่ของไฟล์สำคัญในระบบที่มาพร้อมกับการติดตั้งและตั้งค่า เป็นต้น ทำให้ผู้ไม่ประสงค์ดีสามารถเข้าถึงข้อมูลภายในระบบหรือรวบรวมข้อมูลสำคัญภายในระบบได้

7) Cross-Site Scripting คือ ช่องโหว่ที่ผู้ไม่ประสงค์ดีสามารถแทรกหรือฝังคำสั่งอันตรายเข้าสู่ระบบเว็บแอปพลิเคชันได้ เช่น คำสั่งของ JavaScript หรือ HTML ทำให้ผู้ไม่ประสงค์ดีสามารถขโมย session หรือ เปลี่ยนหน้าเว็บไซต์ไปยังเว็บไซต์ของผู้ไม่ประสงค์ดีได้สำเร็จ

8) Increase Deserialization คือ การใช้งานฟังก์ชัน Deserialization ที่อนุญาตให้ผู้ไม่ประสงค์ดีสามารถแก้ไขโครงสร้างข้อมูลจากระยะไกล ทำให้เกิดช่องโหว่ได้หลากหลายรูปแบบ เช่น Replay Attacks, Injection Attacks และ Privilege Escalation Attacks เป็นต้น

9) Using Components with Know Vulnerability คือ ช่องโหว่ที่เกิดขึ้นจากตัวระบบหรือองค์ประกอบต่าง ๆ ที่มีการใช้งานเช่นที่ libraries, frameworks หรือ software modules ต่าง ๆ ที่มีช่องโหว่ทำให้ผู้ไม่ประสงค์ดีสามารถเข้าถึงข้อมูลสำคัญหรือยึดครองเครื่องเซิร์ฟเวอร์ได้สำเร็จ

10) Insufficient Logging & Monitoring คือ การเฝ้าระวัง Log ที่ไม่เพียงพอต่อการตอบสนองต่อภัยคุกคามจากผู้ไม่ประสงค์ดีทำให้สามารถขยายผลการโจมตีและเข้าสู่ระบบได้สำเร็จ

2.5 ภาษาไพธอน (Python)

ในการออกแบบโปรแกรมเพื่อใช้ในการรับคำสั่งจากผู้ใช้งานและนำคำสั่งที่ได้มาประมวลผล หาช่องโหว่และรวบรวมข้อมูลมีการออกแบบโดยใช้ภาษาไพธอน

2.5.1) ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยกรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA

2.5.2) Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับระบบปฏิบัติการ กล่าวคือสามารถใช้ Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD ภาษาไพธอนเป็น OpenSource ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้โดยไม่ต้องเสียค่าใช้จ่าย และทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครบกับทุกลักษณะงาน

2.6 ภาษาพีเอชพี (PHP)

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษาซี ภาษาจาวา และภาษาเพิร์ล ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีการตอบโต้ได้อย่างรวดเร็ว

2.7 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่น เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องที่ให้บริการเว็บ (Web Server) โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูล Open Source ที่ถูกนำไปใช้งานมาก MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ เป็นการเก็บข้อมูลในรูปของตาราง ในแต่ละตารางแบ่งออกเป็นแถวๆ และในแต่ละแถวจะแบ่งเป็นคอลัมน์

2.8 phpMyAdmin

2.8.1) phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล Mysql แทนการคีย์คำสั่ง เนื่องจากถ้าเราจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัว DBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการนั้นเอง

2.8.2) phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษา PHP ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง TABLE ใหม่ๆ และยังมี function ที่ใช้สำหรับการทดสอบการ query ข้อมูลด้วยภาษา SQL พร้อมกันนั้น ยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกับกันการใช้ภาษา SQL ในการสร้างตารางข้อมูล

2.8.3) phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่าน Web Browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web Sever เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server

2.9 Apache2 Webserver

Apache2 หรือ Apache2 Webserver เป็นซอฟต์แวร์เซิร์ฟเวอร์ที่ใช้กันอย่างแพร่หลาย Apache พัฒนาและดูแลโดย Apache Software Foundation ซึ่งเป็นซอฟต์แวร์ที่สามารถใช้งานได้ฟรี โดยมีการใช้โดยรวมประมาณร้อยละ 67 ของเว็บเซิร์ฟเวอร์ทั้งหมดในโลก ซึ่งรวดเร็วเชื่อถือได้และปลอดภัย สามารถปรับแต่งได้เพื่อตอบสนองความต้องการของสภาพแวดล้อมที่หลากหลาย โดยสามารถเพิ่ม function พิเศษที่เป็น module plugin ได้โดยง่าย

2.10 เอชทีเอ็มแอล (HTML)

เอชที่เอ็มแอล (HTML : Hypertext Markup Language) คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิ้งค์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่าง ๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

2.11 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B เป็น Single Board คอมพิวเตอร์จาก Raspberry Pi Foundation ใช้ชิพการประมวลผล Broadcom BCM2711 Quad-Core ARM Cortex-A72 ความเร็ว 1.5 GHz มีหน่วยความจำสูงสุด LPDDR4-2400 ขนาด 4 GB มาพร้อมชิพ Wireless LAN

แบบ Dual-Band รองรับ 2.4 GHz และ 5 GHz พร้อมรองรับ Bluetooth 5.0 BLE มีพอร์ต LAN รองรับ Gigabit Ethernet พอร์ต USB 3.0 Host Type A จำนวน 2 พอร์ต และ USB 2.0 Host Type A จำนวน 2 พอร์ต มีพอร์ต micro-HDMI จำนวน 2 พอร์ต รองรับการเชื่อมต่อจอความละเอียด 4K 60 fps ดังรูปที่ 2.8

รูปที่ 2.21 RASPBERRY PI 4 MODEL B

2.12 Line Official Account

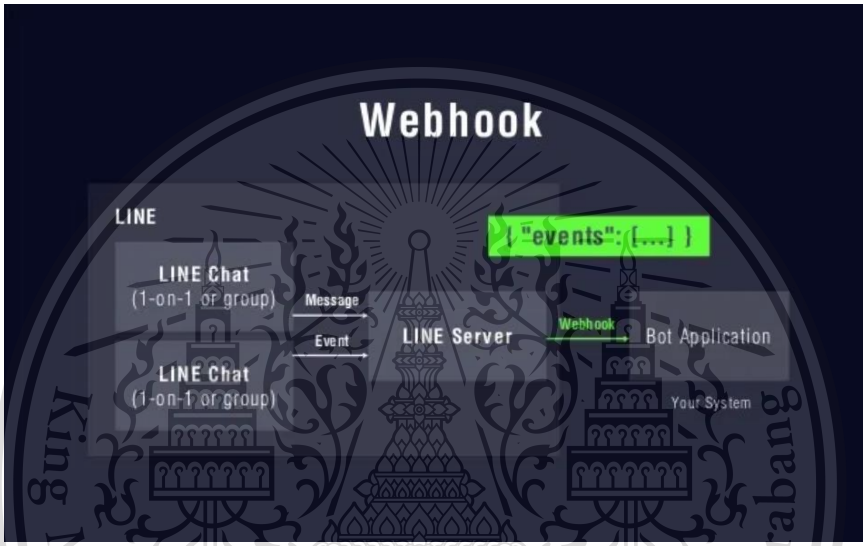
บัญชีทางการของ LINE สำหรับธุรกิจที่ช่วยให้ร้านค้าสามารถสร้างฐานผู้ติดตามสื่อสารและส่งข้อมูลกิจกรรมทางการขายและการตลาด หรือโปรโมชั่นพิเศษไปยัง ลูกค้าผ่านทางไลน์ ตอบโจทย์ธุรกิจด้วยฟีเจอร์ที่หลากหลายที่จะช่วยสร้าง ประสบการณ์ที่ดีให้แก่ลูกค้าของร้านค้า รวมทั้งช่วยให้ร้านค้าสามารถบริหาร จัดการการขายได้อย่างมีประสิทธิภาพ เช่น การสร้างข้อความทักทาย ข้อความ ตอบกลับอัตโนมัติ คูปองและบัตรสะสมแต้ม การแชทแบบ 1-1 การบรอดแคสต์หาผู้ติดตามทั้งหมด หรือการบรอดแคสต์แบบระบุกลุ่มเป้าหมาย เป็นต้น ศึกษาฟีเจอร์เพิ่มเติม

2.13 Web Application

เว็บแอปพลิเคชัน เป็นโปรแกรมคอมพิวเตอร์หนึ่งที่ทำหน้าที่เฉพาะ โดยใช้ Web Browser เช่น Google chrome , Firefox เป็น Client ซึ่ง Client นี้เป็นระบบหรือแอปพลิเคชัน ที่สามารถเชื่อมต่อเข้ากับระบบคอมพิวเตอร์อื่นที่เรียกว่าเซิร์ฟเวอร์ได้ ในปัจจุบันจึงมีการพัฒนาเว็บแอปพลิเคชันควบคู่ไปกับการพัฒนาโปรแกรม เพื่อสอดคล้องต่อการใช้งานในปัจจุบัน นอกจากนี้เว็บแอปพลิเคชันยังช่วยลดความรับผิดชอบของนักพัฒนาในการสร้างClient สำหรับคอมพิวเตอร์หรือระบบปฏิบัติการประเภทใดประเภทหนึ่ง เพื่อให้ทุกคนสามารถใช้แอปพลิเคชันได้ เว็บแอปพลิเคชันมักใช้สคริปต์ทำงานบนฝั่งเซิร์ฟเวอร์ เช่น PHP เป็นต้น และสคริปต์ฝั่ง Client เช่น HTML เป็นต้น เพื่อพัฒนาแอปพลิเคชัน ซึ่งทั้ง 2 ฝั่งจะทำหน้าที่แตกต่างกัน อย่างสคริปต์ฝั่ง Client จะทำหน้าที่จัดการกับการนำเสนอข้อมูล ในขณะที่สคริปต์ฝั่งเซิร์ฟเวอร์จะจัดการกับพวกการจัดเก็บข้อมูล

2.14 Web hook

Webhook คือ event ต่างๆที่เกิดขึ้นกับ LINE Bot(Event trigger) โดยเมื่อ event เกิดขึ้นแล้วจะมีสัญญาณพร้อมกับข้อมูลในรูปแบบที่เป็น JSON วิ่งมาที่ Webhook API ที่เราผูกไว้ใน LINE Developers Console และ LINE Bot สามารถรับ Webhook Events ได้ทั้งหมด 15 สัญญาณ



รูปที่ 2.22 หลักการทำงานของ Webhook

บทที่ 3

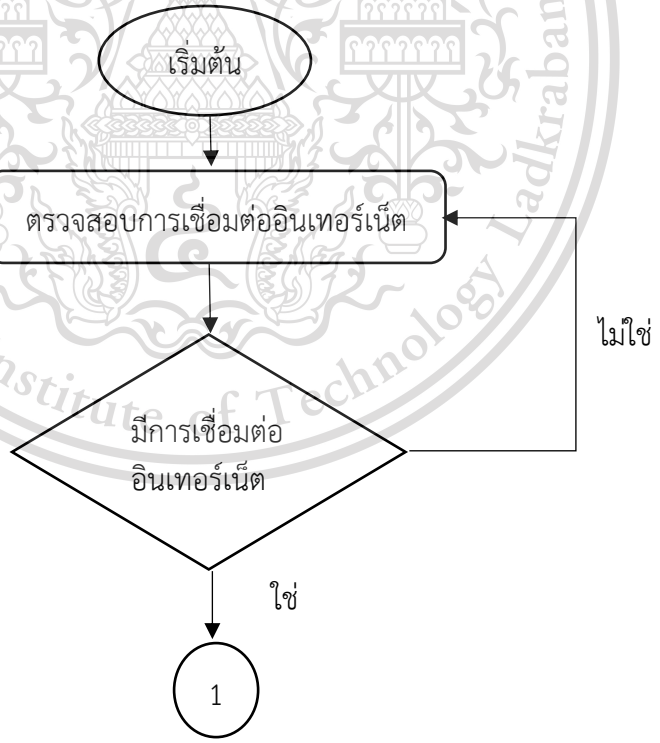
การออกแบบและการจัดทำปริญญานิพนธ์

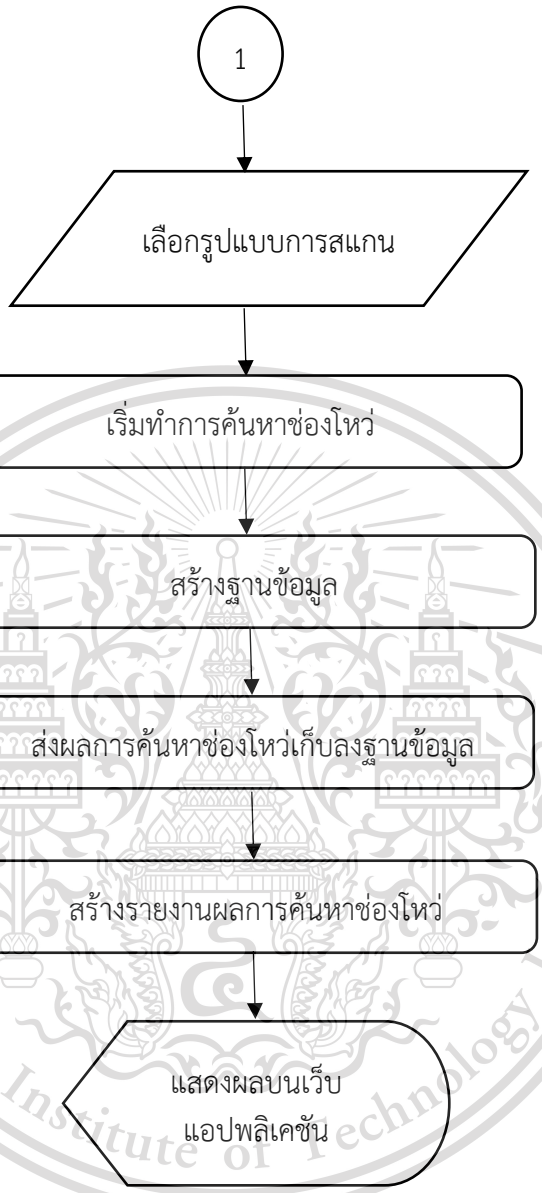
ในการจัดทำปริญญานิพนธ์มีวิธีการดังนี้ คือ ทำการทดลองหาช่องโหว่โดยใช้อุปกรณ์ค้นหาช่องโหว่บนระบบปฏิบัติการ Rasbian จากนั้นทำการสร้างระบบเพื่อให้สะดวกในการใช้งานเครื่องมือค้นหาช่องโหว่ตามที่ได้ศึกษามาและนำโปรแกรมไปทดลองใช้งานจริงบนอุปกรณ์ Raspberry Pi

3.1 การออกแบบ

3.1.1 การออกแบบระบบตรวจจับช่องโหว่

ในการออกแบบระบบตรวจสอบช่องโหว่เพื่อให้ผู้ใช้งานสามารถค้นหาช่องโหว่ที่อยู่ภายในเครือข่ายของตนเองได้และสามารถรายงานผลการค้นหาผ่านทางเว็บแอปพลิเคชันโดยใช้อุปกรณ์ Raspberry Pi มีแผนภาพการทำงานดังรูปที่ 3.1





รูปที่ 3.1 แผนภาพการทำงานของระบบตรวจจับช่องโหว่

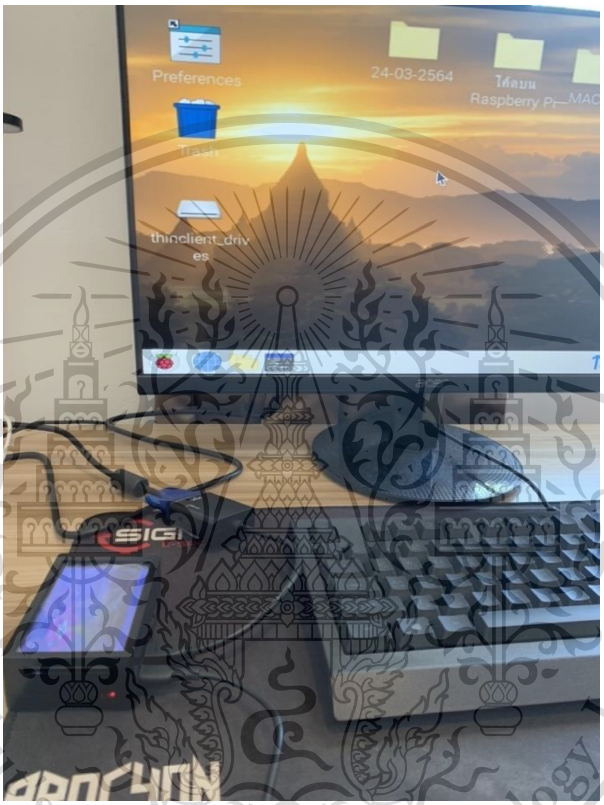
3.1.2 การออกแบบอุปกรณ์ Raspberry Pi เพื่อเตรียมพร้อมการทำงาน

ในการออกแบบอุปกรณ์ Raspberry Pi โดยที่อุปกรณ์ Raspberry Pi ที่นำมาใช้งานเป็น Raspberry Pi 4 Model B มีรายละเอียดดังนี้ RAM 2 GB ROM 16 GB รองรับ

Wireless LAN และ Gigabit Ethernet ใช้งานร่วมกับหน้าจอคอมพิวเตอร์ เพื่อให้การใช้งานอุปกรณ์สะดวกมากยิ่งขึ้นโดยทำตามขั้นตอนดังนี้

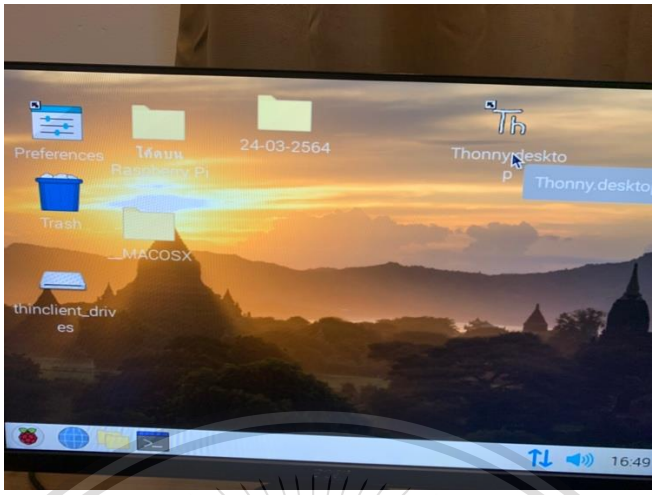
1. ขั้นตอนแรกทำการลงระบบปฏิบัติการ raspbian ให้กับ Raspberry Pi หลังจากนั้นนทำการดาวน์โหลด Driver จากเว็บ https://www.raspberrypi.org/software/

2. ทำการเชื่อมต่อหน้าจอคอมพิวเตอร์เข้ากับ Raspberry Pi ดังรูปที่ 3.2



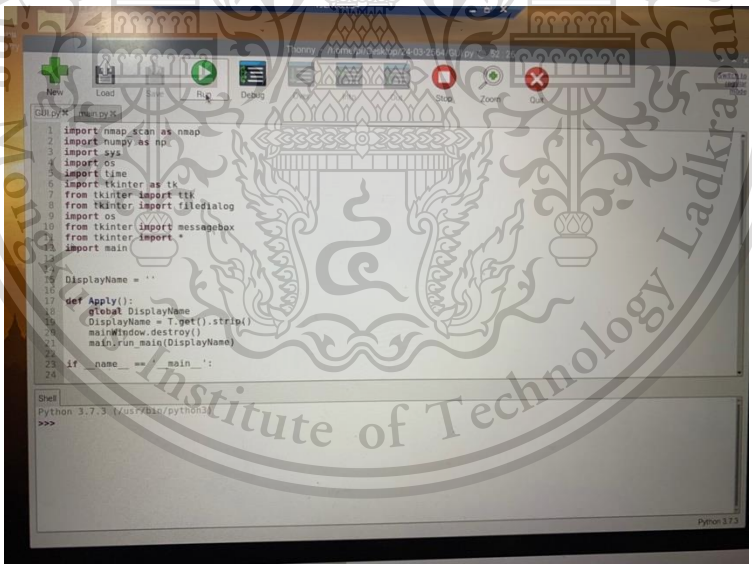
รูปที่ 3.2 การเชื่อมต่อจอคอมพิวเตอร์ กับ Raspberry Pi

3. หลังจากเปิดเครื่อง Raspberry Pi ให้เราไปที่ไฟล์ Thonnydesktop บนหน้า Desktop ดังรูปที่ 3.3



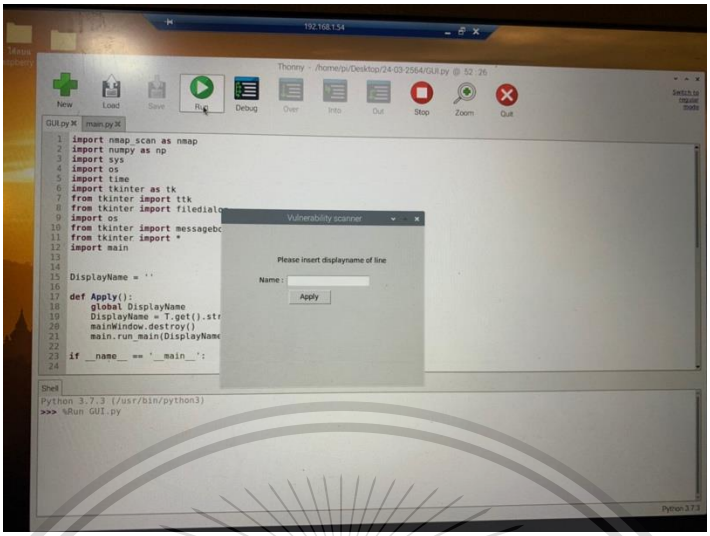
รูปที่ 3.3 ดับเบิ้ลคลิกที่ Thonnydesktop เพื่อทำการเข้าใช้งาน

4. ทำการรันโค้ดจากไฟล์ GUI.py เพื่อเริ่มการทำงานของโปรแกรมตรวจสอบช่องโหว่ตามรูปที่ 3.4



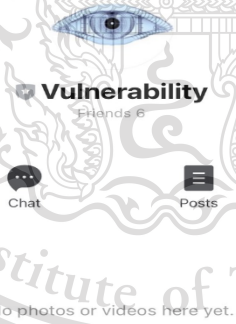
รูปที่ 3.4 กดรันโค้ดจากไฟล์ GUI.py

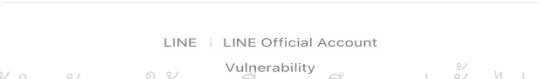
5. หลังจากทำการกดรัน จะปรากฏหน้าต่างให้เราใส่ชื่อ Display Name ของไลน์ผู้ใช้งานเพื่อรับการแจ้งเตือน ตามรูปที่ 3.5



รูปที่ 3.5 หน้าต่างให้เราใส่ชื่อ Display Name

6. หลังจากผู้ใช้งานได้ใส่ชื่อ Display name ของไลน์ผู้ใช้งานเพื่อรับการแจ้งเตือน สถานะสแกนโดยผู้ใช้งาน สามารถแอดได้ผ่าน QR Code line ซึ่งเปิดใช้งานเป็น Line OA ในรูป 3.6 และ 3.7

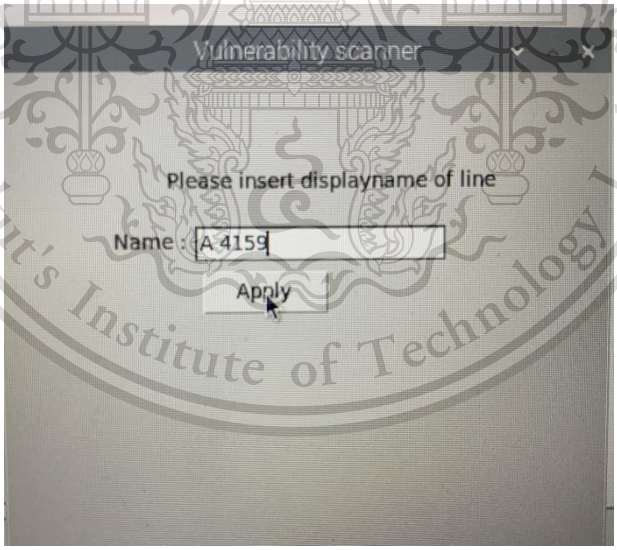




รูปที่ 3.6 Line OA ที่ทำการเพิ่มเพื่อนผ่านแอปพลิเคชั่นไลน์มีชื่อว่า Vulnerability

รูปที่ 3.7 QR Code ของ ไลน์ OA Vulnerability

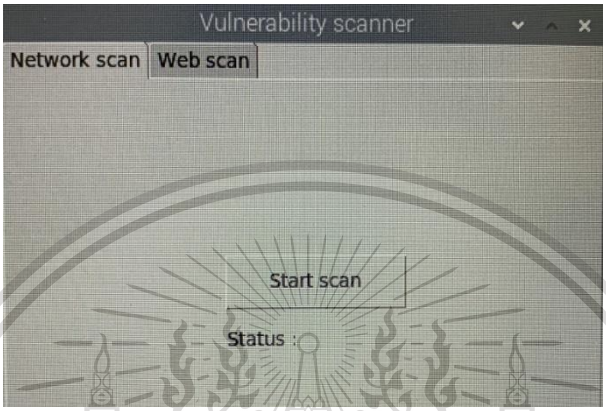
7.หลังจากใส่ชื่อ Display Name ของไลน์แล้วกด Apply เพื่อเริ่มต้นสแกน ตามรูป 3.8



รูปที่ 3.8 กรอกชื่อ display name Line เพื่อรับการแจ้งเตือน

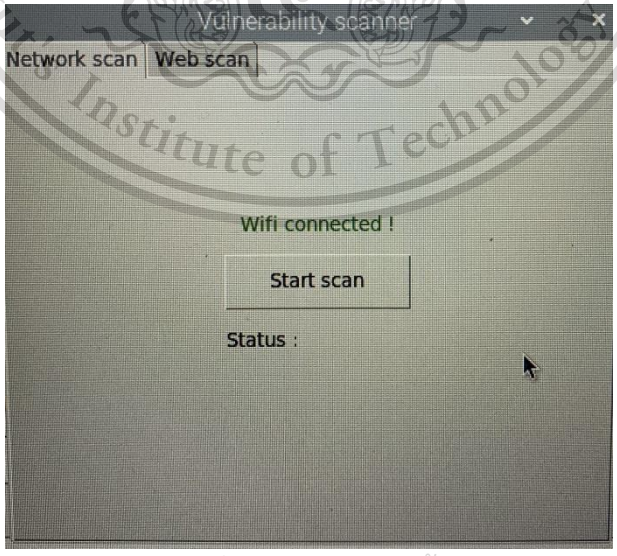
3.1.3 การออกแบบหน้า User GUI

ในการออกแบบหน้า User GUI ที่ทำงานอยู่บน Raspberry Pi โดยให้ผู้ใช้งานเช็คดูว่า Raspberry pi มีการเชื่อมต่ออินเทอร์เน็ตหรือไม่ โดยสามารถเลือกได้ว่าจะทำการสแกน Network หรือ Web ตามรูปที่ 3.9



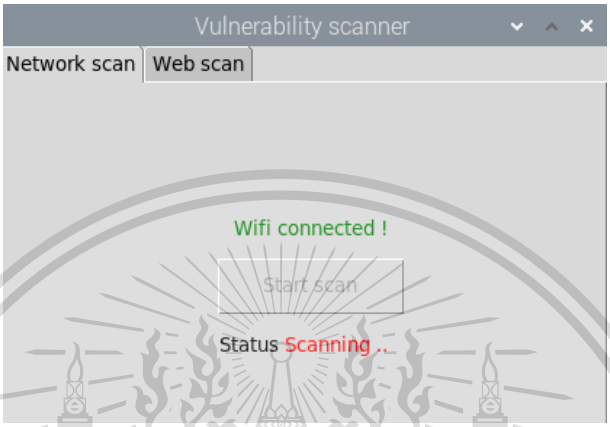
รูปที่ 3.9 หน้า User GUI เริ่มต้นโดยสามารถเลือกสแกนระหว่าง Network หรือ Web

ในส่วนของ ฟังก์ชั่น Network Scan ถ้ามีการเชื่อมต่อเครือข่ายอินเทอร์เน็ตแล้วโปรแกรมสแกนช่องโหว่ จะได้รับ IP Address โดยอัตโนมัติ โดยจะขึ้นสถานะ Ready ว่าพร้อมสำหรับการสแกน ดังรูปที่ 3.10

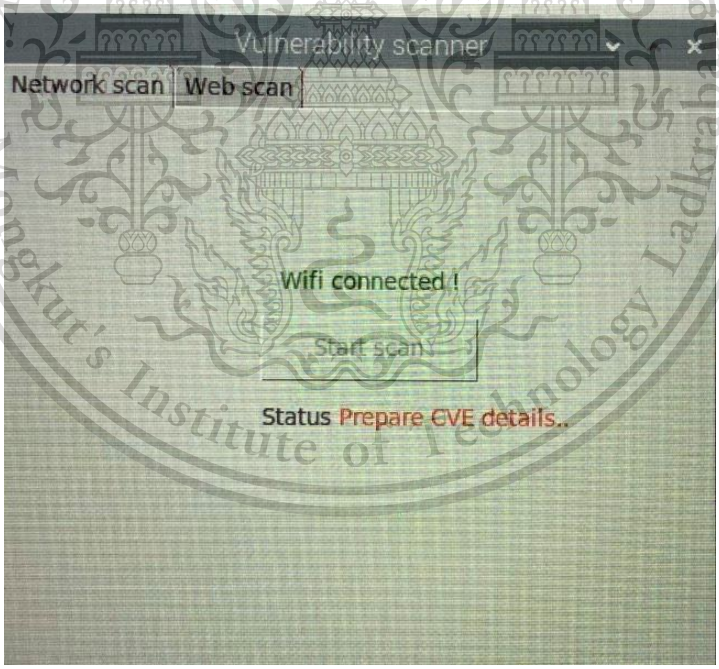


รูปที่ 3.10 หน้า User GUI เมื่อได้รับ IP Address โดยอัตโนมัติ จะขึ้นสถานะ Ready

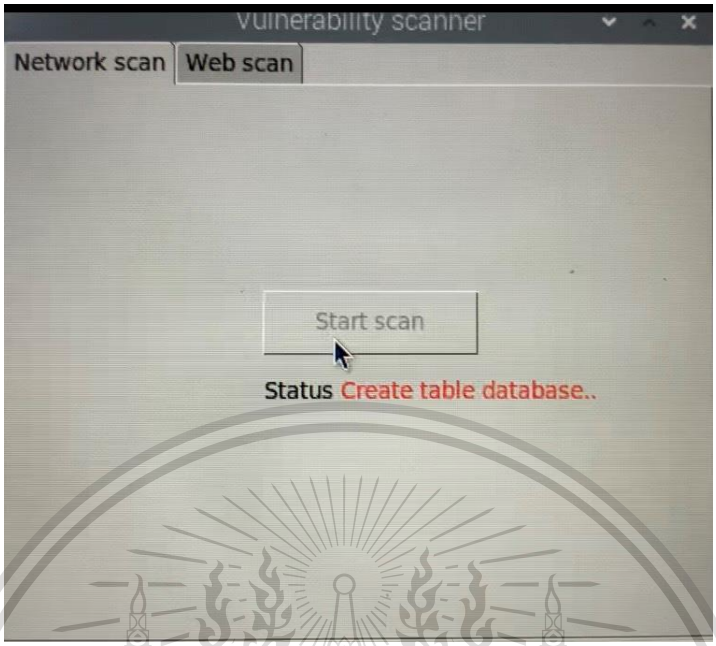
หลังจากกดปุ่มเพื่อเริ่มการสแกน จะปรากฏสถานะการทำงานต่างๆประกอบไปด้วย Scanning , Insert Data to Database , Create Table Database และ Finished เป็นอันเสร็จสิ้นการสแกน ดังรูปที่ 3.11 , 3.12 , 3.13 และ 3.14



รูปที่ 3.11 แสดงสถานะ Scanning



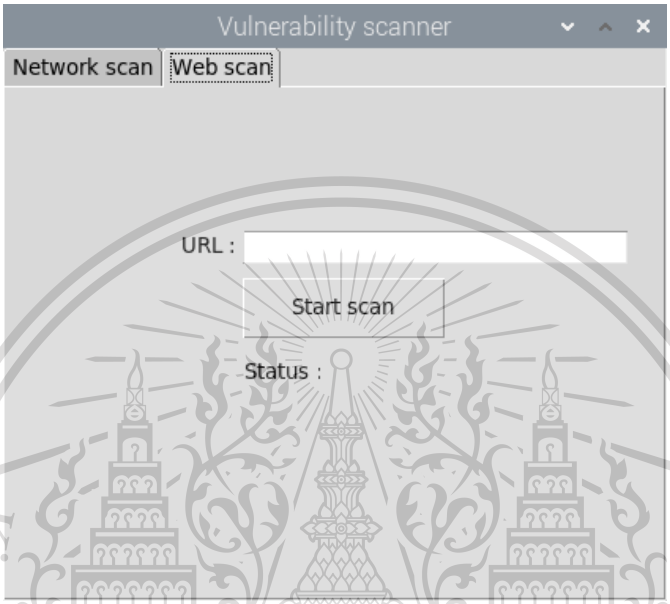
รูปที่ 3.12 แสดงสถานะ Prepare CVE detalis



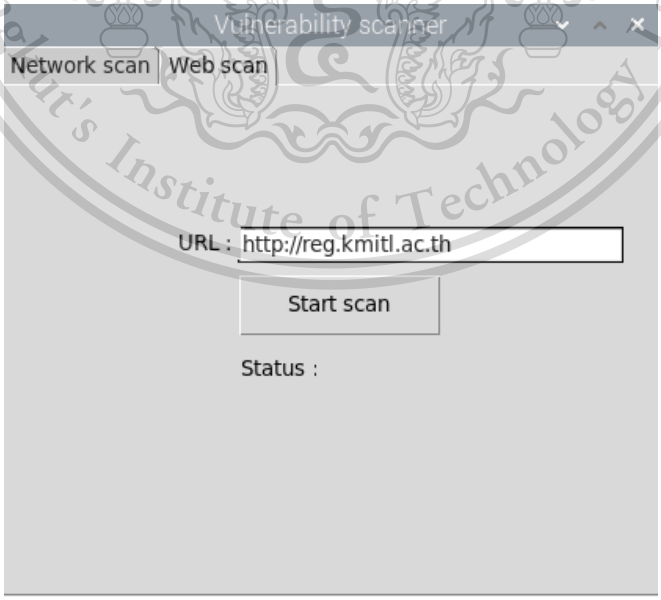
รูปที่ 3.13 แสดงสถานะ Create Table Database

รูปที่ 3.14 แสดงสถานะ Finished

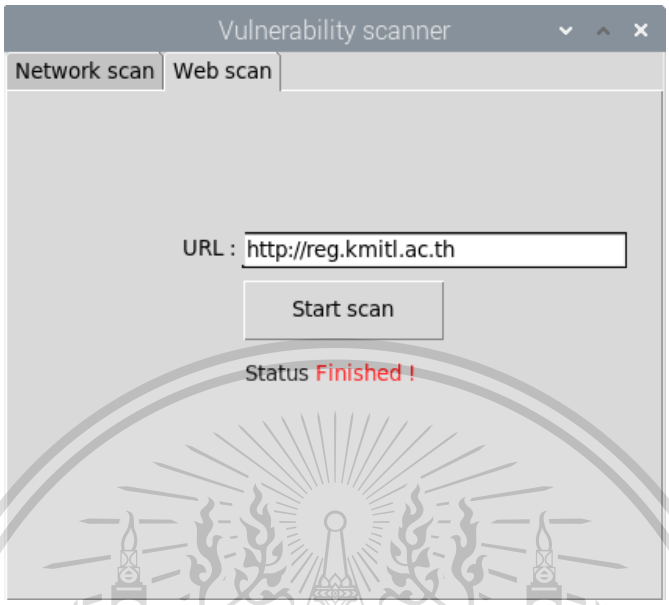
ต่อมาในส่วนของหน้า User GUI ของการสแกน Web จะมีช่องสำหรับให้กรอก URL ของเว็บที่ต้องการสแกนค้นหาช่องโหว่ตามรูปที่ 3.15 โดยเราจะใส่ URL ของมหาวิทยาลัยคือ http//: reg.kmitl.ac.th ตามรูปที่ 3.16 และสแกนสำเร็จตามรูป 3.17



รูปที่ 3.15 หน้าต่าง User GUI ของ Web scan

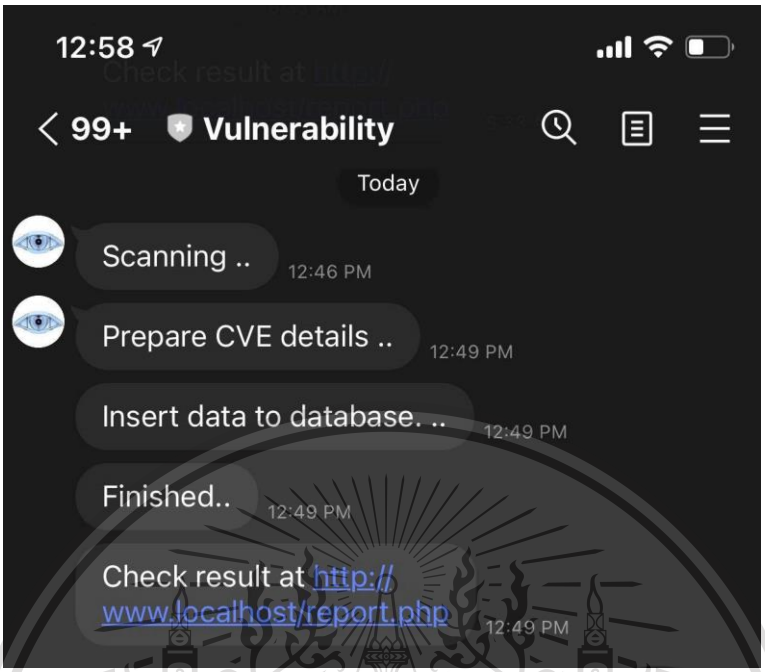


รูปที่ 3.16 หน้า User GUI ของการสแกน Web โดยกรอก URL ของมหาวิทยาลัย

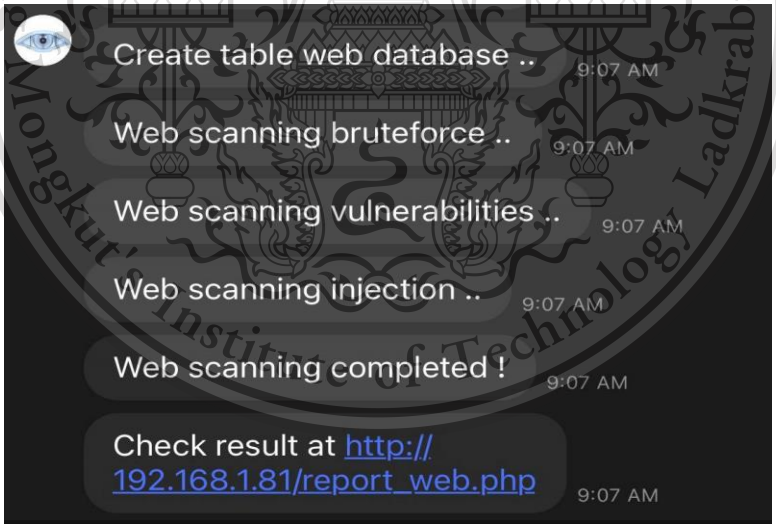


รูปที่ 3.17 หน้า User GUI ที่แสดงสถานะสแกนเว็ป สำเร็จ

หลังผู้ใช้งานทำการสแกน Network หรือ Web เสร็จสิ้นจะมีการแจ้งเตือนผ่าน Line ที่เป็น Line OA Vulnerability ดังรูป 3.18 และ 3.19



รูปที่ 3.18 ตัวอย่างการแจ้งเตือนผ่านไลน์ของ Network scan



รูปที่ 3.19 ตัวอย่างการแจ้งเตือนผ่านไลน์ของ Web scan

3.1.4 การออกแบบฐานข้อมูลในการเก็บรายละเอียดของ CVE

ในการที่จะสร้างฐานข้อมูลบน Raspberry Pi อันดับแรกต้องลงโปรแกรมเหล่านี้ไปก่อนคือ Apache2 , Mysql และ phpMyAdmin เพื่อไว้ใช้จัดการฐานข้อมูล

ในการสร้างฐานข้อมูลเพื่อทำการเก็บข้อมูลรายละเอียดต่างๆ ของ CVE และผลการค้นหาช่องโหว่ประกอบไปด้วยตาราง คำอธิบาย , CVSS Score , ความรุนแรง , vector และวิธีแก้ไขของแต่ละ CVE โดยข้อมูลดังกล่าวนำมาจาก https://nvd.nist.gov/ ซึ่งเป็นเว็บไซต์ของรัฐบาลสหรัฐอเมริกาที่รวบรวมช่องโหว่ประเภท CVE ไว้โดยนำข้อมูลคำอธิบายของแต่ละ CVE ที่ได้มาไปสร้างตารางในฐานข้อมูลบน Raspberry Pi ซึ่งประกอบไปด้วยตารางข้อมูล CVSS Score ความรุนแรง และ VECTOR ของแต่ละ CVE ดังรูปที่ 3.20

IP	PORT	STATE	SERVICE	CVE	SCORE	DES
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10082	6.4	https://vulners.com/cve/CVE-2019-10082
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10097	6.0	https://vulners.com/cve/CVE-2019-10097
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0217	6.0	https://vulners.com/cve/CVE-2019-0217
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0215	6.0	https://vulners.com/cve/CVE-2019-0215
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-1927	5.8	https://vulners.com/cve/CVE-2020-1927
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10098	5.8	https://vulners.com/cve/CVE-2019-10098
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-9490	5.0	https://vulners.com/cve/CVE-2020-9490
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-1934	5.0	https://vulners.com/cve/CVE-2020-1934
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10081	5.0	https://vulners.com/cve/CVE-2019-10081
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0220	5.0	https://vulners.com/cve/CVE-2019-0220
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0196	5.0	https://vulners.com/cve/CVE-2019-0196
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0197	4.9	https://vulners.com/cve/CVE-2019-0197
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-11993	4.3	https://vulners.com/cve/CVE-2020-11993
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10092	4.3	https://vulners.com/cve/CVE-2019-10092
IP address : 192.168.1.54	3389/tcp	open	ms-wbt-server	NULL	NULL	NULL
IP address : 192.168.1.62	8001/tcp	open	vcom-tunnel?	NULL	NULL	NULL
IP address : 192.168.1.62	8002/tcp	open	ssl/teradataordbms?	NULL	NULL	NULL
IP address : 192.168.1.62	8080/tcp	open	http	NULL	NULL	NULL
IP address : 192.168.1.62	9080/tcp	open	http	NULL	NULL	NULL
IP address : 192.168.1.1	21/tcp	filtered	ftp	NULL	NULL	NULL
IP address : 192.168.1.1	22/tcp	filtered	ssh	NULL	NULL	NULL
IP address : 192.168.1.1	23/tcp	filtered	telnet	NULL	NULL	NULL
IP address : 192.168.1.1	53/tcp	open	domain	NULL	NULL	NULL
IP address : 192.168.1.1	80/tcp	open	ssl/http	NULL	NULL	NULL
IP address : 192.168.1.1	443/tcp	filtered	https	NULL	NULL	NULL

รูปที่ 3.20 ตารางข้อมูลที่เก็บคำอธิบายของแต่ละ CVE ซึ่งประกอบไปด้วยตารางข้อมูล CVSS Score ความรุนแรง

ID	ATTACK	SCAN
2021_04_22_23_41_19	bruteforce	22-Apr-21 23:41:21 - scrapy.utils.log - INFO - Scr...
2021_04_22_23_41_19	vulnerabilities	22-Apr-21 23:41:26 - scrapy.utils.log - INFO - Scr...
2021_04_22_23_41_19	injection	22-Apr-21 23:41:32 - scrapy.utils.log - INFO - Scr...

รูปที่ 3.21 ตารางข้อมูลเก็บการค้นหาช่องโหว่เมื่อใช้เครื่องมือ Sitadel

จากตารางข้อมูลที่เก็บการค้นหาช่องโหว่ , ตารางข้อมูลที่เก็บวิธีแก้ไขที่เป็นHyperlink ของแต่ละ CVE , ตารางข้อมูลที่เก็บ CVSS Score , ความรุนแรง และ Vector ของแต่ละ CVE , ตารางข้อมูลที่เก็บคำอธิบายของแต่ละ CVE , ตารางข้อมูลเก็บการค้นหาช่องโหว่เมื่อใช้เครื่องมือ Nmap และ ตารางข้อมูลเก็บการค้นหาช่องโหว่เมื่อใช้เครื่องมือ Sitadel

สามารถนำมาสร้างความสัมพันธ์ระหว่างตารางข้อมูล (Entity) โดยมี Primary Key (PK) เป็นหมายเลขไอพีได้ดังรูปที่ 3.22

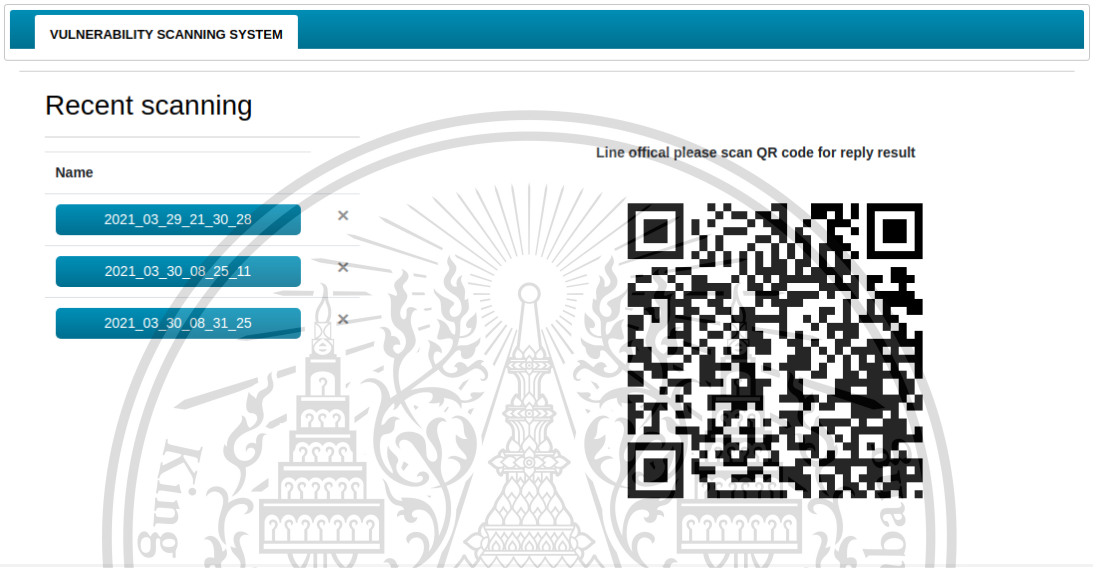
รูปที่ 3.22 ความสัมพันธ์ระหว่างตารางข้อมูล

3.1.5 การออกแบบส่วนแสดงผลบนเว็บแอปพลิเคชัน

การออกแบบส่วนแสดงผลสำหรับอ่านค่าผลลัพธ์ที่ได้จากการใช้งานตัวโปรแกรมบน Raspberry Pi 4 ของผู้ใช้งาน ได้มีการออกแบบแสดงผลบนเว็บแอปพลิเคชัน เพื่อให้ง่ายต่อการใช้งานอ่านผลที่ได้จากการสแกน โดยในเว็บแอปพลิเคชันจะทำการดึงผลจากฐานข้อมูล MySQL นำมาแสดงผล โดยมีรายละเอียดดังนี้

3.1.5.1 การออกแบบหน้าจอเลือกดูผลการสแกนในแต่ละครั้ง

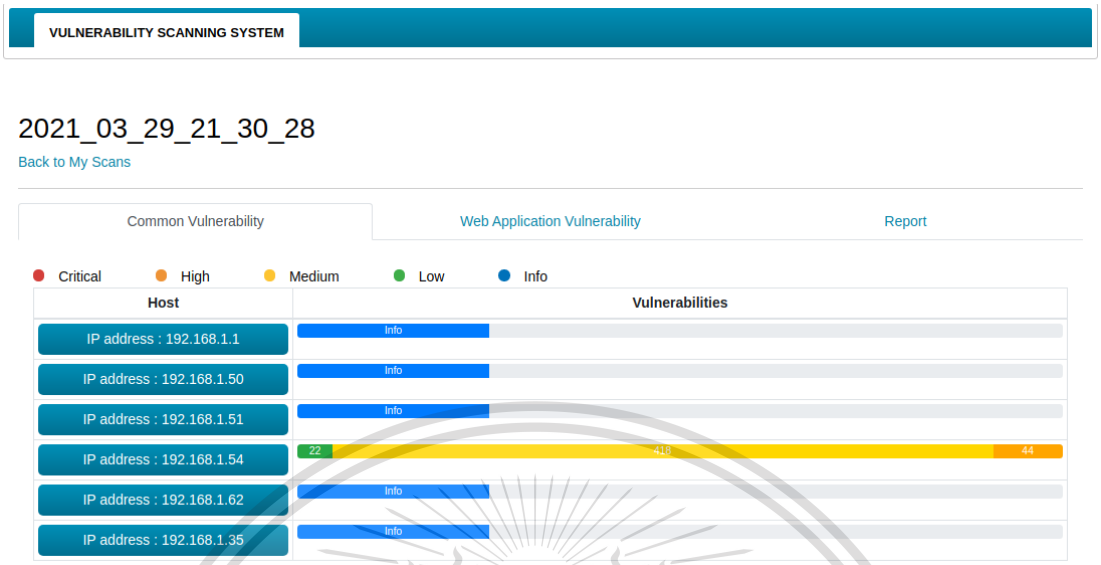
หน้าเมนูการเลือกดูผลการสแกน ทุกครั้งที่มีการสแกนจะมีการแสดงชื่อผลการสแกนในแต่ละครั้ง บนหน้าจอนี้ โดยเมื่อทำการคลิกเข้าไปในตัวเลือกนั้น ๆ จะเป็นการเข้าไปเพื่อดูรายละเอียดของผลการสแกนโดยผลการสแกนแต่ละครั้งสามารถลบได้จากหน้านี้ ในส่วนของการแจ้งเตือนสามารถ แอดไลน์ OA ผ่าน QR code ที่แสดงบน เว็บแอปพลิเคชัน



รูปที่ 3.23 หน้าเมนูการเลือกดูผลการค้นหาและรับการแจ้งเตือนบนเว็บแอปพลิเคชัน

3.1.5.2 การออกแบบหน้าเว็บเพื่อเลือกดูไอพีที่ตัวโปรแกรมทำการสแกนพบ

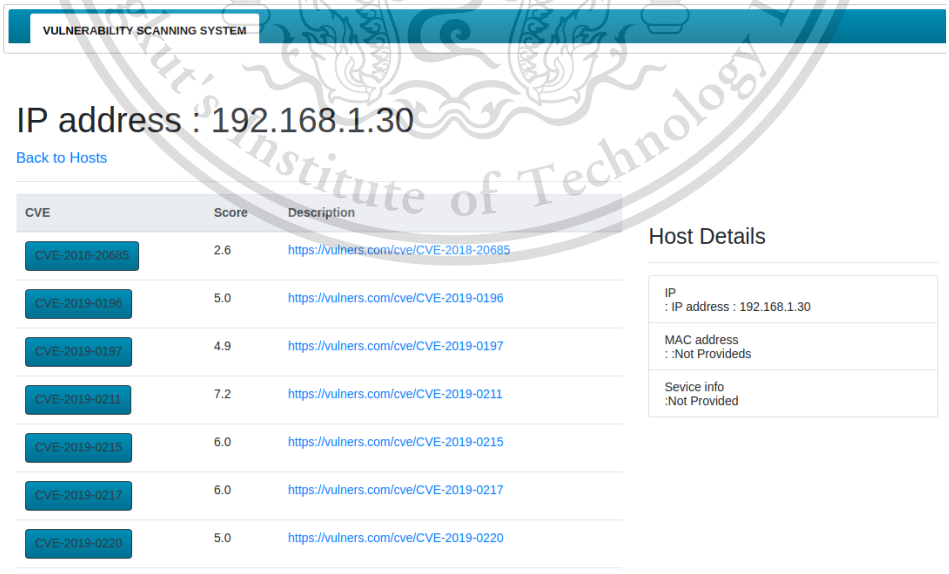
ทำการแสดงหน้าเว็บเพื่อเลือกดูรายละเอียดของหมายเลขไอพีแอดเดรสที่ตัวโปรแกรมทำการสแกนพบในการสแกนนั้น ๆ มีโปรเกรสบาร์ในการบอกจำนวนของ CVE(Common Vulnerabilities and Exposures) และระดับความรุนแรงด้วยสี ในกรณีที่ไม่พบ CVE โปรเกรสบาร์จะแสดงขึ้นเพียงว่าพบเพียงข้อมูล โดยที่การคลิกไปที่ไอพีนั้น ๆ เป็นการเลือกดูรายละเอียดของหมายเลขไอพีแอดเดรสดังรูปที่ 3.24



รูปที่ 3.24 ออกแบบหน้าเว็บเพื่อเลือกดูหมายเลขไอพีที่ตัวโปรแกรมทำการค้นหาพบ

3.1.5.3 การออกแบบแสดงรายละเอียดของในแต่ละหมายเลขไอพีแอดเดรส

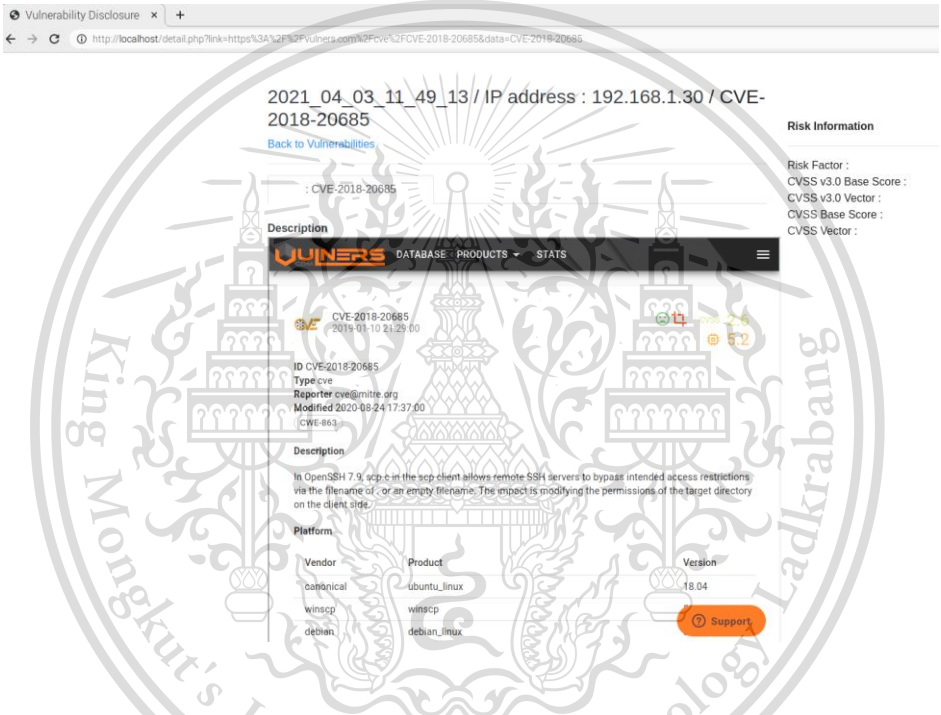
หน้าต่างแสดงรายละเอียดของในแต่ละหมายเลขไอพีแอดเดรสที่ต้องทราบของระดับความรุนแรงและค่า CVSS (Common Vulnerability Scoring System) รายละเอียดของ MAC address และ Service info ของหมายเลขไอพีแอดเดรส โดยที่การคลิกไปที่ CVE นั้น ๆ ดังรูปที่ 3.25



รูปที่ 3.25 หน้าแสดงรายละเอียดของในแต่ละหมายเลขไอพี

3.1.5.4 การออกแบบหน้าแสดงรายละเอียด CVE

ในส่วนของหน้าเว็บแสดงรายละเอียดของ CVE จะมีการบอกถึงรายละเอียดของ CVE นั้น ๆ ได้แก่คำอธิบายถึงรายละเอียด แหล่งอ้างอิงข้อมูลวิธีแก้ไขอยู่ในรูปแบบ Hyperlink ข้อมูลด้านความเสี่ยง อันประกอบไปด้วย ระดับความรุนแรง ค่า CVSS และ CVSS Vector ในทั้ง v2.0 และ v3.0 ในส่วนสุดท้ายคือการแสดงพอร์ตที่สแกนพบใน บริการของพอร์ต ของหมายเลขไอพีแอดเดรสดังรูปที่ 3.26



รูปที่ 3.26 การออกแบบหน้าแสดงรายละเอียด CVE

3.1.5.5 การออกแบบหน้าเว็บเพื่อเลือกดูผลลัพธ์การสแกนของ Sitadel ในการสแกนแบบ bruteforce, vulnerabilities , injection ดังรูปที่ 3.26, 3.27 และ 3.28

2021_04_30_09_07_11

Back to My Scans

Result scan vulnerability | Report

bruteforce

30-Apr-21 09:07:13 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot) 30-Apr-21 09:07:13 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0, Twisted 21.2.0, Python 3.7.3 (default, Jul 25 2020, 13:03:44) - [GCC 8.3.0], pyOpenSSL 19.0.0 (OpenSSL 1.1.1d 10 Sep 2019), cryptography 2.6.1, Platform Linux-5.10.17-v7l+-armv7l-with-debian-10.8 30-Apr-21 09:07:13 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor 30-Apr-21 09:07:13 - scrapy.crawler - INFO - Overridden settings: {'CONCURRENT_REQUESTS': 15, 'LOG_LEVEL': 'CRITICAL', 'RETRY_ENABLED': False, 'USER_AGENT': 'Sitadel 1.0.1'} 30-Apr-21 09:07:13 - scrapy.extensions.telnet - INFO - Telnet Password: f8753bf19881cb69 30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled extensions: ['scrapy.extensions.corestats.CoreStats', 'scrapy.extensions.telnet.TelnetConsole', 'scrapy.extensions.memusage.MemoryUsage', 'scrapy.extensions.logstats.LogStats'] 30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled downloader middlewares: ['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware', 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware', 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware', 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware', 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware', 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware', 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware', 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware', 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware', 'scrapy.downloadermiddlewares.stats.DownloaderStats'] 30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled spider middlewares: ['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware', 'scrapy.spidermiddlewares.offsite.OffsiteMiddleware', 'scrapy.spidermiddlewares.referer.RefererMiddleware', 'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware', 'scrapy.spidermiddlewares.depth.DepthMiddleware'] 30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled item pipelines: [] 30-Apr-21 09:07:13 - scrapy.core.engine - INFO - Spider opened 30-Apr-21 09:07:13 - scrapy.extensions.logstats - INFO - Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min) 30-Apr-21 09:07:13 - scrapy.extensions.telnet - INFO - Telnet console listening on 127.0.0.1:6023 30-Apr-21 09:07:13 - scrapy.downloadermiddlewares.redirect - DEBUG - Redirecting (302) to from 30-Apr-21 09:07:13 - scrapy.downloadermiddlewares.redirect - DEBUG - Redirecting (301) to from 30-Apr-21 09:07:13 - scrapy.core.engine - DEBUG - Crawled (200) (referer: None) 30-Apr-21 09:07:13 - scrapy.core.engine - INFO - Closing spider (finished) 30-Apr-21 09:07:13 - scrapy.statscollectors - INFO - Dumping Scrapy stats: {'downloader/request_bytes': 690, 'downloader/request_count': 3, 'downloader/request_method_count/GET': 3, 'downloader/response_bytes': 6826, 'downloader/response_count': 3, 'downloader/response_status_count/200': 1, 'downloader/response_status_count/301': 1, 'downloader/response_status_count/302': 1, 'elapsed_time_seconds': 0.265629, 'finish_reason': 'finished', 'finish_time': datetime.datetime(2021, 4, 30, 2, 7, 13, 651349), 'memusage/max': 39272448, 'memusage/startup': 39272448, 'response_received_count': 1, 'scheduler/dequeued': 3, 'scheduler/dequeued/memory': 3, 'scheduler/enqueued': 3, 'scheduler/enqueued/memory': 3, 'start_time': datetime.datetime(2021, 4, 30, 2, 7, 13, 385720)} 30-Apr-21 09:07:13 - scrapy.core.engine - INFO - Spider closed (finished)

รูปที่ 3.27 ผลลัพธ์การสแกนแบบ bruteforce

vulnerabilities

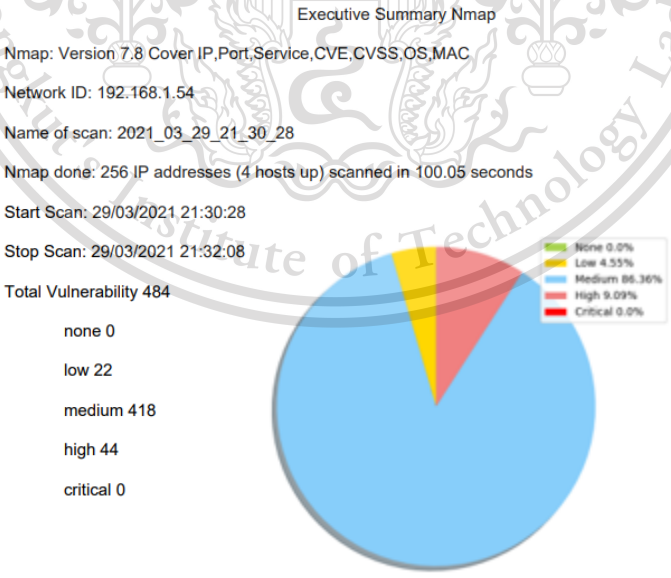
30-Apr-21 09:07:15 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot) 30-Apr-21 09:07:15 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0, Twisted 21.2.0, Python 3.7.3 (default, Jul 25 2020, 13:03:44) - [GCC 8.3.0], pyOpenSSL 19.0.0 (OpenSSL 1.1.1d 10 Sep 2019), cryptography 2.6.1, Platform Linux-5.10.17-v7l+-armv7l-with-debian-10.8 30-Apr-21 09:07:15 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor 30-Apr-21 09:07:15 - scrapy.crawler - INFO - Overridden settings: {'CONCURRENT_REQUESTS': 15, 'LOG_LEVEL': 'CRITICAL', 'RETRY_ENABLED': False, 'USER_AGENT': 'Sitadel 1.0.1'} 30-Apr-21 09:07:15 - scrapy.extensions.telnet - INFO - Telnet Password: e448a51644184df1 30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled extensions: ['scrapy.extensions.corestats.CoreStats', 'scrapy.extensions.telnet.TelnetConsole', 'scrapy.extensions.memusage.MemoryUsage', 'scrapy.extensions.logstats.LogStats'] 30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled downloader middlewares: ['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware', 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware', 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware', 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware', 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware', 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware', 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware', 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware', 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware', 'scrapy.downloadermiddlewares.stats.DownloaderStats'] 30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled spider middlewares: ['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware', 'scrapy.spidermiddlewares.offsite.OffsiteMiddleware', 'scrapy.spidermiddlewares.referer.RefererMiddleware', 'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware', 'scrapy.spidermiddlewares.depth.DepthMiddleware'] 30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled item pipelines: [] 30-Apr-21 09:07:15 - scrapy.core.engine - INFO - Spider opened 30-Apr-21 09:07:15 - scrapy.extensions.logstats - INFO - Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min) 30-Apr-21 09:07:15 - scrapy.extensions.telnet - INFO - Telnet console listening on 127.0.0.1:6023 30-Apr-21 09:07:15 - scrapy.downloadermiddlewares.redirect - DEBUG - Redirecting (302) to from 30-Apr-21 09:07:16 - scrapy.downloadermiddlewares.redirect - DEBUG - Redirecting (301) to from 30-Apr-21 09:07:16 - scrapy.core.engine - DEBUG - Crawled (200) (referer: None) 30-Apr-21 09:07:16 - scrapy.core.engine - INFO - Closing spider (finished) 30-Apr-21 09:07:16 - scrapy.statscollectors - INFO - Dumping Scrapy stats: {'downloader/request_bytes': 690, 'downloader/request_count': 3, 'downloader/request_method_count/GET': 3, 'downloader/response_bytes': 6826, 'downloader/response_count': 3, 'downloader/response_status_count/200': 1, 'downloader/response_status_count/301': 1, 'downloader/response_status_count/302': 1, 'elapsed_time_seconds': 0.270218, 'finish_reason': 'finished', 'finish_time': datetime.datetime(2021, 4, 30, 2, 7, 16, 211716), 'memusage/max': 39235584, 'memusage/startup': 39235584, 'response_received_count': 1, 'scheduler/dequeued': 3, 'scheduler/dequeued/memory': 3, 'scheduler/enqueued': 3, 'scheduler/enqueued/memory': 3, 'start_time': datetime.datetime(2021, 4, 30, 2, 7, 15, 941498)} 30-Apr-21 09:07:16 - scrapy.core.engine - INFO - Spider closed (finished)

รูปที่ 3.28 ผลลัพธ์การสแกนแบบ vulnerabilities

รูปที่ 3.29 ผลลัพธ์การสแกนแบบ injection

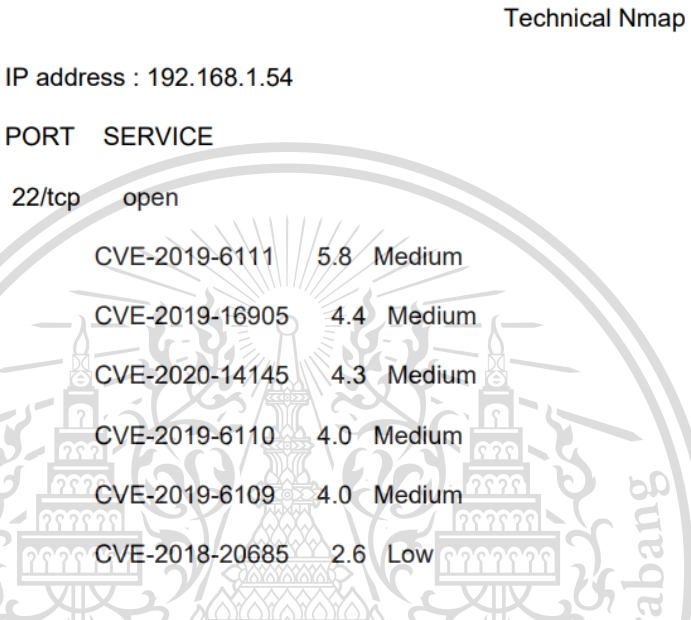
3.1.6 การออกแบบรายงายสรุปผลการค้นหาช่องโหว่ของ Nmap และ Sitadel

3.1.6.1 สรุปผลการค้นหาช่องโหว่โดยประกอบไปด้วยรายละเอียดของ Network Id ที่ทำการค้นหา , เวลาเริ่มต้นค้นหาช่อง , เวลาสิ้นสุดการค้นหาช่องโหว่ , Pie chart plot , จำนวนเครื่องที่ทำการค้นหาช่องโหว่ โดย Executive Nmap จะเป็นดังรูปที่ 3.30



รูปที่ 3.30 รายงาน Executive Nmap

3.1.6.2 ผลการค้นหาช่องโหว่ในแต่ละหมายเลขไอพีประกอบไปด้วยหมายเลขไอพี , พอร์ตที่พบ , บริการของพอร์ตที่เปิดใช้งาน , เวอร์ชันของบริการที่เปิดใช้งาน , หมายเลข CVE ที่พบ , CVSS Score , ระดับความรุนแรง , MAC Address และระบบปฏิบัติการโดยรายงานที่สร้างจะใช้ชื่อ Technical Nmap แสดงดังรูปที่ 3.31


Technical Nmap

IP address : 192.168.1.54

PORT SERVICE

22/tcp open

CVE-2019-6111 5.8 Medium

CVE-2019-16905 4.4 Medium

CVE-2020-14145 4.3 Medium

CVE-2019-6110 4.0 Medium

CVE-2019-6109 4.0 Medium

CVE-2018-20685 2.6 Low

รูปที่ 3.31 รายงาน Technical Nmap

3.1.6.2 ผลการค้นหาทั้งหมดของ Sitadel ในการสแกนแบบ bruteforce, vulnerabilities , injection เป็นไฟล์ pdf แสดงดังรูปที่ 3.32,3.33 และ 3.34

2021_04_30_09_07_11

Bruteforce

30-Apr-21 09:07:13 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot)

30-Apr-21 09:07:13 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0

30-Apr-21 09:07:13 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor

30-Apr-21 09:07:13 - scrapy.crawler - INFO - Overridden settings:

{'CONCURRENT_REQUESTS': 15,

'LOG_LEVEL': 'CRITICAL',

'RETRY_ENABLED': False,

'USER_AGENT': 'Sitadel 1.0.1'}

30-Apr-21 09:07:13 - scrapy.extensions.telnet - INFO - Telnet Password: f8753bf19881cb69

30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled extensions:

['scrapy.extensions.corestats.CoreStats',

'scrapy.extensions.telnet.TelnetConsole',

'scrapy.extensions.memusage.MemoryUsage',

'scrapy.extensions.logstats.LogStats']

30-Apr-21 09:07:13 - scrapy.middleware - INFO - Enabled downloader middlewares:

['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',

'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',

'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',

รูปที่ 3.32 รายงาน Sitadel แบบ bruteforce

Vulnerabilities

30-Apr-21 09:07:15 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot)

30-Apr-21 09:07:15 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22

30-Apr-21 09:07:15 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor

30-Apr-21 09:07:15 - scrapy.crawler - INFO - Overridden settings:

{'CONCURRENT_REQUESTS': 15,

'LOG_LEVEL': 'CRITICAL',

'RETRY_ENABLED': False,

'USER_AGENT': 'Sitadel 1.0.1'}

30-Apr-21 09:07:15 - scrapy.extensions.telnet - INFO - Telnet Password: e448a51644184df1

30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled extensions:

['scrapy.extensions.corestats.CoreStats',

'scrapy.extensions.telnet.TelnetConsole',

'scrapy.extensions.memusage.MemoryUsage',

'scrapy.extensions.logstats.LogStats']

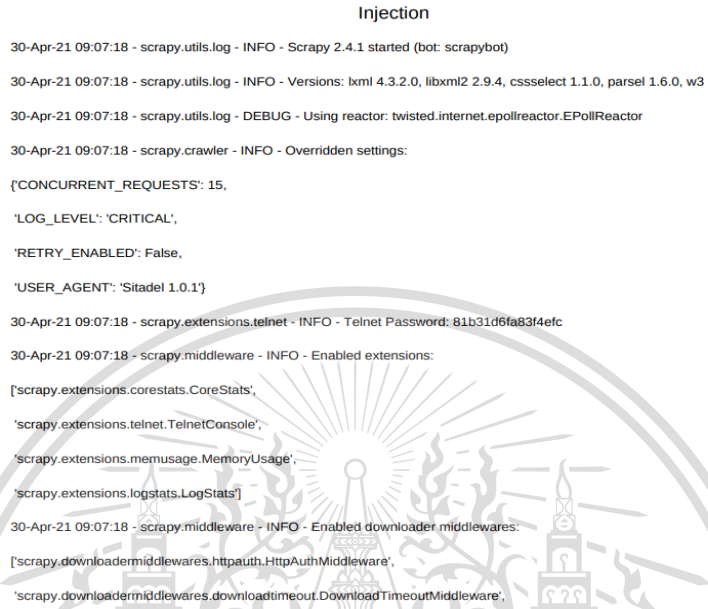
30-Apr-21 09:07:15 - scrapy.middleware - INFO - Enabled downloader middlewares:

['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',

'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',

'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',

รูปที่ 3.33 รายงาน Sitadel แบบ vulnerabilities



รูปที่ 3.34 รายงาน Sitadel แบบ injection

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 อุปกรณ์ที่ใช้ในการทดลอง

3.2.1.1 หน้าจอคอมพิวเตอร์

3.2.1.2 Raspberry Pi 4

3.2.2 ระบบปฏิบัติการและโปรแกรมที่ใช้ในการทดลอง

3.2.2.1 ระบบปฏิบัติการ Raspbian

3.2.2.2 Nmap

3.2.2.3 Sitadel

3.2.2.4 Apache2

3.2.2.5 Mysql

3.2.3 ภาษาที่ใช้ในการสร้างโปรแกรม

3.2.3.1 ไพธอน

3.2.3.2 เอชทีเอ็มแอล

3.2.3.3 พีเอชพี

3.3 การจัดเก็บผลการทดลอง

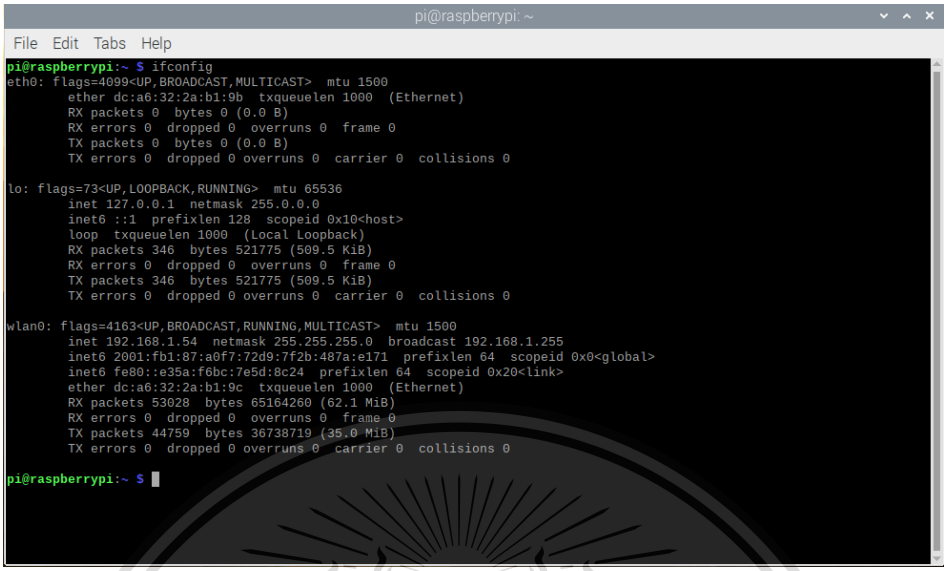
ระบบตรวจจับช่องโหว่ทำงานบนอุปกรณ์ Raspberry Pi 4 โดยที่อุปกรณ์ Raspberry Pi ที่นำมาใช้งานเป็น Raspberry Pi 4 Model B มีรายละเอียดดังนี้ RAM 2 GB ROM 16 GB รองรับ Wireless LAN และ Gigabit Ethernet ทำงานอยู่บนระบบปฏิบัติการ Raspbian Os ใช้งานร่วมกับหน้าจอคอมพิวเตอร์ และใช้เครื่องมือทำการสแกน 2 ตัว

3.3.1 การใช้งาน Nmap

ส่วนของการค้นหาช่องโหว่ คือ ส่วนที่มีการใช้โปรแกรม Nmap เวอร์ชันที่ใช้งานเป็นเวอร์ชัน 7.8 โดยมีการใช้ฟีเจอร์ Nmap Scripting Engine (NSE) ทำให้ความสามารถในการทำงานของ Nmap นั้นถูกขยายขอบเขตออกไปขึ้นอยู่กับฟังก์ชันการทำงานของสคริปต์ที่ถูกเรียกใช้ สคริปต์ที่เราเลือกใช้คือ vulscan NSE script ตัวสคริปต์ถูกออกแบบเพื่อเสริม Nmap's version detection โดยจะให้ค่า ช่องโหว่ CVE (Common Vulnerabilities and Exposures) เริ่มต้นการค้นหาช่องโหว่ด้วยการใส่ IP Address (Internet Protocol address) โดยการใส่ IP Address สามารถใส่ได้ 3 แบบคือ IP เดียว , หลาย IP หรือเป็นช่วงของหมายเลข IP

3.3.1.1 การค้นหา IP Address

IP Address สามารถหาได้จาก ifconfig (interface configuration) โดยเป็นคำสั่งที่จะแสดงข้อมูลการองค์ประกอบเครือข่ายในปัจจุบัน บนระบบปฏิบัติการ ดังรูปที่ 3.35



รูปที่ 3.35 ข้อมูลโครงข่ายในปัจจุบัน

3.3.1.2 การเพิ่ม vulscan NSE script

ในตัวของ Nmap นั้นจะมี NSE script อยู่ภายในตัวแล้วและสามารถเพิ่ม NSE script ที่ต้องการได้ด้วยคำสั่ง git clone ซึ่งเป็นคำสั่งที่จะทำการดาวน์โหลด Git repository มาในคอมพิวเตอร์ ในที่นี้เราจะทำการเพิ่ม vulscan NSE script ได้ดังรูปที่ 3.36

รูปที่ 3.36 การเพิ่ม vulscan NSE script

3.3.1.3 การใช้งานคำสั่ง vulscan NSE script

การใช้งาน NSE script ทำได้โดยการใส่ –script ไปในคำสั่ง Nmap ตามด้วยตัวสคริปต์ที่จะใช้งาน เพื่อที่จะใช้ vulnscan script ดังรูปที่ 3.27 โดย -sV เป็นการตรวจหาข้อมูลเวอร์ชันเป้าหมาย -A เป็นการค้นหาข้อมูลเบื้องต้นในแต่ละหมายเลขไอพี –script-args vulscandb=cve.csv เป็นการค้นหาโดยใช้ฐานข้อมูล cve.csv หลังจากนั้นตามด้วยเครือข่ายที่ทำการค้นหาช่องโหว่ โดยมีผลลัพธ์ที่ได้ดังรูปที่ 3.37

```
pi@raspberrypi:~ $ nmap -sV -A--script=vulscan/vulscan.nse --script-arg vulscandb=cve.csv 192.168.1.0/24
```

รูปที่ 3.37 คำสั่ง nmap โดยใช้ vulscan NSE script

```
Nmap scan report for 192.168.1.54
Host is up (0.0027s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 7.9p1 Raspbian 10+deb10u2+rpt1 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:7.9p1:
|       EXPLOITPACK:98FE96309F9524B8C84C508837551A19    5.8     https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A19
|       EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97    5.8     https://vulners.com/exploitpack/EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97
|       EDB-ID:46516    5.8     https://vulners.com/exploitdb/EDB-ID:46516      *EXPLOIT*
|       CVE-2019-6111   5.8     https://vulners.com/cve/CVE-2019-6111
|       CVE-2019-16905  4.4     https://vulners.com/cve/CVE-2019-16905
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|       CVE-2019-6110   4.0     https://vulners.com/cve/CVE-2019-6110
|       CVE-2019-6109   4.0     https://vulners.com/cve/CVE-2019-6109
|       CVE-2018-20685  2.6     https://vulners.com/cve/CVE-2018-20685
|       PACKETSTORM:151227      0.0     https://vulners.com/packetstorm/PACKETSTORM:151227      *EXPLOIT*
|       EDB-ID:46193    0.0     https://vulners.com/exploitdb/EDB-ID:46193      *EXPLOIT*
|_      1337DAY-ID-32009        0.0     https://vulners.com/zdt/1337DAY-ID-32009        *EXPLOIT*
80/tcp   open  http          Apache httpd 2.4.38 ((Raspbian))
|_http-server-header: Apache/2.4.38 (Raspbian)
| vulners:
|   cpe:/a:apache:http_server:2.4.38:
|       CVE-2020-11984  7.5     https://vulners.com/cve/CVE-2020-11984
|       EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB    7.2     https://vulners.com/exploitpack/EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB
|       CVE-2019-0211   7.2     https://vulners.com/cve/CVE-2019-0211
|       1337DAY-ID-32502        7.2     https://vulners.com/zdt/1337DAY-ID-32502        *EXPLOIT*
|       CVE-2019-10082  6.4     https://vulners.com/cve/CVE-2019-10082
|       CVE-2019-10097  6.0     https://vulners.com/cve/CVE-2019-10097
|       CVE-2019-0217   6.0     https://vulners.com/cve/CVE-2019-0217
|       CVE-2019-0215   6.0     https://vulners.com/cve/CVE-2019-0215
|       EDB-ID:47689    5.8     https://vulners.com/exploitdb/EDB-ID:47689      *EXPLOIT*
|       CVE-2020-1927   5.8     https://vulners.com/cve/CVE-2020-1927
|       CVE-2019-10098  5.8     https://vulners.com/cve/CVE-2019-10098
|       1337DAY-ID-33577        5.8     https://vulners.com/zdt/1337DAY-ID-33577        *EXPLOIT*
|       CVE-2020-9490   5.0     https://vulners.com/cve/CVE-2020-9490
|       CVE-2020-1934   5.0     https://vulners.com/cve/CVE-2020-1934
|       CVE-2019-10081  5.0     https://vulners.com/cve/CVE-2019-10081
|       CVE-2019-0220   5.0     https://vulners.com/cve/CVE-2019-0220
|       CVE-2019-0196   5.0     https://vulners.com/cve/CVE-2019-0196
```

รูปที่ 3.38 ผลลัพธ์จากคำสั่ง nmap-vulners NSE script

3.3.2 การใช้งาน sitadel

ในการติดตั้ง Sitadel เป็นการอัปเดตสำหรับ WAScan ทำให้เข้ากันได้กับภาษาไพธอน > = 3.4 และมีคำสั่งติดตั้งดังรูป 3.39

```
pi@raspberrypi: ~
File  Edit  Tabs  Help
pi@raspberrypi:~ $ git clone https://github.com/shenril/sitadel.git
```

รูปที่ 3.39 คำสั่งในการติดตั้ง Sitadel

โดยจะใช้คำสั่ง -r และ -a โดยคำสั่งของ -r คือ การตัดสินใจเลือกระดับความเสี่ยงที่ต้องการให้ Sitadel ทำงาน และ -a คือ การโจมตีของใช้งานบนเว็บไซต์มีโมดูลให้เลือกหลายอย่าง แต่ใช้เพียง 3 โมดูลคือ bruteforce, injection, vulns ในรูป 3.40 คือการเรียกใช้งาน โมดูลของ Sitadel

```
pi@raspberrypi:~ $ sitadel.py -r 1-a ['vulns','injection','bruteforce']http://reg.kmitl.ac.th
```

รูปที่ 3.40 script ที่ใช้ในการค้นหาของ Sitadel

เมื่อทำการค้นหาช่องโหว่เสร็จเรียบร้อยจะได้ผลการสแกนเป็นไฟล์ log files ดังรูปที่ 3.41

```
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): reg.kmitl.ac.th:80
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /guestbook2/ HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): reg.kmitl.ac.th:80
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /guestbook_backup/ HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /guestbook1/ HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /client~/ HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): reg.kmitl.ac.th:80
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /guestbook_old/ HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): reg.kmitl.ac.th:80
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /client.tar HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:http://reg.kmitl.ac.th:80 "HEAD /client.zip HTTP/1.1" 404 0
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): reg.kmitl.ac.th:80
```

รูปที่ 3.41 ผลการค้นหาช่องโหว่ของ sitadel ในรูปแบบ log files

บทที่ 4

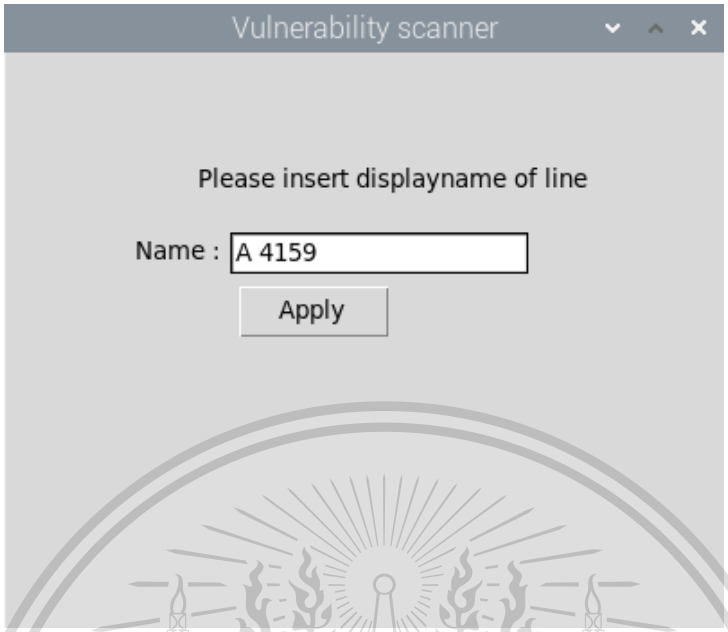
ผลการทดลอง

4.1 ผลการทดลองเชื่อมต่อเครือข่าย

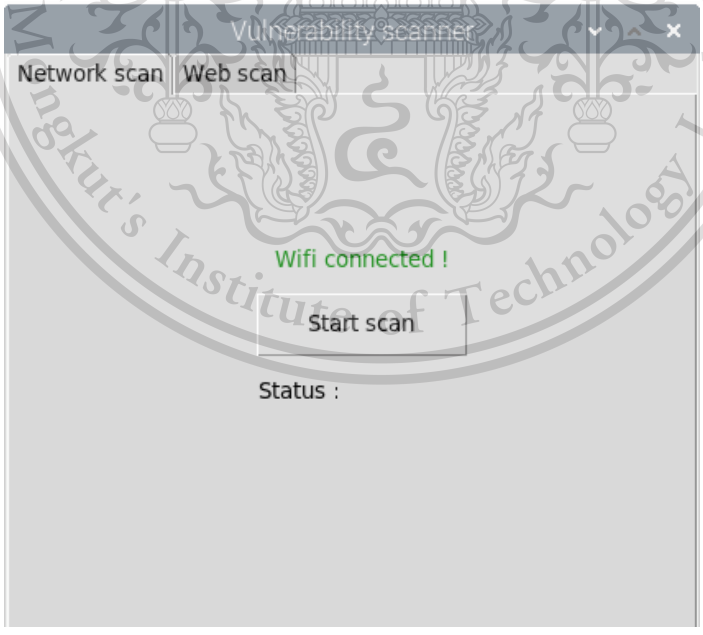
4.1.1 ผลการทดลองเชื่อมต่อผ่านเครือข่ายผ่าน Wi-Fi

เมื่อผู้ใช้งานทำการรันคำสั่งการทำงานขึ้นมาจะได้หน้าต่างตามรูปที่ 4.1 และใส่ Display Name ของไลน์ที่ทำการแอดเพื่อนกับ Line OA ไว้ ตามรูปที่ 4.2 จากนั้นกด Apply จะขึ้นหน้าต่างสแกนช่องโหว่และขึ้นสถานะว่า Wifi Connected พร้อมใช้งาน ตามรูปที่ 4.3

รูปที่ 4.1 หน้าต่างเริ่มต้นการทำงาน



รูปที่ 4.2 หน้าต่างให้กรอกชื่อ Display name ที่เพิ่มเพื่อนไว้กับไลน์ OA



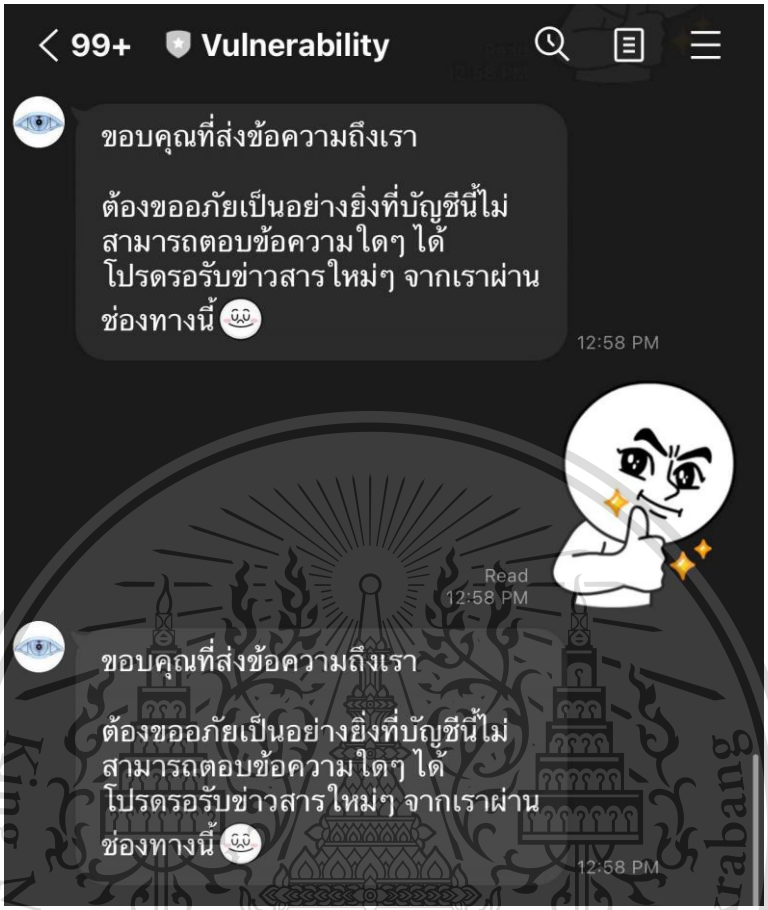
รูปที่ 4.3 หน้าต่างสแกนช่องโหว่พร้อมใช้งาน

4.2 ผลการทดลองการเก็บ Display name และ User id ของไลน์ และนำไปเก็บบนเว็ปเซิฟเวอร์

4.2.1 ทำการแอดเพื่อนไลน์ OA ที่ชื่อว่า Vulnerability

การที่จะได้รับการแจ้งเตือนผ่านไลน์ OA นั้นจะต้องทำการแอดเพื่อนผ่านระบบของไลน์หรือสแกน QR Code ตามรูปที่ 4.4 หรือส่งข้อความไปหาไลน์ OA ตามรูปที่ 4.5

รูปที่ 4.4 ทำการแอดเพื่อนไลน์ OA บนมือถือ



รูปที่ 4.5 ส่งข้อความไปหาไลน์ OA

4.2.2 เก็บ Display name และ User id ไลน์ ไว้ที่เวปเซิฟเวอร์

เก็บบันทึก Display name และ User id line ไลน์ไว้ที่เว็ปเซิฟเวอร์หรือ Web hook เพื่อที่จะทำการแจ้งเตือนไปหาผู้ใช้งานที่ต้องการรับแจ้งเตือนตามรูปที่ 4.6

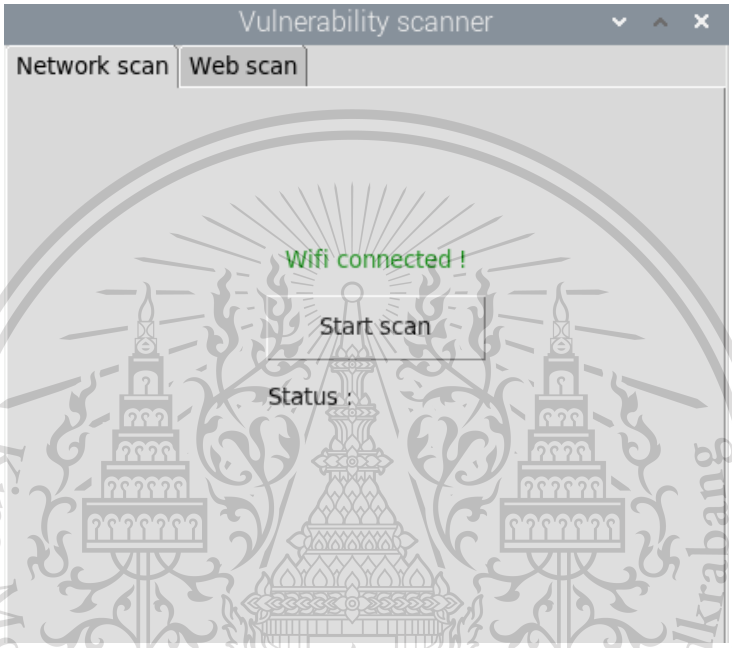
```
A 4159@Ucb0cd6e6b34905309e18920f4be375e5
Smileilu@U91be1439f80601a1808ea9d8809c0f42
```

รูปที่ 4.6 Display name และ User id ของไลน์ผู้ใช้

4.3 ผลการทดลองการค้นหาช่องโหว่

4.3.1 การค้นหาช่องโหว่แบบ Network Scan

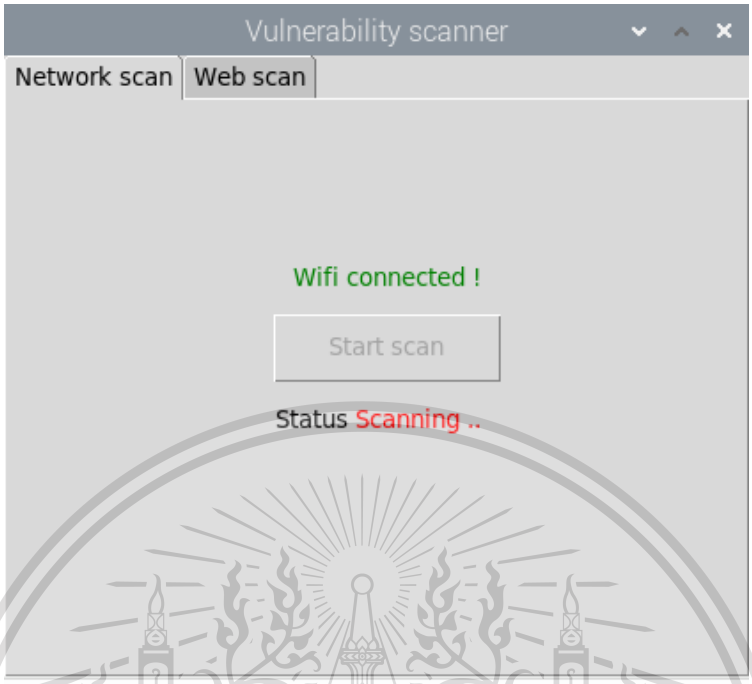
การค้นหาช่องโหว่แบบ Network Scan เป็นการค้นหาช่องโหว่โดยใช้เครื่องมือ Nmap เพื่อทำการค้นหาช่องโหว่ ดังรูปที่ 4.7



รูปที่ 4.7 เลือกการค้นหาช่องโหว่แบบ Network Scan

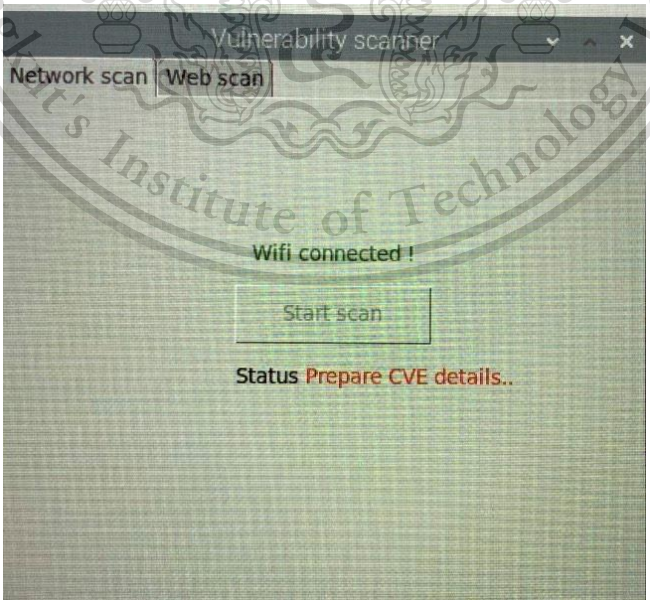
4.3.1.1 สถานะ Scanning

การค้นหาช่องโหว่แบบ Network Scan หลังจากที่เลือกการค้นหาทำการกดปุ่ม Scan เพื่อทำการเริ่มการค้นหาโดยระบบจะแสดงสถานะ Scanning ดังรูปที่ 4.8



รูปที่ 4.8 สถานะ Scanning ของการสแกนแบบ Network Scan

4.3.1.2 สถานะ Prepare CVE details แสดงดังรูปที่ 4.4 เป็นสถานะที่แสดงว่าการค้นหาช่องโหว่เสร็จเรียบร้อยและกำลังทำการสร้างฐานข้อมูลและแสดงให้เห็นว่ากำลังเตรียมข้อมูลเก็บยังฐานข้อมูลโดยสามารถตรวจสอบได้จากฐานข้อมูลผ่านทาง phpMyAdmin รูปที่ 4.9



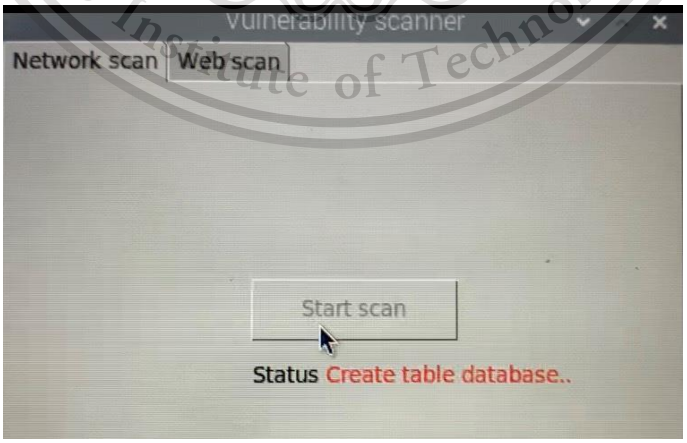
รูปที่ 4.9 สถานะ Prepare CVE detail ของการสแกนแบบ Network Scan

IP	PORT	STATE	SERVICE	CVE	SCORE	DES
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10082	6.4	https://vulners.com/cve/CVE-2019-10082
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10097	6.0	https://vulners.com/cve/CVE-2019-10097
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0217	6.0	https://vulners.com/cve/CVE-2019-0217
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0215	6.0	https://vulners.com/cve/CVE-2019-0215
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-1927	5.8	https://vulners.com/cve/CVE-2020-1927
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10098	5.8	https://vulners.com/cve/CVE-2019-10098
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-9490	5.0	https://vulners.com/cve/CVE-2020-9490
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-1934	5.0	https://vulners.com/cve/CVE-2020-1934
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10081	5.0	https://vulners.com/cve/CVE-2019-10081
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0220	5.0	https://vulners.com/cve/CVE-2019-0220
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0196	5.0	https://vulners.com/cve/CVE-2019-0196
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-0197	4.9	https://vulners.com/cve/CVE-2019-0197
IP address : 192.168.1.54	80/tcp	open	http	CVE-2020-11993	4.3	https://vulners.com/cve/CVE-2020-11993
IP address : 192.168.1.54	80/tcp	open	http	CVE-2019-10092	4.3	https://vulners.com/cve/CVE-2019-10092
IP address : 192.168.1.54	3389/tcp	open	ms-wbt-server	NULL	NULL	NULL
IP address : 192.168.1.62	8001/tcp	open	vcom-tunnel?	NULL	NULL	NULL
IP address : 192.168.1.62	8002/tcp	open	ssl/teradataordbms?	NULL	NULL	NULL
IP address : 192.168.1.62	8080/tcp	open	http	NULL	NULL	NULL
IP address : 192.168.1.62	9080/tcp	open	http	NULL	NULL	NULL
IP address : 192.168.1.1	21/tcp	filtered	ftp	NULL	NULL	NULL
IP address : 192.168.1.1	22/tcp	filtered	ssh	NULL	NULL	NULL
IP address : 192.168.1.1	23/tcp	filtered	telnet	NULL	NULL	NULL
IP address : 192.168.1.1	53/tcp	open	domain	NULL	NULL	NULL
IP address : 192.168.1.1	80/tcp	open	ssl/http	NULL	NULL	NULL
IP address : 192.168.1.1	443/tcp	filtered	https	NULL	NULL	NULL

รูปที่ 4.10 ฐานข้อมูลใน PhpMyAdmin

4.3.1.3 สถานะ Create table to Database

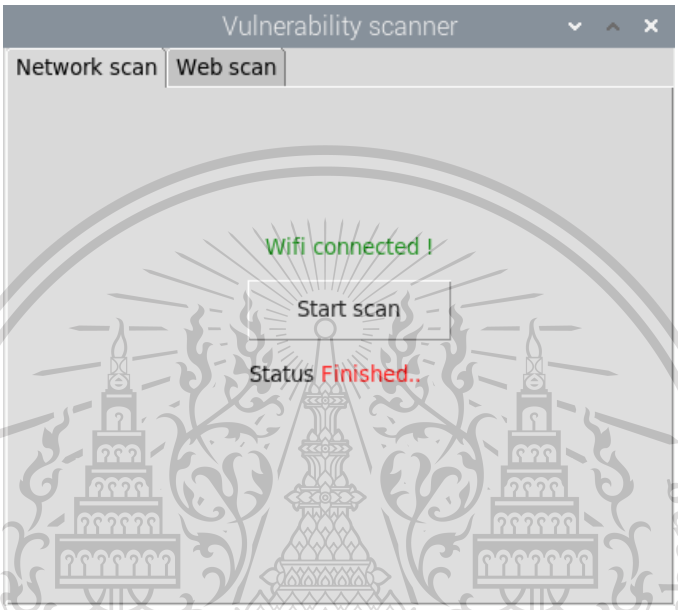
สถานะ Create table to Database แสดงดังรูปที่ 4.11 แสดงให้เห็นว่ากำลังนำข้อมูลเข้าฐานข้อมูล



รูปที่ 4.11 สถานะ Create table to Database ของการค้นหาแบบ Network Scan

4.3.1.4 สถานะ Finishing

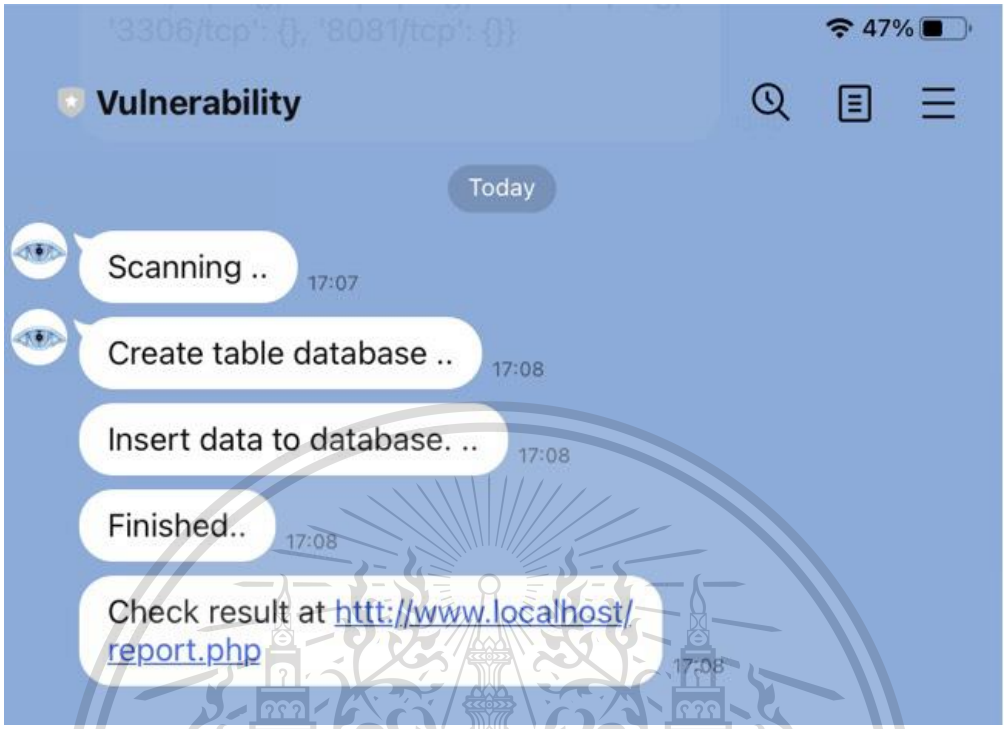
สถานะ Finish แสดงดังรูปที่ 4.12 แสดงสถานะเสร็จสิ้น



รูปที่ 4.12 สถานะ Finishing ของการค้นหาแบบ Network Scan

4.3.1.5 สถานะแจ้งเตือนผ่านไลน์ OA

สถานะแจ้งเตือน แสดงดังรูปที่ 4.13 แสดงให้เห็นว่าระบบทำงานเสร็จสิ้นเรียบร้อยหลังจากนั้นระบบจะทำการแจ้งเตือนไปใน Line OA Vulnerability



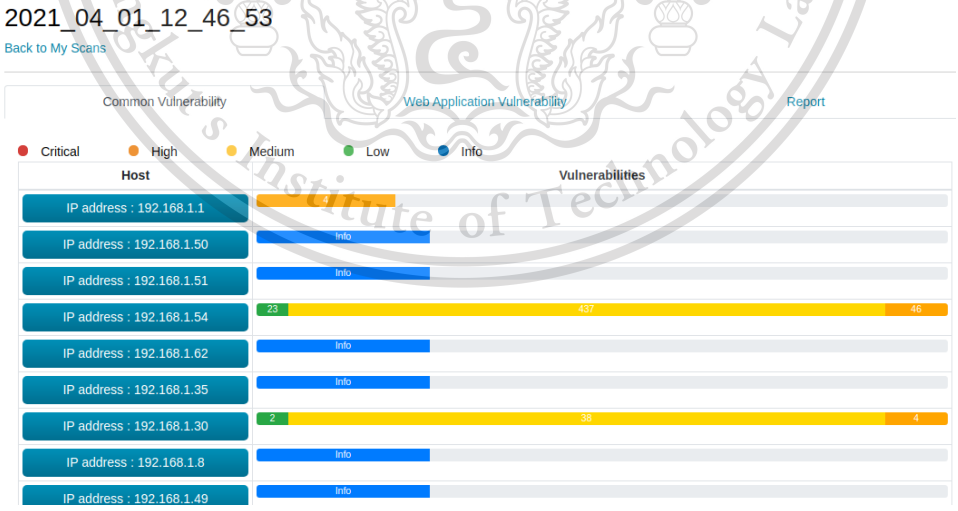
รูปที่ 4.13 สถานะแจ้งเตือนของการค้นหาแบบ Network Scan

4.3.1.6 การดูผลการค้นหาช่องโหว่ผ่านเว็บแอปพลิเคชัน

จาก URL ที่ได้หลังจากการค้นหาช่องโหว่เสร็จสิ้น เมื่อเข้าไป URL ดังกล่าวจะแสดงผลการค้นหาช่องโหว่ทั้งหมดที่เคยค้นหาในอดีตดังรูปที่ 4.14 โดยมีการค้นหาที่ได้ค้นหาไว้คือ 2021_04_01_12_46_53 ซึ่งเป็นชื่อที่ผู้ใช้งานทำการตั้งไว้และมีวันเวลาที่เริ่มทำการค้นหาช่องโหว่ต่อท้าย

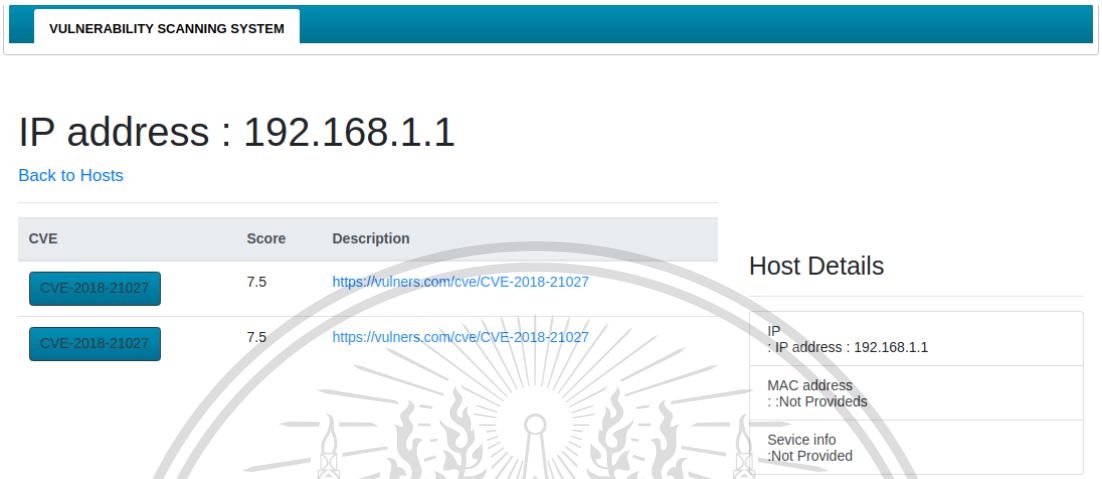
รูปที่ 4.14 หน้าเว็บแอปพลิเคชันแสดงผลการค้นหาแบบ Web Application Vulnerability

เมื่อทำการกดเข้าไปดูผลกาค้นหาช่องโหว่ที่ชื่อ 2021_04_01_12_46_53 จะแสดงชื่อการค้นหาช่องโหว่หมายเลขไอพีคือ 192.168.1.54 ,192.168.1.30 และ192.168.1.1 และมีแถบแสดงความรุนแรงของช่องโหว่ในแต่ละหมายเลขไอพีดังรูปที่ 4.15



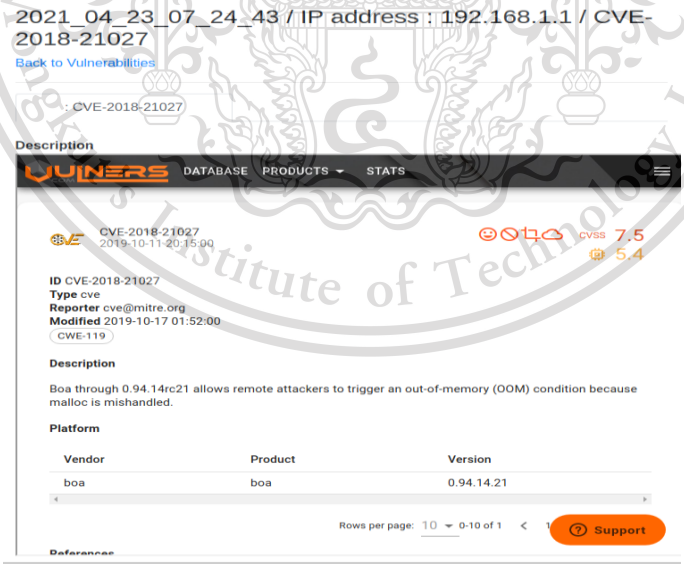
รูปที่ 4.15 หมายเลขไอพีที่มีช่องโหว่ของการค้นหาแบบ Network scan

เมื่อกดเข้าไปในหมายเลขไอพี 192.168.1.1 จะแสดงรายละเอียดของหมายเลขไอพี และบอกช่องโหว่ที่พบเป็นหมายเลข CVE ระดับความรุนแรง และคำอิบายหมายเลข CVE-2018-21027 ค่าคะแนนความรุนแรง 7.5 ดังรูปที่ 4.16



รูปที่ 4.16 ช่องโหว่ที่พบของการค้นหาแบบ Network scan

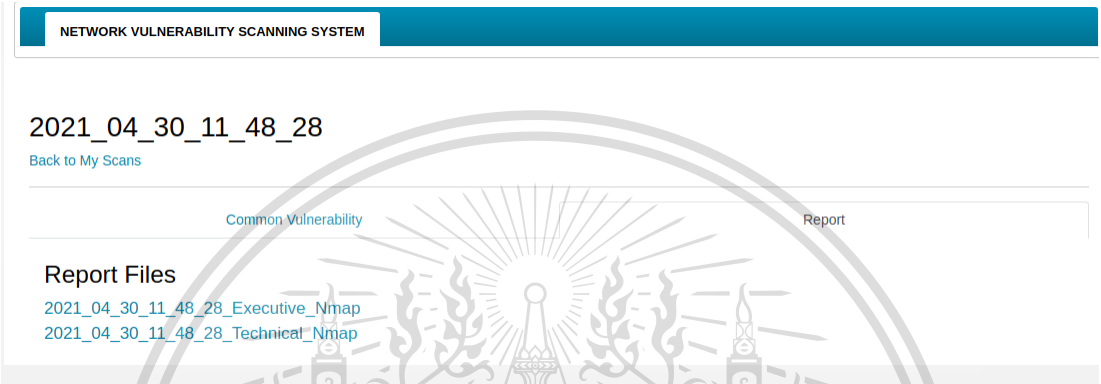
เมื่อกดเข้าไปยังหมายเลข CVE-2018-21027 จะพบรายละเอียดของหมายเลข CVE-2018-21027 คำอธิบาย , ระดับความรุนแรง MEDIUM , ค่าคะแนนความรุนแรง 5.0 แหล่งอ้างอิง คำแนะนำ ที่เป็น URL ดังรูปที่ 4.17



รูปที่ 4.17 รายละเอียดของ CVE ที่พบของการค้นหาแบบ Network Scan

4.3.1.7 รายงานผลช่องโหว่ในรูปแบบไฟล์ PDF

ในการดูรายงานผลการค้นหาช่องโหว่ในรูปแบบไฟล์ PDF มี 2 รูปแบบคือ Executive และ Technical โดยสามารถเข้าไปดูผลได้ทางเว็บแอปพลิเคชันของการรายงานผลการค้นหาช่องโหว่ครั้งนั้นๆ ดังรูปที่ 4.18



รูปที่ 4.18 รายงานผลช่องโหว่รูปแบบ PDF

เมื่อทำการกดเข้าไปดูไฟล์รายงาน Executive แสดงดังรูปที่ 4.19 โดยข้อมูลที่แสดงเป็น หมายเลขเครือข่าย 192.168.1.81/24 , เวลาที่เครื่องมือใช้ค้นหา 121.87 วินาที , ชื่อการค้นหา 2021_04_30_11_48_28 , จำนวนเครื่องที่พบ 2 เครื่อง , เวลาที่เริ่มทำงาน 30/04/2021 เวลา 11:48:28 เวลาที่สิ้นสุดการทำงาน 30/04/2021 เวลา 11:50:30 ช่องโหว่ที่พบทั้งหมด 758 ช่องโหว่ แบ่งเป็นระดับต่ำ 34 ช่องโหว่ ระดับปานกลาง 646 ช่องโหว่ ระดับสูง 78 ช่องโหว่ ระดับร้ายแรง 0 ช่องโหว่ และแสดงเป็นแผนภาพวงกลมบอกความร้ายแรงของช่องโหว่ที่ค้นพบ

Executive Summary Nmap

Nmap: Version 7.8 Cover IP,Port,Service,CVE,CVSS,OS,MAC

Network ID: 192.168.1.81

Name of scan: 2021_04_30_11_48_28

Nmap done: 256 IP addresses (6 hosts up) scanned in 121.87 seconds

Start Scan: 30/04/2021 11:48:28

Stop Scan: 30/04/2021 11:50:30

Total Vulnerability 758

none 0

low 34

medium 646

high 78

critical 0

รูปที่ 4.19 รายงาน Executive Network scan

เมื่อทำการกดเข้าไปดูไฟล์รายงาน Technical แสดงดังรูปที่ 4.20 โดยข้อมูลที่แสดงเป็นหมายเลข CVE ที่พบในแต่ละหมายเลขไอพีในแต่ละพอร์ต

Technical Nmap

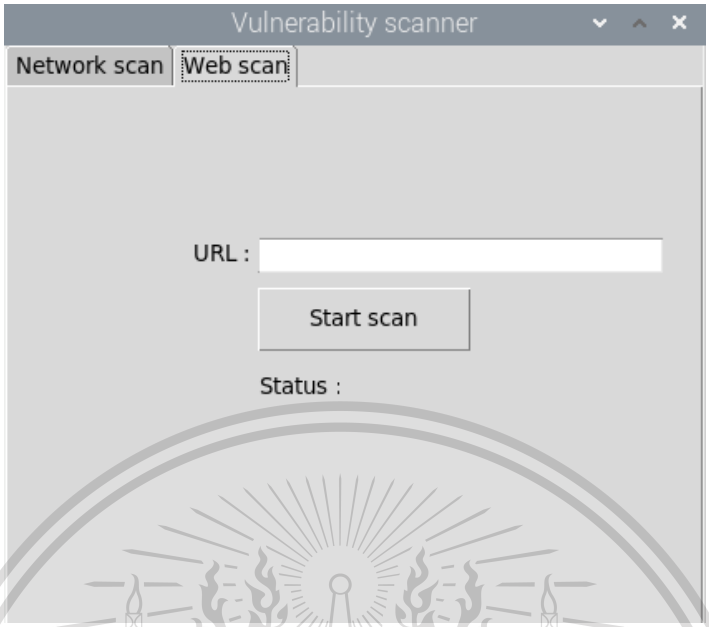
IP address : 192.168.1.54

PORT	SERVICE		
22/tcp	open		
	CVE-2019-6111	5.8	Medium
	CVE-2019-16905	4.4	Medium
	CVE-2020-14145	4.3	Medium
	CVE-2019-6110	4.0	Medium
	CVE-2019-6109	4.0	Medium
	CVE-2018-20685	2.6	Low
80/tcp	open		
	CVE-2020-11984	7.5	High
	CVE-2019-0211	7.2	High
	CVE-2019-10082	6.4	Medium
	CVE-2019-10097	6.0	Medium
	CVE-2019-0217	6.0	Medium
	CVE-2019-0215	6.0	Medium
	CVE-2020-1927	5.8	Medium
	CVE-2019-10098	5.8	Medium
	CVE-2020-9490	5.0	Medium

รูปที่ 4.20 รายงาน Technical Network scan

4.3.2 การค้นหาช่องโหว่แบบ Web scan

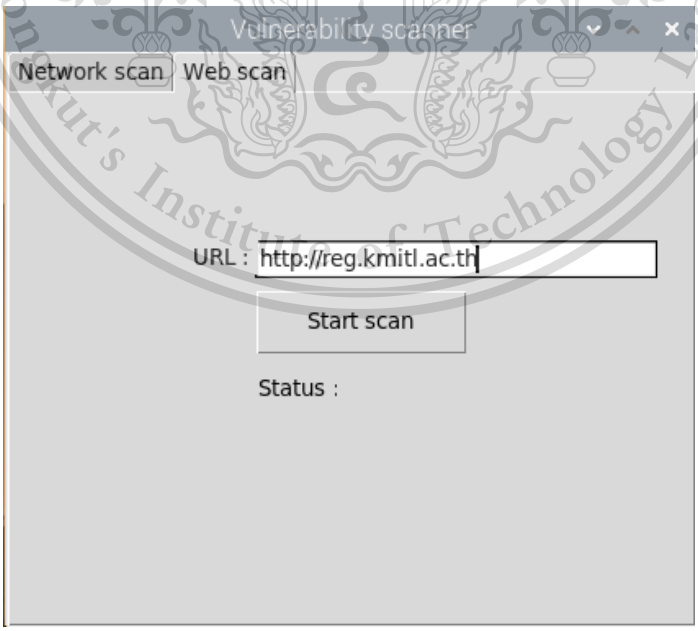
การค้นหาช่องโหว่แบบ Web Scan เป็นการค้นหาช่องโหว่โดยใช้เครื่องมือ Sitadel เพื่อทำการค้นหาช่องโหว่ ดังรูปที่ 4.21



รูปที่ 4.21 เลือกการค้นหาช่องโหว่แบบ Web Scan

4.3.2.1 ใส่ URL ที่ต้องการสแกน

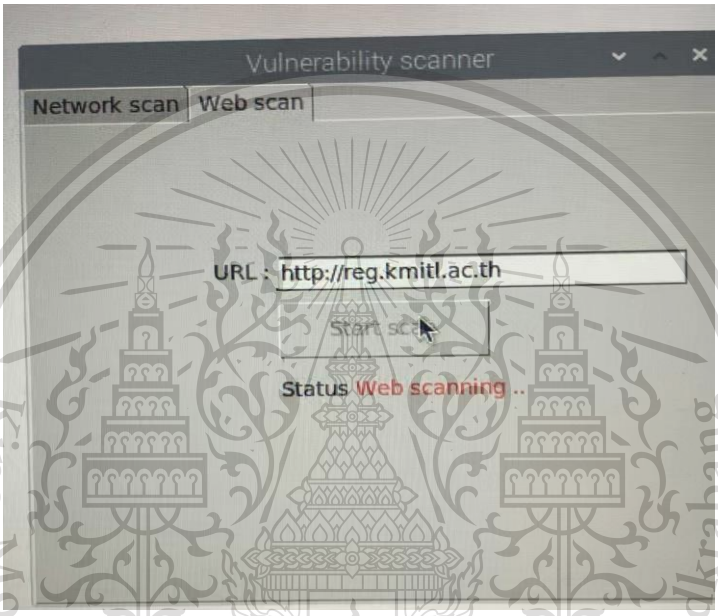
การค้นหาช่องโหว่แบบ Web Scan หลังจากที่เลือกการค้นหาทำการใส่ URL ที่ต้องการสแกน ดังรูปที่ 4.22



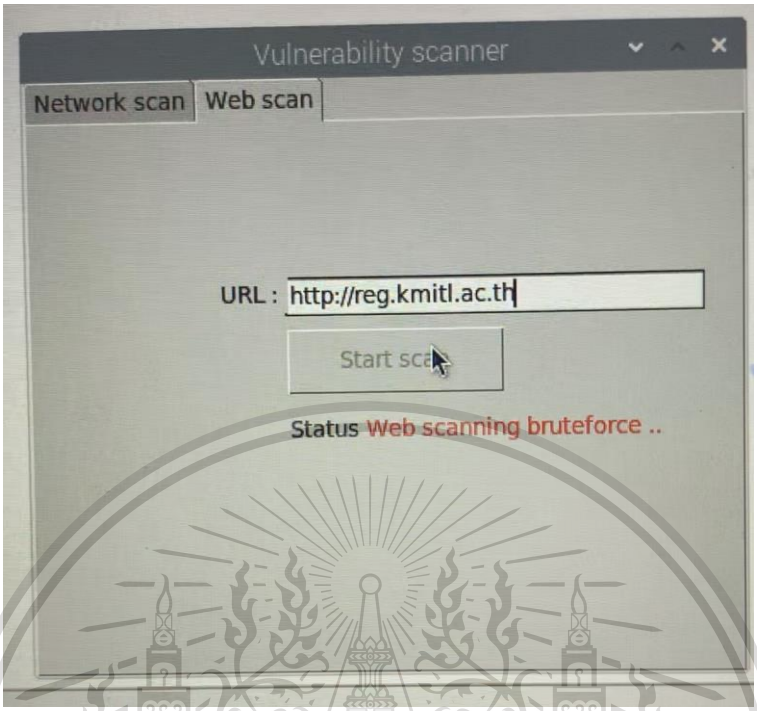
รูปที่ 4.22 ใส่ URL ลงในการสแกนแบบ Web Scan

4.3.2.2 สถานะ Web Scanning

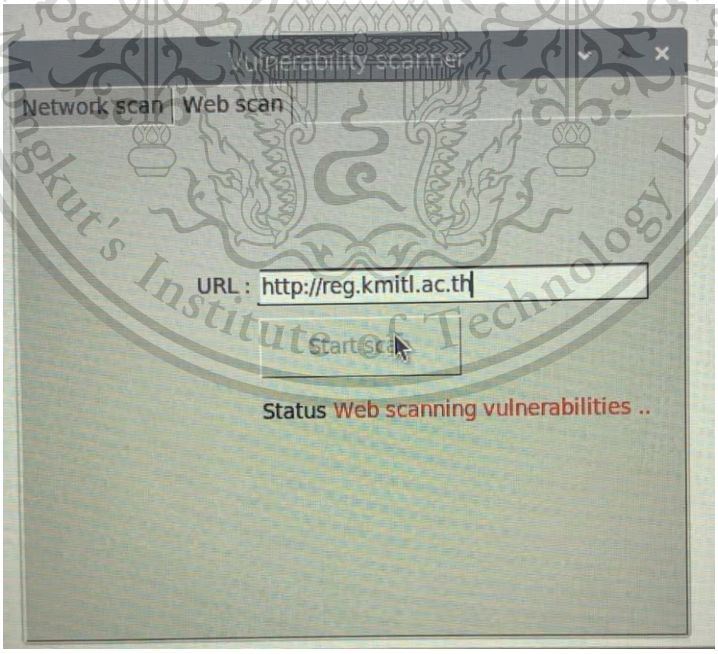
แสดงดังรูปที่ 4.23 เป็นสถานะที่แสดงว่ากำลังค้นหาเว็ป URL ที่ใส่ไป ในภาพใส่เป็น URL ของ สถาบัน หลังจากพบเวป URL ดังกล่าวจะเริ่มทำการโจมตีหรือสแกน URL นั้นทั้ง 3 รูปแบบได้แก่ bruteforce, vulnerabilities , injection ขึ้นเป็นสถานะใน GUI ดังรูปที่ 4.24, 4.25, 4.26 หลังจากสแกนเสร็จจะนำข้อมูลเก็บไว้บนฐานข้อมูลดังรูปที่ 4.27



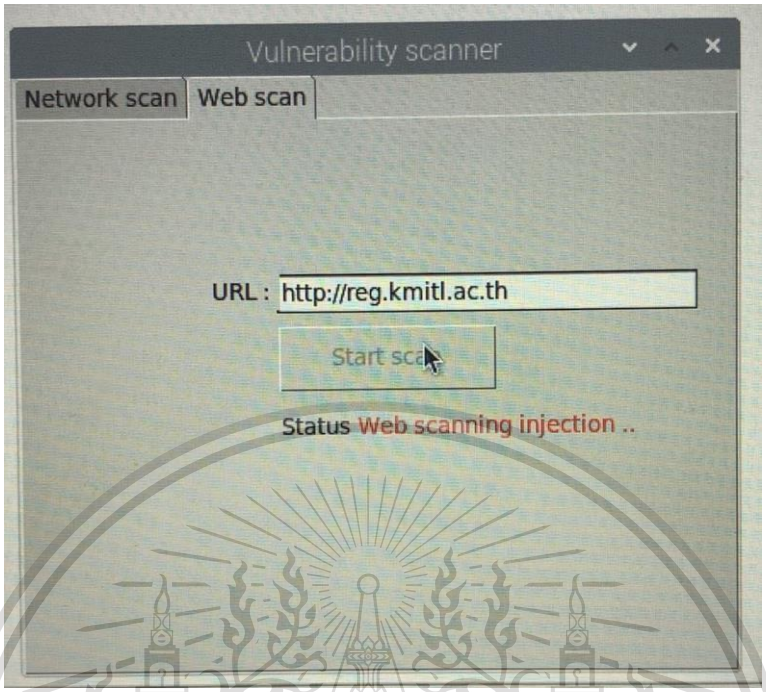
รูปที่ 4.23 สถานะ Web Scanning ของการสแกนแบบ Web Scan



รูปที่ 4.24 สถานะ Web Scanning bruteforce



รูปที่ 4.25 สถานะ Web Scanning vulnerabilities



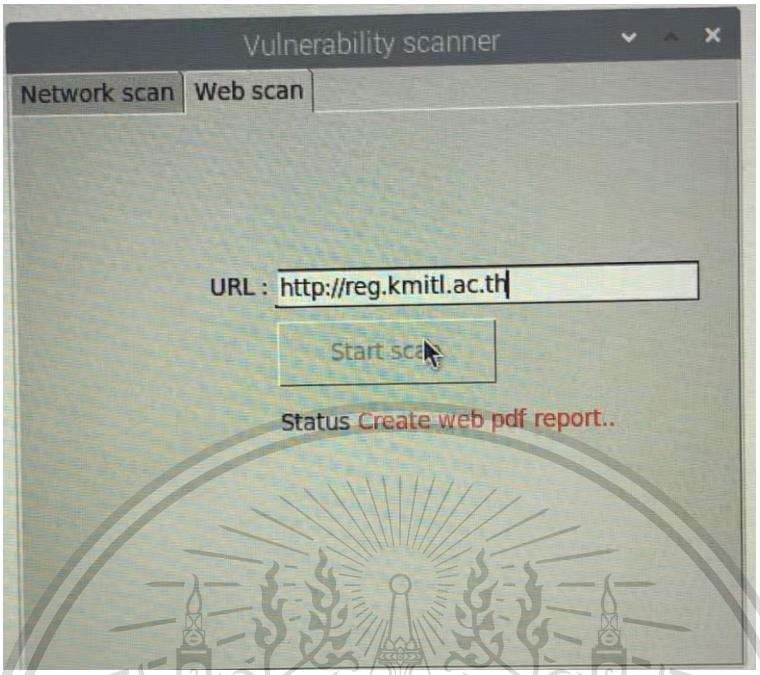
รูปที่ 4.26 สถานะ Web Scanning injection

ID	ATTACK	SCAN
2021_04_22_23_41_19	bruteforce	22-Apr-21 23:41:21 - scrapy.utils.log - INFO - Scr...
2021_04_22_23_41_19	vulnerabilities	22-Apr-21 23:41:26 - scrapy.utils.log - INFO - Scr...
2021_04_22_23_41_19	injection	22-Apr-21 23:41:32 - scrapy.utils.log - INFO - Scr...

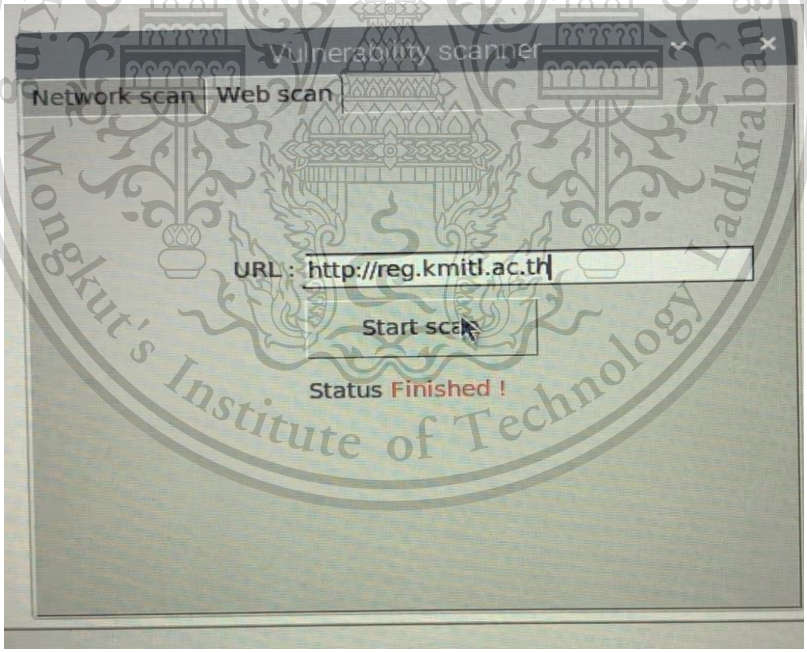
รูปที่ 4.27 ฐานข้อมูลใน PhpMyAdmin

4.3.2.2 สถานะ Create web pdf และ สถานะ finished

เมื่อ Sitadel ทำการโจมตีหรือสแกนทั้ง 3 รูปแบบเสร็จสิ้นจะทำการสร้างรูปแบบรายงานเป็นไฟล์ pdf และขึ้นสถานะสำเร็จ ดังรูปที่ 4.28และ 4.29



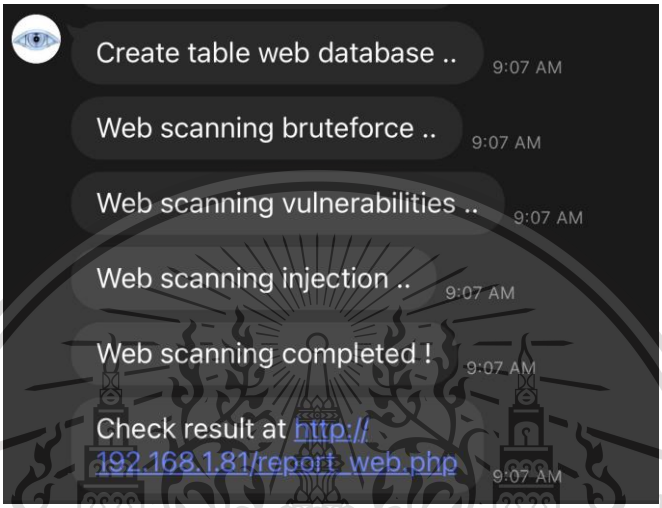
รูปที่ 4.28 สถานะ Create web pdf report



รูปที่ 4.29 สถานะ Finished

4.3.2.3 สถานะแจ้งเตือนผ่านไลน์ OA

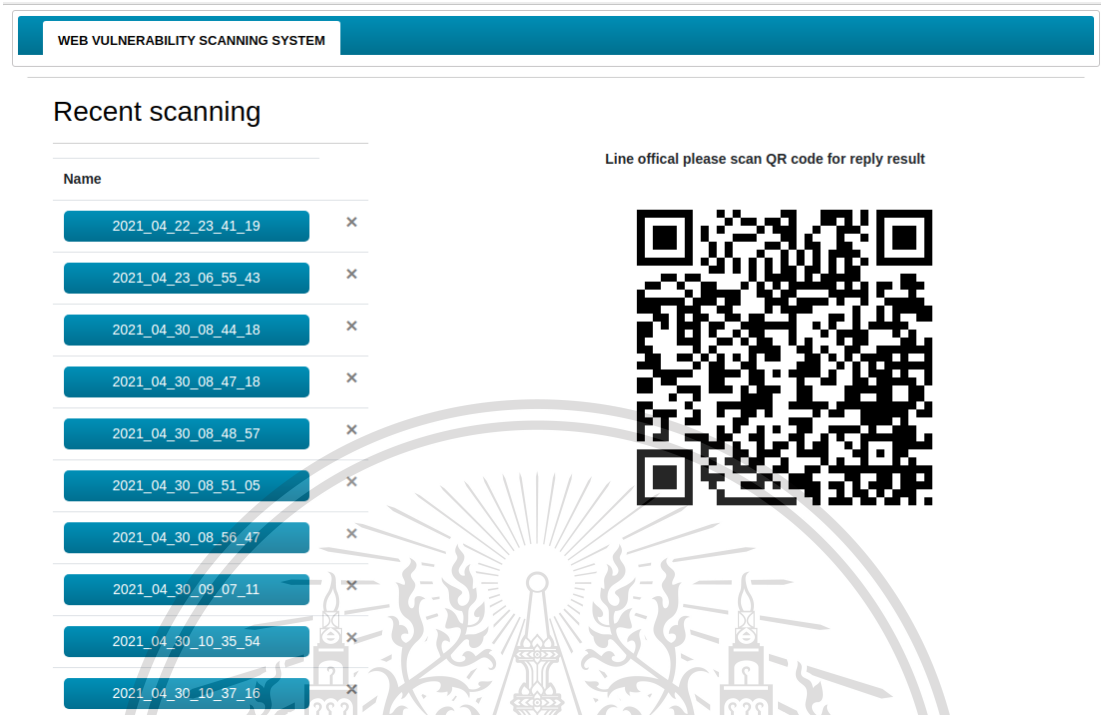
สถานะแจ้งเตือน แสดงดังรูปที่ 4.30 แสดงให้เห็นว่าระบบทำงานเสร็จสิ้นเรียบร้อยหลังจากนั้นระบบจะทำการแจ้งเตือนไปใน Line OA Vulnerability



รูปที่ 4.30 สถานะแจ้งเตือนของการค้นหาแบบ Web scan

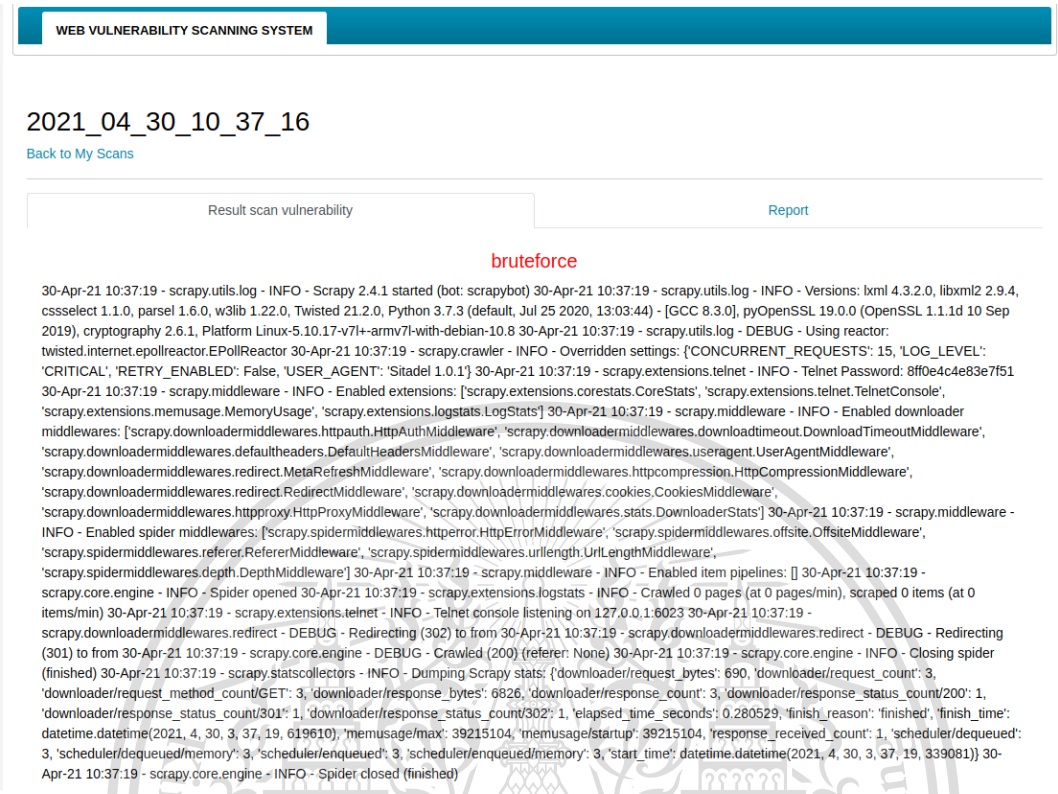
4.3.2.4 การดูผลการค้นหาช่องโหว่ผ่านเว็บแอปพลิเคชัน

จาก URL ที่ได้หลังจากการค้นหาช่องโหว่เสร็จสิ้น เมื่อเข้าไป URL ดังกล่าวจะแสดงผลการค้นหาช่องโหว่ทั้งหมดที่เคยค้นหาในอดีตดังรูปที่ 4.31 โดยมีการค้นหาที่ได้ค้นหาไว้คือ 2021_04_30_10_37_16 ซึ่งเป็นชื่อที่ผู้ใช้งานทำการตั้งไว้และมีวันเวลาที่เริ่มทำการค้นหาช่องโหว่ต่อท้าย



รูปที่ 4.31 หน้าเว็บแอปพลิเคชันแสดงผลการค้นหาแบบ Web Application Vulnerability

เมื่อทำการกดเข้าไปดูผลกาค้นหาช่องโหว่ที่ชื่อ 2021_04_30_10_37_16 จะแสดงผลการสแกนทั้ง 3 แบบ ดังรูป 4.32,4.33 และ 4.34



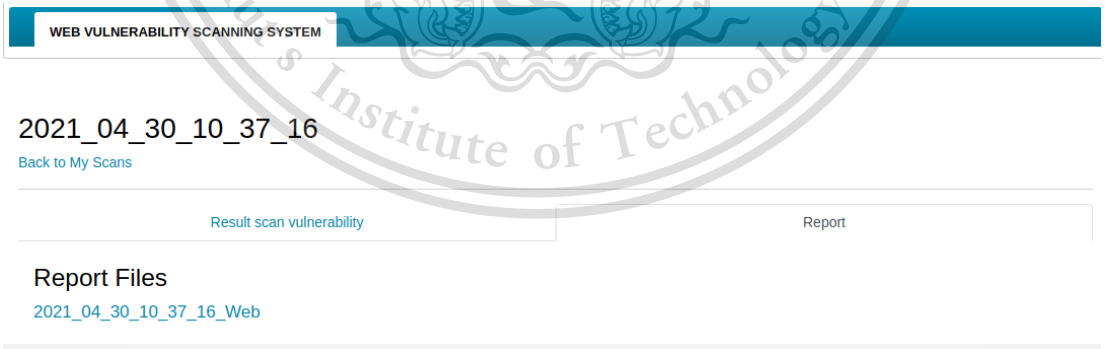
รูปที่ 4.32 ผลการสแกนแบบ bruteforce บนเว็ปแอพพลิเคชั่น

รูปที่ 4.33 ผลการสแกนแบบ vulnerabilities บนเว็ปแอพพลิเคชั่น

รูปที่ 4.34 ผลการสแกนแบบ injection บนเว็บแอพพลิเคชั่น

4.3.2.5 รายงานผลช่องโหว่ในรูปแบบไฟล์ PDF

ในการดูรายงานผลการค้นหาช่องโหว่ในรูปแบบไฟล์ PDF มี 1 รูปแบบ โดยสามารถเข้าไปดูผลได้ทางเว็บแอปพลิเคชันของการรายงานผลการค้นหาช่องโหว่ครั้งนั้นๆ ดังรูปที่ 4.35



รูปที่ 4.35 รายงานผลช่องโหว่รูปแบบ PDF

เมื่อทำการกดเข้าไปดูไฟล์รายงาน ชื่อการค้นหา 2021_04_30_10_37_16_Web แสดงดังรูปที่ 4.36,4.37 และ 4.38 โดยข้อมูลที่แสดงเป็นผลสแกนทั้ง 3 แบบได้แก่ bruteforce, vulnerabilities , injection

2021_04_30_10_37_16

Bruteforce

30-Apr-21 10:37:19 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot)

30-Apr-21 10:37:19 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0,

30-Apr-21 10:37:19 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor

30-Apr-21 10:37:19 - scrapy.crawler - INFO - Overridden settings:

{'CONCURRENT_REQUESTS': 15,

'LOG_LEVEL': 'CRITICAL',

'RETRY_ENABLED': False,

'USER_AGENT': 'Sitadel 1.0.1'}

30-Apr-21 10:37:19 - scrapy.extensions.telnet - INFO - Telnet Password: 8ff0e4c4e83e7f51

30-Apr-21 10:37:19 - scrapy.middleware - INFO - Enabled extensions:

['scrapy.extensions.corestats.CoreStats',

'scrapy.extensions.telnet.TelnetConsole',

'scrapy.extensions.memusage.MemoryUsage',

'scrapy.extensions.logstats.LogStats']

30-Apr-21 10:37:19 - scrapy.middleware - INFO - Enabled downloader middlewares:

['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',

'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',

รูปที่ 4.36 รายงานผลของ Web scan Bruteforce

Vulnerabilities

```
30-Apr-21 10:37:21 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot)
30-Apr-21 10:37:21 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0,
30-Apr-21 10:37:21 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor
30-Apr-21 10:37:21 - scrapy.crawler - INFO - Overridden settings:
{'CONCURRENT_REQUESTS': 15,
 'LOG_LEVEL': 'CRITICAL',
 'RETRY_ENABLED': False,
 'USER_AGENT': 'Sitadel 1.0.1'}
30-Apr-21 10:37:21 - scrapy.extensions.telnet - INFO - Telnet Password: 5cd945daf8709df9
30-Apr-21 10:37:21 - scrapy.middleware - INFO - Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.memusage.MemoryUsage',
 'scrapy.extensions.logstats.LogStats']
30-Apr-21 10:37:21 - scrapy.middleware - INFO - Enabled downloader middlewares:
['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
```

รูปที่ 4.37 รายงานผลของ Web scan vulnerabilities

Injection

```
30-Apr-21 10:37:24 - scrapy.utils.log - INFO - Scrapy 2.4.1 started (bot: scrapybot)
30-Apr-21 10:37:24 - scrapy.utils.log - INFO - Versions: lxml 4.3.2.0, libxml2 2.9.4, cssselect 1.1.0, parsel 1.6.0, w3lib 1.22.0,
30-Apr-21 10:37:24 - scrapy.utils.log - DEBUG - Using reactor: twisted.internet.epollreactor.EPollReactor
30-Apr-21 10:37:24 - scrapy.crawler - INFO - Overridden settings:
{'CONCURRENT_REQUESTS': 15,
 'LOG_LEVEL': 'CRITICAL',
 'RETRY_ENABLED': False,
 'USER_AGENT': 'Sitadel 1.0.1'}
30-Apr-21 10:37:24 - scrapy.extensions.telnet - INFO - Telnet Password: dea7a0b7f6c1e81c
30-Apr-21 10:37:24 - scrapy.middleware - INFO - Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.memusage.MemoryUsage',
 'scrapy.extensions.logstats.LogStats']
30-Apr-21 10:37:24 - scrapy.middleware - INFO - Enabled downloader middlewares:
['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
```

รูปที่ 4.38 รายงานผลของ Web scan injection

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อทำการค้นหาช่องโหว่ภายในระบบเครือข่ายขององค์กรหรือบริษัทที่นำไปสู่ภัยคุกคามอันไม่พึงประสงค์ โดยผู้จัดทำได้ทำการแบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ การศึกษาเครื่องมือที่ใช้ในการค้นหาช่องโหว่รูปแบบต่างๆ ในระบบเครือข่ายที่ใช้งานในระบบปฏิบัติการ Linux และทำการเขียนโปรแกรมลงบน Raspberry Pi เพื่อทำการค้นหาช่องโหว่อัตโนมัติและนำไปแสดงผลผ่านเว็บแอปพลิเคชัน

จากการศึกษาข้อมูลและจัดทำในภาคเรียนที่ 1 สามารถค้นหาช่องโหว่ได้แล้วโดยใช้ โปรแกรม Nmap แต่ยังไม่สามารถใช้งานโปรแกรม Metasploit ได้เนื่องจากมีความซับซ้อนมากกว่าและอยู่ในช่วงศึกษาทำความเข้าใจจึงทำให้ยังไม่สามารถออกแบบหน้าเว็บและรายงานการบันทึกผลได้

จากการศึกษาข้อมูลและจัดทำในภาคเรียนที่ 2 ได้ทำการปรับปรุงการสแกนช่องโหว่แบบ Nmap ให้สามารถใช้งานได้อย่างสะดวกยิ่งขึ้นและสามารถแจ้งเตือนผ่าน Line Official Account ได้ แต่ในส่วนโปรแกรม Metasploit เกิดปัญหาเนื่องจาก server ปิดให้บริการ ทำให้ต้องเปลี่ยนโปรแกรมที่ใช้เป็น Sitadel ในการสแกน Web Application

5.2 ข้อเสนอแนะ

การค้นหาช่องโหว่ในเครือข่ายในเครือข่ายแลช่องโหว่บนเว็ปแอพลิเคชั่นมีเครื่องมือช่วยในการค้นหาอีกมากมาย คณะผู้จัดทำใช้เพียง Nmap และ Sitadel ซึ่งสามารถใช้เครื่องมือที่แตกต่างกันได้ถ้าผู้ใช้งานมีเวลาศึกษาเครื่องมือที่มากขึ้นมีฟังก์ชั่นที่มีประสิทธิภาพมากกว่านี้

บรรณานุกรม

[1] “ระบบเครือข่ายและความหมาย.” https://sites.google.com/site/rabbkheruxkhaylaeakhwamhmayart/khwam-hmay-khxng-lan-wan-man.

[2] “อุปกรณ์ LAN มีอะไรบ้าง.” https://www.comsiam.com/network/อุปกรณ์-lan-มีอะไรบ้าง.

[3] “มาตรฐานระบบเครือข่าย LAN.” http://natthawam.blogspot.com/p/lan-lan-topology-media-media-access.html.

[4] “IP Address.” https://sites.google.com/site/bkkweerachai0/ip-address.

[5] “Network Port Number คืออะไร” https://itkmr.blogspot.com/2016/01/network-port-number.html

[6] Gordon Fyodor Lyon. “Nmap Network Scanning.” http://nmap.org/book/.

[7] “Bypassing Firewall Rules.” https://nmap.org/book/firewall-subversion.html.

[8] “Determining Firewall Rules.” https://nmap.org/book/determining-firewall-rules.html

[9] “IDS.” http://www.spo.moph.go.th/web/dict/index.php/columarticle/55-ids.

[10] “Basic Pentesting” https://medium.com/@thapanarath.k/play-with-basic-pentesting-1-33b00cceffe9

[11] “What is a vulnerability?.” https://owasp.org/www-community/vulnerabilities/.

[12] “Metasploit คืออะไร” https://ichi.pro/th/khxmul-beuxng-tn-keiyw-kab-metasploit-28466246317861

[13] “วิธีตรวจสอบเว็บไซต์ที่โดน Hack #5.” https://sysadmin.psu.ac.th/2013/12/13/hacking-website-detection-05/.

[14] “NVD Vulnerability Severity Ratings.” https://nvd.nist.gov/vuln-metrics/cvss#.

[15] “CVSS v2 Vector.” https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator.

[16] “CVSS v3 Vector.” https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

[17] “OWASP TOP 10.” https:// owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf.

[18] “HTML.” http://www.codingbasic.com/html.html.

[19] “PHP.” https://th.wikipedia.org/wiki/ภาษาพีเอชพี.

[20] “Web Application.” https://www.ar.co.th/kp/th/560.

[21] “Apache Webserver.” https://saixiii.com/apache-webserver/

[22] “Raspberry Pi 4 Model B.”
https://www.thaieasyelec.com/raspberry-pi-4-model-b-4gb.html

[23] ”Sitadel” https://github.com/shenril/Sitadel

[24] “ทำความรู้จักกับ Injection” https://medium.com/blog-blog/ทำความรู้จักกับ-injection-cfb7dacb0bc5

[26] “การโจมตีและช่องโหว่ประเภท SQL Injection”
https://blog.tamacorp.co/sql_injection/

[27] “Brute Force Attack” https://tha.4meahc.com/what-is-brute-force-attack-50376

ภาคผนวก ก
โค้ด

1. ส่วนการทำงานของ Nmap

```
from sqlalchemy import create_engine, create_engine, and_
from sqlalchemy.orm import sessionmaker
import datetime
import vulners
import re
import time
import subprocess
from datetime import datetime
import pymysql
from fpdf import FPDF
import urllib.request
from PIL import ImageTk,Image
from bs4 import BeautifulSoup
import keyboard
import xml.etree.ElementTree as ET
import matplotlib.pyplot as plt
from PyPDF2 import PdfFileMerger
import os

vulners_api =
vulners.Vulners(api_key="7NNARV5RD6U90AQ18CKWQSSLDOAOTHCSTATMHKYZNDQ
C2DJD7VTCVAVVI0MB2QHO")

# output_path = 'scan.text'

def Scan(IP):

    with open('nmap.txt','a+') as f:
        command = 'nmap -sV -A --script nmap-vulners,vulscan --script-args
vulscandb=cve.csv ' + IP + '/24'
        subprocess.run(command, shell=True, text=True, stdout=f)
```

```
    '''
    with open('nmap.txt','a+') as f:
        command = 'nmap -sV -A --script nmap-vulners,vulscan --script-args
vulscandb=cve.csv ' + IP + '/24'
        subprocess.run(command, shell=True, text=True, stdout=f)
    '''

    '''
    with open('nmap.txt','a+') as f:
        #command = 'nmap -sV -A --script-arg=vulscan/vulscan.nse --script-args
vulscandb=vulscan/cve.csv ' + IP + '/24'
        command = 'nmap -sV -A --script vulscan/vulscan.nse ' + IP + '/24'
        subprocess.run(command, shell=True, text=True, stdout=f)
    '''


def create_table_nmap(scan_name):

connection=pymysql.connect(host="localhost",user="root",password="12345678",dat
abase="NMAP",port=3306,charset='utf8mb4') #เชื่อมต่อฐานข้อมูล
    cursor=connection.cursor()
    BuildTableSql="CREATE TABLE "+scan_name +"(IP TEXT,PORT TEXT,STATE
TEXT,SERVICE TEXT,CVE TEXT,SCORE TEXT,DES TEXT,OS TEXT,MAC TEXT)" #สร้าง
ตารางไว้บนฐานข้อมูล
    cursor.execute(BuildTableSql)
    connection.close()

def create_table_web(scan_name):

connection=pymysql.connect(host="localhost",user="root",password="12345678",dat
abase="Sitadel",port=3306,charset='utf8mb4') #เชื่อมต่อฐานข้อมูล
    cursor=connection.cursor()
```

```
    BuildTableSql="CREATE TABLE "+scan_name+"(IP TEXT,NAME TEXT,SEVERITY
TEXT,DES TEXT,LONGDES TEXT,FIX TEXT,REF TEXT,OS TEXT)" #สร้างตารางไว้บน
ฐานข้อมูล
    cursor.execute(BuildTableSql)
    connection.close()


def insert_data_nmap(scan_name):

    with open("nmap.txt") as openfile:
        for line in openfile:
            f=open("nmap_filtered.txt","a+")
            x=line.split()
            if "Nmap scan report for" in line:
                x=x[-1].replace("(","")
                x=x.replace(")","")
                f.write("IP address : " + x)
                f.write('\n')
            if ("/tcp" in line) and (not "|" in line):
                str = re.sub('\s+',',',line)
                f.write("PORT_STATE_SERVICE : " + str + '\n')
            #if "vulners:" in line:
                #f.write(line)
            #if "vulscan:" in line:
                #f.write(line)
            if "CVE-" in line:
                #print(line)
                #cve=line.split(",",2)[0].replace("| ","")
                #print(cve)
                #des=line.split(",",2)[2]
                '''
                with open("score.txt","r") as openfiles:
                    for line in openfiles:
                        if cve==line.split(",")[0]:
```

```
                    score=line.split(",")[1].rstrip("\n")
                result=cve+","+score+","+des
                '''
                str = re.sub('\s+',',',line)
                f.write("SCVE :" + str + '\n')

            #if "[CVE" in line:
             #   f.write(line)

            if "MAC Address:" in line:
                f.write(re.sub('\s+',',',line) + '\n')
                #f.write('\n')
            if "Service Info:" in line:
                f.write(re.sub('\s+',',',line) + '\n')
                #f.write('\n')
            if not line.strip():
                f.write('\n')
                f.close()


connection=pymysql.connect(host="localhost",user="root",password="12345678",dat
abase="NMAP",port=3306,charset='utf8mb4')
    cursor=connection.cursor()

    with open("nmap_filtered.txt") as openfile: #ส่งข้อมูลขึ้นไปเก็บไว้ที่ฐานข้อมูล
        for line in openfile:
            if "IP address" in line:
                ip=line.strip()

            if "SCVE :|,CVE" in line:
                cve=line.split(",")[1]
                score=line.split(",")[2]
                des=line.split(",")[3]
```

```
                insert1 = "INSERT INTO
"+scan_name+"(IP,PORT,STATE,SERVICE,CVE,SCORE,DES)
VALUES(%s,%s,%s,%s,%s,%s,%s)"
                cursor.execute(insert1,(ip,port,state,service,cve,score,des))
            if "PORT_STATE_SERVICE" in line:
                port=line.split(',')[0].split(':')[1].strip()
                state=line.split(',')[1]
                service=line.split(',')[2]
                insert1 = "INSERT INTO "+scan_name+"(IP,PORT,STATE,SERVICE)
VALUES(%s,%s,%s,%s)"
                cursor.execute(insert1,(ip,port,state,service))

            if "Service Info:" in line:
                os=line.split(",")[2]
                #os=os.split(",")[2]
                insert1 = "INSERT INTO "+scan_name+"(IP,OS) VALUES(%s,%s)"
                cursor.execute(insert1,(ip,os))
            if "MAC Address:" in line:
                m=line.replace("MAC Address: ","")
                insert1 = "INSERT INTO "+scan_name+"(IP,MAC) VALUES(%s,%s)"
                cursor.execute(insert1,(ip,m.rstrip()))
    connection.commit()
    connection.close()

def executivenmap(nameofscan,networkid,startscan,stopscan): #สร้างรีพอร์ต
    x=startscan.split("_")
    dmy=x[2]+"/"+x[1]+"/"+x[0]
    time=x[3]+":"+x[4]+":"+x[5]
    startscan=dmy+" "+time

    y=stopscan.split("_")
    dmy=y[2]+"/"+y[1]+"/"+y[0]
    time=y[3]+":"+y[4]+":"+y[5]
    stopscan=dmy+" "+time
```

```
with open('nmap.txt','r') as f:
    for line in f:
        if "Nmap done" in line:
            total=line
countcve=0
with open('nmap_filtered.txt','r') as f:
    for line in f:
        if "SCVE :|,CVE" in line:
            countcve=countcve+1
none=0
low=0
medium=0
high=0
critical=0
cvenone=[]
cvelow=[]
cvemedium=[]
cvehigh=[]
cvecritical=[]
with open('nmap_filtered.txt','r') as f:
    for line in f:
        if "SCVE :|,CVE" in line:

            if float(line.split(",")[2])==0.0:
                cvenone.append(line.split(",")[1])
                none=none+1
            elif float(line.split(",")[2])>=0.1 and float(line.split(",")[2])<=3.9:
                cvelow.append(line.split(",")[1])
                low=low+1
            elif float(line.split(",")[2])>=4.0 and float(line.split(",")[2])<=6.9:
                cvemedium.append(line.split(",")[1])
                medium=medium+1
            elif float(line.split(",")[2])>=7.0 and float(line.split(",")[2])<=8.9:
```

```
                cvehigh.append(line.split(",")[1])
                high=high+1
            elif float(line.split(",")[2])>=9.0 and float(line.split(",")[2])<=10.0:
                cvecritical.append(line.split(",")[1])
                critical=critical+1
    pdf = PDF()
    pdf.set_font("Arial", size=12)
    pdf.add_page()
    pdf.cell(200, 10, txt="Executive Summary Nmap", ln=1, align="C")
    pdf.cell(200, 10, txt="Nmap: Version 7.8 Cover
IP,Port,Service,CVE,CVSS,OS,MAC",ln=1, align="l")
    pdf.cell(200, 10, txt="Network ID: %s"%networkid, ln=1, align="l")
    pdf.cell(200, 10, txt="Name of scan: %s"%nameofscan, ln=1, align="l")

    pdf.cell(200, 10, txt="%s"%total, ln=1, align="l")
    pdf.cell(200, 10, txt="Start Scan: %s"%startscan, ln=1, align="l")
    pdf.cell(200, 10, txt="Stop Scan: %s"%stopscan, ln=1, align="l")
    pdf.cell(200, 10, txt="Total Vulnerability %s"%str(countcve), ln=1, align="l")
    pdf.cell(15)
    pdf.cell(200, 10, txt="none %s"%str(none), ln=1, align="l")
    pdf.cell(15)
    pdf.cell(200, 10, txt="low %s"%str(low), ln=1, align="l")
    pdf.cell(15)
    pdf.cell(200, 10, txt="medium %s"%str(medium), ln=1, align="l")
    pdf.cell(15)
    pdf.cell(200, 10, txt="high %s"%str(high), ln=1, align="l")
    pdf.cell(15)
    pdf.cell(200, 10, txt="critical %s"%str(critical), ln=1, align="l")

    total=none+low+medium+high+critical
    sizes=[]
    sizes.append(none/total*100)
    sizes.append(low/total*100)
    sizes.append(medium/total*100)
```

```
    sizes.append(high/total*100)
    sizes.append(critical/total*100)
    labels=[]
    labels.append("None"+" "+str(round(none/total*100,2))+"%")
    labels.append("Low"+" "+str(round(low/total*100,2))+"%")
    labels.append("Medium"+" "+str(round(medium/total*100,2))+"%")
    labels.append("High"+" "+str(round(high/total*100,2))+"%")
    labels.append("Critical"+" "+str(round(critical/total*100,2))+"%")
    colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral','red']
    patches, texts = plt.pie(sizes, colors=colors, shadow=True, startangle=90)
    plt.legend(patches,labels, loc="best")
    plt.axis('equal')
    plt.tight_layout()
    print(os.getcwd())
    Path = os.getcwd()
    plt.savefig(Path + '/piechartnmap.png',dpi=100)
    plt.savefig(Path + '/piechartnmap1.png',dpi=40)
    pdf.image(name=Path + "/piechartnmap.png", x=60, y=150, w=100)
    pdf.output('Executive_Nmap.pdf')
    '''
    namefile='executive_Nmap.pdf'
    pdfs = ['cover_nmap.pdf',namefile]
    merger = PdfFileMerger()
    for pdf in pdfs:
        merger.append(pdf)
    merger.write("Executive_Nmap.pdf")
    merger.close()
    '''


def technicalnmap(nameofscan, IP):
    cvenone=[]
    cvelow=[]
```

```
cvemedium=[]
cvehigh=[]
cvecritical=[]
with open('nmap_filtered.txt','r') as f:
    for line in f:
        if "SCVE :|,CVE" in line:
            if float(line.split(",")[2])==0.0:
                cvenone.append(line.split(",")[1])

            elif float(line.split(",")[2])>=0.1 and float(line.split(",")[2])<=3.9:
                cvelow.append(line.split(",")[1])

            elif float(line.split(",")[2])>=4.0 and float(line.split(",")[2])<=6.9:
                cvemedium.append(line.split(",")[1])

            elif float(line.split(",")[2])>=7.0 and float(line.split(",")[2])<=8.9:
                cvehigh.append(line.split(",")[1])

            elif float(line.split(",")[2])>=9.0 and float(line.split(",")[2])<=10.0:
                cvecritical.append(line.split(",")[1])


pdf =PDF()
pdf.set_font("Arial", size=12)
ip=IP
with open('nmap_filtered.txt','r') as f:
    for line in f:
        if "IP address" in line:
            pdf.add_page()
            pdf.cell(200, 10, txt="Technical Nmap", ln=1, align="C")
            pdf.cell(200, 10, txt=line, ln=1, align="l")
            pdf.cell(200, 10, txt="PORT     SERVICE", ln=1, align="l")
        if "PORT_STATE_SERVICE" in line:
            y=line.split(',')
            for i in y:
```

```
                if i == y[0]:
                    write=i.split(':')[1]+"      "
                elif i == y[1]:
                    write=write+" "+i
            pdf.cell(200, 10, txt=write, ln=1, align="l")
        if "SCVE :|,CVE" in line:
            y=line.split(",")
            cve=y[1]
            link=link="http://"+ip+"/detail.php?data="+cve
            for i in y:
                if i == y[1]:
                    write=i+"      "
                elif i==y[2]:
                    if float(i)==0.0:
                        write=write+" "+i+"    None"
                    elif (float(i)>=0.1) and (float(i)<=3.9):
                        write=write+" "+i+"    Low"
                    elif (float(i)>=4.0) and (float(i)<=6.9):
                        write=write+" "+i+"    Medium"
                    elif (float(i)>=7.0) and (float(i)<=8.9):
                        write=write+" "+i+"    High"
                    elif (float(i)>=9.0) and (float(i)<=10.0):
                        write=write+" "+i+"    Critical"
            pdf.cell(15)
            pdf.cell(200, 10, txt=write, ln=1, align="l",link=link)
        if "MAC Address:" in line:
            pdf.cell(200, 10, txt=line, ln=1, align="l")
        if "Service Info:" in line:
            pdf.cell(200, 10, txt=line, ln=1, align="l")
pdf.add_page()
pdf.cell(200, 10, txt="Critical", ln=1, align="C")
for line in cvecritical:
    cve=line.split(",")[0]
    with open("nmap_filtered.txt") as openfile:
```

```
        for line in openfile:
            if cve in line:
                d=line.split(",",2)[2]
                d=d.replace("\\x0D","")
                pdf.cell(200, 10, txt=cve, ln=1, align="l")
                pdf.multi_cell(0,5,txt=d,align="L")
pdf.add_page()
pdf.cell(200, 10, txt="High", ln=1, align="C")
for line in cvehigh:
    #print(line)
    cve=line.split(",")[0]
    with open("nmap_filtered.txt") as openfile:
        for line in openfile:
            if cve in line:
                d=line.split(",",2)[2]
                d=d.replace("\\x0D","")
                pdf.cell(200, 10, txt=cve, ln=1, align="l")
                pdf.multi_cell(0,5,txt=d,align="L")
pdf.add_page()
pdf.cell(200, 10, txt="Medium", ln=1, align="C")
for line in cvemedium:
    cve=line.split(",")[0]
    with open("nmap_filtered.txt") as openfile:
        for line in openfile:
            if cve in line:
                d=line.split(",",2)[2]
                d=d.replace("\\x0D","")
                pdf.cell(200, 10, txt=cve, ln=1, align="l")
                pdf.multi_cell(0,5,txt=d,align="L")
pdf.add_page()
pdf.cell(200, 10, txt="Low", ln=1, align="C")
for line in cvelow: #ดึงคำอธิบายแต่ละ
    cve=line.split(",")[0]
    with open("nmap_filtered.txt") as openfile:
```

```
            for line in openfile:
                if cve in line:
                    d=line.split(",",2)[2]
                    d=d.replace("\\x0D","")
                    pdf.cell(200, 10, txt=cve, ln=1, align="l")
                    pdf.multi_cell(0,5,txt=d,align="L")
    pdf.add_page()
    pdf.cell(200, 10, txt="NONE", ln=1, align="C")
    for line in cvenone:
        cve=line.split(",")[0]
        with open("nmap_filtered.txt") as openfile:
            for line in openfile:
                if cve in line:
                    d=line.split(",",2)[2]
                    d=d.replace("\\x0D","")
                    pdf.cell(200, 10, txt=cve, ln=1, align="l")
                    pdf.multi_cell(0,5,txt=d,align="L")
    #subprocess.run('rm nmap_filtered.txt',cwd='/root',shell=True,text=True)
    pdf.output('Technical_Nmap.pdf')

    '''
    namefile='technical_Nmap.pdf'
    pdfs = ['cover_nmap.pdf',namefile]
    merger = PdfFileMerger()
    for pdf in pdfs:
        merger.append(pdf)
    merger.write("Technical_Nmap.pdf")
    merger.close()
    '''


def managepathnmap(nameofscan):
    #os.chdir("/var/www/html/report")
    os.makedirs("/var/www/html/report/" + nameofscan)
    #os.chdir(os.getcwd())
```

```
    #print('Dir : ' + os.getcwd())
    os.system("mv Executive_Nmap.pdf %s_Executive_Nmap.pdf"%nameofscan)
    os.system("mv Technical_Nmap.pdf %s_Technical_Nmap.pdf"%nameofscan)
    os.system("mv %s_Executive_Nmap.pdf %s_Technical_Nmap.pdf
/var/www/html/report/%s"%(nameofscan,nameofscan,nameofscan))

class PDF(FPDF):
    def footer(self):
        # Go to 1.5 cm from bottom
        self.set_y(-15)
        # Select Arial italic 8
        self.set_font('Arial', 'I', 8)
        # Print centered page number
        self.cell(0, 10, 'Page %s' % self.page_no(), 0, 0, 'C')
```

2. ส่วนการทำงาน Sitadel

```
import argparse
import logging
import sys
import signal
from lib import __version__
from lib.config import settings
from lib.config.settings import Risk
from lib.request.request import SingleRequest
from lib.utils import banner, manager, output, validator
from lib.utils.container import Services
from lib.utils.datastore import Datastore
from lib.utils.output import Output
from threading import *

class Sitadel(object):
    bn = banner.Banner()
    ma = manager
```

```
    url = None

    def main(self, url_address):

        parser = argparse.ArgumentParser(
            formatter_class=argparse.ArgumentDefaultsHelpFormatter,
            usage=self.bn.banner(),
        )

        # Prepare the possible values for risk levels
        risk_values = [r.value for r in Risk]
        # Add arguments
        #parser.add_argument("url", help="URL of the website to scan",
default=url_address)
        parser.add_argument(
            "-r",
            "--risk",
            type=int,
            help="Level of risk allowed for the scan",
            choices=risk_values,
            default = 1
        )
        parser.add_argument(
            "-ua",
            "--user-agent",
            default="Sitadel " + __version__,
            help="User-agent to set for the scan requests",
        )
        parser.add_argument(
            "--redirect",
            dest="redirect",
            help="Whether or not the scan should follow redirection",
            action="store_true",
        )
```

```
        parser.add_argument(
            "--no-redirect",
            dest="redirect",
            help="Whether or not the scan should follow redirection",
            action="store_false",
        )
        parser.set_defaults(redirect=True)
        parser.add_argument(
            "-t",
            "--timeout",
            type=int,
            default=30,
            help="Timeout to set for the scan HTTP requests",
        )
        parser.add_argument(
            "-c", "--cookie", help="Cookie to set for the scan HTTP requests"
        )
        parser.add_argument(
            "-p", "--proxy", help="Proxy to set for the scan HTTP requests"
        )
        parser.add_argument(
            "-f", "--fingerprint", nargs="+", help="Fingerprint modules to activate",
default='header'
        )
        parser.add_argument(
            "-a", "--attack", nargs="+", help="Attack modules to activate", default=['vulns',
'injection', 'bruteforce']
        )
        parser.add_argument(
            "--config", help="Path to the config file", default="Sitadel/config/config.yml"
        )
        parser.add_argument(
            "-v",
            "--verbosity",
```

```
    action="count",
    default=0,
    help="Increase output verbosity",
)
parser.add_argument("--version", action="version", version=self.bn.version())
args = parser.parse_args()

# Verify the target URL
self.url = validator.validate_target(url_address)

# Reading configuration
settings.from_yaml(args.config)
if args.risk is not None:
    settings.risk = Risk(args.risk)

# Setting up the logger
logger = logging.getLogger("sitadelLog")
logging.basicConfig(
    filename="sitadel.log",
    filemode="w",
    format="%(asctime)s - %(name)s - %(levelname)s - %(message)s",
    datefmt="%d-%b-%y %H:%M:%S",
    level=(logging.CRITICAL - (args.verbosity * 10)),
)

# Create handlers
console_handler = logging.StreamHandler()
console_handler.setLevel(logging.WARNING)

file_handler = logging.FileHandler("sitadel.log")
file_handler.setLevel(level=(logging.CRITICAL - (args.verbosity * 10)))

# Create formatters and add it to handlers
    console_format = logging.Formatter("%(name)s - %(levelname)s - %(message)s")
```

```
console_handler.setFormatter(console_format)

file_format = logging.Formatter(
    "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
)
file_handler.setFormatter(file_format)

# Add handlers to the logger
logger.addHandler(console_handler)
logger.addHandler(file_handler)

# Register services
Services.register("datastore", Datastore(settings.datastore))
Services.register("logger", logger)
Services.register("output", Output())
Services.register(
    "request_factory",
    SingleRequest(
        url=self.url,
        agent=args.user_agent,
        proxy=args.proxy,
        redirect=args.redirect,
        timeout=args.timeout,
    ),
)

# Display target and scan starting time
self.bn.preamble(self.url)
try:
    # Run the fingerprint modules
    self.ma.fingerprints(
        args.fingerprint,
        self.url,
        args.cookie,
```

```
            )

            # Run the crawler to discover urls
            discovered_urls = self.ma.crawler(self.url, args.user_agent)

            # Hotfix on KeyboardInterrupt being redirected to scrapy crawler process
            signal.signal(signal.SIGINT, signal.default_int_handler)

            # Run the attack modules on discovered urls
            self.ma.attacks(args.attack, self.url, discovered_urls)
        except Exception as e:
            print(e)
        '''
        except KeyboardInterrupt:
            raise
        finally:
            self.bn.postscript()
        '''


#if __name__ == "__main__":
 #   try:
  #      Sitadel().main()
   # except KeyboardInterrupt:
    #    sys.exit(output.Output().error("Interruption by the user, Quitting..."))
```

3. ส่วน GUI

```
import nmap_scan as nmap
import numpy as np
import sys
import os
import time
import tkinter as tk
```

```
from tkinter import ttk
from tkinter import filedialog
import os
from tkinter import messagebox
from tkinter import *
import main


DisplayName = ''

def Apply():
    global DisplayName
    DisplayName = T.get().strip()
    mainWindow.destroy()
    main.run_main(DisplayName)

if __name__ == '__main__':

    mainWindow = tk.Tk()
    mainWindow.title('Vulnerability scanner')
    mainWindow.resizable(width=False, height=False)

    window_height = 320
    window_width = 400

    screen_width = mainWindow.winfo_screenwidth()
    screen_height = mainWindow.winfo_screenheight()

    x_cordinate = int((screen_width / 2) - (window_width / 2))
    y_cordinate = int((screen_height / 2) - (window_height / 2))

    mainWindow.geometry("{}x{}+{}+{}".format(window_width, window_height,
x_cordinate, y_cordinate))
```

```
    l2 = Label(mainWindow, text="Please insert displayname of line")
    l2.place(x=105, y=60, bordermode="outside")

    T = Entry(mainWindow, width=20)
    T.place(x=125, y=100, bordermode="outside")

    l = Label(mainWindow, text="Name : ")
    l.place(x=70, y=100, bordermode="outside")

    btn = ttk.Button(mainWindow, text="Apply", command=Apply)
    btn.place(x=130, y=130, bordermode="outside")

    mainWindow.mainloop()
```

4. ส่วนแป้นพิมพ์ GUI (keyboard.py)

```
import tkinter as tk

alphabets = [
    ['`','1','2','3','4','5','6','7','8','9','0','-','='],
    ['Tab','q','w','e','r','t','y','u','i','o','p','[',']'],
    ['Shift','a','s','d','f','g','h','j','k','l',';',"'","Del"],
    ['z','x','c','v','b','n','m',',','.','/',"\\","Space"],
    ['@','#','$','%','^','&','*','-','+','(',')','"',':' ],
    ['!','?','OK']

]

uppercase = False  # use uppercase chars.

def select(entry, value):
    global uppercase

    if value == "Space":
```

```
            value = ' '
        elif value == 'Enter':
            value = '\n'
        elif value == 'Tab':
            value = '\t'


        if value == "Del":
            if isinstance(entry, tk.Entry):
                entry.delete(len(entry.get())-1, 'end')
            #elif isinstance(entry, tk.Text):
            else: # tk.Text
                entry.delete('end - 2c', 'end')
        elif value in ('Cap','Shift'):
            uppercase = not uppercase # change True to False, or False to True
        else:
            if uppercase:
                value = value.upper()
            entry.insert('end', value)

    def create(root, entry):
        def destroy():
            window.destroy()

        window = tk.Toplevel(root)
        window.geometry("470x190")
        window.attributes('-fullscreen', True)

        window.wm_attributes("-alpha", 0.7)

        for y, row in enumerate(alphabets):

            x = 0
```

```
        #for x, text in enumerate(row):
        for text in row:
            if text == 'Space':
                width = 5
                columnspan =2
            else:
                width = 1
                columnspan = 1


            if text =='OK':
                tk.Button(window, text=text,height=1,width=width, command=destroy,
bg="black", fg="white").grid(row=y, column=x, columnspan=columnspan)
                x += columnspan
            else:
                tk.Button(window, text=text,height=1,width=width, command=lambda
value=text: select(entry, value), bg="black", fg="white").grid(row=y, column=x,
columnspan=columnspan)
                x += columnspan

# --- main ---
```

5. ส่วน main.py

```
        import nmap_scan as nmap
import numpy as np
import sys
import os
import time
import tkinter as tk
from tkinter import ttk
from tkinter import filedialog
import os
from tkinter import messagebox
```

```
from tkinter import *
from threading import *
import socket

from sitadel import *
from threading import *
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from PIL import ImageTk, Image
from linebot import (LineBotApi, WebhookHandler)
from linebot.models import (
 MessageEvent, TextMessage, TextSendMessage, ImageSendMessage,
 SourceUser, SourceGroup, SourceRoom,
 TemplateSendMessage, ConfirmTemplate, MessageTemplateAction,
 ButtonsTemplate, URITemplateAction, PostbackTemplateAction,
 CarouselTemplate, CarouselColumn, PostbackEvent,
 StickerMessage, StickerSendMessage, LocationMessage, LocationSendMessage,
 ImageMessage, VideoMessage, AudioMessage,
 UnfollowEvent, FollowEvent, JoinEvent, LeaveEvent, BeaconEvent
)
from linebot.exceptions import LineBotApiError
import GUI
import requests
from subprocess import check_output
import pymysql
from datetime import datetime
import os.path
from os import path


UserId = ''
Line_Name = ''
```

```
class Vulnerability_scanner():

    def __init__(self, parent, ind):

        self.TAB = parent
        self.token =
'W9N/kT4ycwSPFrzZwTqJY5REZMja06pKVTLHZcAh75ZsJG9geLB7UX3jUrQLccmuLteOl
k+zraWvULt3o6NHYFF3fYm0W177BUj2c2bXOFfe4LYrxbVUieMKV5kO+YR3WCvXkiogaK
Ug564L7oR8HwdB04t89/1O/w1cDnyilFU='

        if ind == 1:
            self.button_start = tk.Button(parent, text="Start scan",
command=self.threading_network, height=3, width=20)
            self.button_start.place(x=150, y=120, width=130, height=40)

            #self.l = Label(parent, text="IP address : ")
            #self.l.place(x=150, y=90, bordermode="outside")

            # IP address
            with open('ip.txt','w') as f:
                S = check_output(['hostname', '-I'], text=True)
                f.write(S)

            IP_address = ''
            with open('ip.txt','r') as f:
                for line in f:
                    IP_address = line.split(' ')[0]
                    break


            self.IP = IP_address

            if self.Connect_wifi():
```

```
                self.l2 = Label(parent, text='Wifi connected !', fg='green')
                self.l2.place(x=160, y=90, bordermode="outside")
            else:
                self.l2 = Label(parent, text='Wifi not connected !', fg='red')
                self.l2.place(x=160, y=90, bordermode="outside")

            self.l = Label(parent, text="Status : ")
            self.l.place(x=150, y=170, bordermode="outside")

            self.l2 = Label(parent, text="", fg='red')
            self.l2.place(x=195, y=170, bordermode="outside")
        else:
            self.web_url = StringVar()
            self.T = Entry(parent, textvariable=self.web_url, width=30)
            self.T.place(x=150, y=90, bordermode="outside")

            self.l3 = Label(parent, text="URL : ")
            self.l3.place(x=110, y=90, bordermode="outside")

            self.l = Label(parent, text="Status : ")
            self.l.place(x=150, y=170, bordermode="outside")

            self.l2 = Label(parent, text="", fg='red')
            self.l2.place(x=195, y=170, bordermode="outside")

            self.button_web_scan = tk.Button(parent, text="Start scan",
command=self.Web_scan, height=3, width=20)
            self.button_web_scan.place(x=150, y=120, width=130, height=40)

    def Connect_wifi(self):
        R = False
        url = "http://www.google.com"
        timeout = 5
        try:
```

```
            request = requests.get(url, timeout=timeout)
            print("Connected to the Internet")
            R = True
        except (requests.ConnectionError, requests.Timeout) as exception:
            print("No internet connection.")

        return R

    def generate_times_name(self):
        N = str(datetime.now().replace(microsecond=0))
        N = N.replace("-","_")
        N = N.replace(" ","_")
        N = N.replace(":","_")
        return N

    def Network_scan(self):

        if path.exists("nmap.txt"):
            os.remove("nmap.txt")

        if path.exists("filtered.txt"):
            os.remove("filtered.txt")

        global UserId, Line_Name

        scan_name = self.generate_times_name()

        # Line API
        #token =
'W9N/kT4ycwSPFrzZwTqJY5REZMja06pKVTLHZcAh75ZsJG9geLB7UX3jUrQLccmuLteOl
k+zraWvULt3o6NHYFF3fYm0W177BUj2c2bXOFfe4LYrxbVUieMKV5kO+YR3WCvXkiogaK
Ug564L7oR8HwdB04t89/1O/w1cDnyilFU='
        line_bot_api = LineBotApi(self.token)
        to = UserId
```

```
        startscan = self.generate_times_name()

        self.l2.config(text='Scanning ..')

        line_bot_api.push_message(to, TextSendMessage(text='Scanning ..'))

        self.button_start['state'] = tk.DISABLED
        nmap.Scan(self.IP)

        stopscan = self.generate_times_name()

        line_bot_api.push_message(to, TextSendMessage(text='Create table database
..'))

        self.l2.config(text='Create table database..')
        nmap.create_table_nmap(scan_name)

        line_bot_api.push_message(to, TextSendMessage(text='Insert data to database.
..'))

        self.l2.config(text='Insert data to database..')
        nmap.insert_data_nmap(scan_name)

        #line_bot_api.push_message(to, TextSendMessage(text='Show scan result ..'))

        #with open('nmap.txt') as f:
            #lines = f.readlines()

        #J = json.dumps(lines)
        #print(J)
        #print(type(J))

        #str = ""
```

```
        #for ele in lines:
            #str += ele

        #print(str)
        #print(type(str))

        #line_bot_api.push_message(to, TextSendMessage(text=J))

        self.button_start['state'] = tk.NORMAL

        line_bot_api.push_message(to, TextSendMessage(text='Finished..'))

        self.l2.config(text='Finished..')

        line_bot_api.push_message(to, TextSendMessage(text='Check result at
http://www.localhost/report.php'))

        nmap.executivenmap(scan_name, self.IP, startscan, stopscan)
        nmap.technicalnmap(scan_name, self.IP)
        nmap.managepathnmap(scan_name)


        '''
        # Chart
        fig = plt.figure(figsize=(5, 3))
        ax = fig.add_axes([0, 0, 1, 1])
        ax.axis('equal')
        langs = ['Low', 'High', 'Medium']
        students = [12.5, 9.8, 77.7]
        ax.pie(students, labels=langs, autopct='%1.2f%%')

        plt.legend(['Low', 'High', 'Medium'], loc="upper right")
        plt.savefig('plot.png')
        plt.show()
```

```
    '''

    '''
    with open('Result_network.csv', 'w') as f:
        for key, value in data.items():
            f.write("%s,%s\n" % (key, value))

    str = ''
    with open("Result_network.csv") as file_in:
        for line in file_in:
            str += line + '\r\n'

    print(str)

    self.button_start['state'] = tk.NORMAL

    line_bot_api.push_message(to, TextSendMessage(text=str))
    '''

def threading_network(self):
    # Call work function
    t1 = Thread(target=self.Network_scan)
    t1.start()

def threading_web(self):
    self.button_web_scan['state'] = tk.DISABLED
    self.l2.config(text='Web scanning ..')
    # Call work function
    t2 = Thread(target=self.Web_scan)
    t2.start()

def Web_scan(self):
    global UserId, Line_Name
```

```
        line_bot_api = LineBotApi(self.token)
        to = UserId
        scan_name = self.generate_times_name()

        line_bot_api.push_message(to, TextSendMessage(text='Create table database
..'))

        self.l2.config(text='Create table database..')
        nmap.create_table_web(scan_name)

        self.button_web_scan['state'] = tk.DISABLED
        self.l2.config(text='Web scanning ..')
        url = self.web_url.get()
        cls = Sitadel()
        Sitadel().main(url)
        self.l2.config(text='Completed !')
        self.button_web_scan['state'] = tk.NORMAL

def on_tab_selected(event, TAB1, TAB2):

    selected_tab = event.widget.select()
    tab_text = event.widget.tab(selected_tab, "text")
    if tab_text == 'Network scan' :
        Vulnerability_scanner(TAB1, 1)
    elif tab_text == 'Web scan':
        Vulnerability_scanner(TAB2, 2)



#if __name__ == '__main__':
def run_main(DisplayName):

    global UserId, Line_Name
```

```
    url = "http://www.televulnerability.com/line_API.txt"
    r = requests.get(url, stream=True)

    with open('line_API.txt', 'wb') as f:
        f.write(r.content)

    with open("line_API.txt") as file_in:
        for line in file_in:
            A = line.split('@')
            Name = A[0]
            if Name == DisplayName:
                UserId = A[1]
                Line_Name = Name


    # GUI
    mainWindow = tk.Tk()
    mainWindow.title('Vulnerability scanner')
    mainWindow.resizable(width=False, height=False)

    window_height = 350
    window_width = 420

    screen_width = mainWindow.winfo_screenwidth()
    screen_height = mainWindow.winfo_screenheight()

    x_cordinate = int((screen_width / 2) - (window_width / 2))
    y_cordinate = int((screen_height / 2) - (window_height / 2))

    mainWindow.geometry("{}x{}+{}+{}".format(window_width, window_height,
x_cordinate, y_cordinate))

    TAB_CONTROL = ttk.Notebook(mainWindow)
```

```
    TAB1 = ttk.Frame(TAB_CONTROL)
    TAB2 = ttk.Frame(TAB_CONTROL)

    TAB_CONTROL.add(TAB1, text='Network scan')
    TAB_CONTROL.add(TAB2, text='Web scan')

    TAB_CONTROL.pack(expand=1, fill="both")

    #var = StringVar()
    #T = Entry(mainWindow, textvariable = var, width = 90)
    #T.place(x=265, y=835, bordermode="outside")

    #l = Label(mainWindow, text="Result : ")
    #l.config(font=("Courier", 10))
    #l.place(x=210, y=835, bordermode="outside")

    TAB_CONTROL.bind("<<NotebookTabChanged>>", lambda event:
on_tab_selected(event, TAB1, TAB2))

    # Line API
    '''
    token =
'W9N/kT4ycwSPFrzZwTqJY5REZMja06pKVTLHZcAh75ZsJG9geLB7UX3jUrQLccmuLteOl
k+zraWvULt3o6NHYFF3fYm0W177BUj2c2bXOFfe4LYrxbVUieMKV5kO+YR3WCvXkiogaK
Ug564L7oR8HwdB04t89/1O/w1cDnyilFU='
    line_bot_api = LineBotApi(token)

    try:
        line_bot_api.push_message(to, TextSendMessage(text='Hello World!'))
    except LineBotApiError as e:
        err = e.message
        print(err)
    '''
```

```
#line_bot_api.push_message(to, TextSendMessage(text='Hello World!'))
#line_bot_api.broadcast(TextSendMessage(text='Hello World!'))

# Chart
'''
fig = plt.figure(figsize=(5, 3))
ax = fig.add_axes([0, 0, 1, 1])
ax.axis('equal')
langs = ['Low', 'High', 'Medium']
students = [12.5, 9.8, 77.7]
ax.pie(students, labels=langs, autopct='%1.2f%%')

plt.legend(['C', 'C++', 'Java', 'Python', 'PHP'], loc="upper right")

canvas = FigureCanvasTkAgg(fig, master=TAB_CONTROL)
canvas.get_tk_widget().place(x=25, y=250, bordermode="outside")
canvas.draw()
'''
mainWindow.mainloop()
```

6. Webhook

```
<?php
        /*Get Data From POST Http Request*/
        $datas = file_get_contents('php://input');
        /*Decode Json From LINE Data Body*/
        $deCode = json_decode($datas,true);

        file_put_contents('log.txt', file_get_contents('php://input') . PHP_EOL,
FILE_APPEND);

        $replyToken = $deCode['events'][0]['replyToken'];
        $userId = $deCode['events'][0]['source']['userId'];
        $text = $deCode['events'][0]['message']['text'];

        $messages = [];
        $messages['replyToken'] = $replyToken;
        $messages['messages'][0] = getFormatTextMessage("เอ้ย ถามอะไรก็ตอบได้");

        $encodeJson = json_encode($messages);

        $token =
'W9N/kT4ycwSPFrzZwTqJY5REZMja06pKVTLHZcAh75ZsJG9geLB7UX3jUrQLccmuLteOl
k+zraWvULt3o6NHYFF3fYm0W177BUj2c2bXOFfe4LYrxbVUieMKV5kO+YR3WCvXkiogaK
Ug564L7oR8HwdB04t89/1O/w1cDnyilFU=';
        #$LINEDatas['url'] = "https://api.line.me/v2/bot/message/reply";
        #$LINEDatas['token'] = $token;

        $LINEProfileDatas['url'] = "https://api.line.me/v2/bot/profile/".$userId;
        $LINEProfileDatas['token'] = $token;

        //$results = sentMessage($encodeJson,$LINEDatas);

        $resultsLineProfile = getLINEProfile($LINEProfileDatas);
```

```
$LINEUserProfile = json_decode($resultsLineProfile['message'],true);
$displayName = $LINEUserProfile['displayName'];
$userid = $LINEUserProfile['userId'];

$myfile = fopen("line_API.txt", "a") or die("Unable to open file!");
fwrite($myfile, $displayName.'@'.$userid."\r\n");
fclose($myfile);

/*Return HTTP Request 200*/
http_response_code(200);

function getFormatTextMessage($text)
{
        $datas = [];
        $datas['type'] = 'text';
        $datas['text'] = $text;

        return $datas;
}

 function getLINEProfile($datas)
{
        $datasReturn = [];

        $curl = curl_init();

        curl_setopt_array($curl, array(
          CURLOPT_URL => $datas['url'],
          CURLOPT_RETURNTRANSFER => true,
          CURLOPT_ENCODING => "",
          CURLOPT_MAXREDIRS => 10,
          CURLOPT_TIMEOUT => 30,
          CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
          CURLOPT_CUSTOMREQUEST => "GET",
```

```
            CURLOPT_HTTPHEADER => array(
              "Authorization: Bearer ".$datas['token'],
              "Postman-Token: 32d99c7d-9f6e-4413-a4d2-fa0a9f1ecf6d",
              "cache-control: no-cache"
            ),
          ));

          $response = curl_exec($curl);
          $err = curl_error($curl);

          curl_close($curl);

          if ($err) {
    $datasReturn['result'] = 'E';
    $datasReturn['message'] = $err;
} else {
    if($response == "{}"){
        $datasReturn['result'] = 'S';
        $datasReturn['message'] = 'Success';
    }else{
        $datasReturn['result'] = 'E';
        $datasReturn['message'] = $response;
    }
}

return $datasReturn;
  }

  function sentMessage($encodeJson,$datas)
  {
          $datasReturn = [];
          $curl = curl_init();
          curl_setopt_array($curl, array(
            CURLOPT_URL => $datas['url'],
```

```
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
  CURLOPT_POSTFIELDS => $encodeJson,
  CURLOPT_HTTPHEADER => array(
    "authorization: Bearer ".$datas['token'],
    "cache-control: no-cache",
    "content-type: application/json; charset=UTF-8",
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    $datasReturn['result'] = 'E';
    $datasReturn['message'] = $err;
} else {
    if($response == "{}"){
        $datasReturn['result'] = 'S';
        $datasReturn['message'] = 'Success';
    }else{
        $datasReturn['result'] = 'E';
        $datasReturn['message'] = $response;
    }
}

return $datasReturn;
}
```