ตัวสร้างรายงานจุดอ่อนช่องโหว่ด้านการรักษาความปลอดภัยอัตโนมัติ

AUTOMATIC VULNERABILITY REPORT GENERATOR

วิรัฒน์ วนไพศาล

ธนชัย รื่นจันทบุ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ตัวสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยอัตโนมัติ

Automatic Vulnerability Report Generator

วิวัฒน์ วนไพศาล
สุชัย รื่นสำราญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตัวสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยอัตโนมัติ

Automatic Vulnerability Report Generator

ผู้จัดทำ

1. นายวิวัฒน์	วนไพศาล	รหัสนักศึกษา	53011492
2. นายสุชัย	รื่นสำราญ	รหัสนักศึกษา	53011729

..... อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

ตัวสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยอัตโนมัติ

นาย วิวัฒน์	วนไพศาล	53011492
นาย สุชัย	รื่นสำราญ	53011729
อาจารย์ อัครเดช	วัชระภูพงษ์	อาจารย์ที่ปรึกษา

ปีการศึกษา 2556

บทคัดย่อ

งานด้านการรักษาความปลอดภัยระบบสารสนเทศต้องยุ่งเกี่ยวกับการสร้างและแก้ไขเอกสารจำนวนมาก ซึ่งมักวนซ้ำเดิม หากใช้คนจัดการทั้งหมดอาจเกิดความล่าช้าและข้อผิดพลาดง่าย แม้ว่ามีเครื่องมืออัตโนมัติมากมายสำหรับการค้นหาจุดอ่อนแอด้านการรักษาความปลอดภัย แต่ยังคงติดขัดในการแปลผลให้อยู่ในรูปแบบที่ต้องการ

โครงการนี้จึงมุ่งนำเสนอระบบต้นแบบเพื่อสร้างรายงานจุดอ่อนด้านการรักษาความปลอดภัยโดยอัตโนมัติ เพื่อเข้าใช้ร่วมกับเครื่องมือค้นหาจุดอ่อนแอดังกล่าว และสามารถปรับแต่งเอกสารรายงานให้อยู่ในรูปแบบที่เหมาะสมต่อการดำเนินงานต่อไป

Automatic Vulnerability Report Generator

Mr. Wiwat	Wanapaisan	53011492
Mr. Suchai	Ruensamran	53011729
Mr. Akkradach	Watcharapupong	Advisor

Academic Year 2013

ABSTRACT

Information security involves with creation and editing of abundance documents. Manual working makes all delays and error prone. Although there are many automated tools for finding vulnerabilities but still stuck in the interpretation of results in the desired format.

This project aims to create a prototype system for security flaws report documenting automatically. To use with other scanning tools then can customize report format suitable for following operations.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการดำเนินงาน การศึกษาและวิจัย โดยได้รับความช่วยเหลือจากบุคลากรดังต่อไปนี้

ขอขอบพระคุณอาจารย์ อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้ความรู้ คำแนะนำต่าง ๆ อีกทั้งความช่วยเหลือและแนวทางในการแก้ปัญหาที่พบระหว่างการทำโครงงานชิ้นนี้ จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (ISAG) สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ได้สนับสนุนเครื่องมือ รวมไปถึงข้อมูล และหนังสือต่าง ๆ ตลอดจนเพื่อน ๆ พี่ ๆ น้อง ๆ ที่อยู่ร่วมกันมา คอยให้กำลังใจและคำแนะนำเสมอมา

ขอขอบคุณนายพัฒนพล รัตนพงษ์พร, นายอาทิตยพงษ์ สุชินโรจน์ และนายธันยา ล้วนศรีติสกุล ที่ให้ความอนุเคราะห์ในการให้คำปรึกษาในการทำโครงงานชิ้นนี้ ในส่วนของการเก็บข้อมูลการพัฒนาต้นแบบการออกรายงานความปลอดภัย และแนวทางในการแก้ไขปัญหาต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ผู้ซึ่งเป็นทั้งกำลังใจตลอดจนการให้การสนับสนุนในทุก ๆ เรื่อง

นาย วิวัฒน์ วนไพศาล

นาย สุชัย รื่นสำราญ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญตาราง

ตารางที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันปัญหาในเรื่องของความปลอดภัยของข้อมูลในระบบสารสนเทศนั้นมีความสำคัญมาก เนื่องด้วยกิจกรรมส่วนใหญ่ในชีวิตประจำวันล้วนแล้วแต่มีระบบสารสนเทศเข้ามาเกี่ยวข้องด้วยทั้งสิ้น ซึ่งถ้าหากระบบสารสนเทศมีปัญหานั้น ก็จะทำให้เกิดผลกระทบมากมายขึ้น โดยอาจจะมีผู้ไม่หวังดีทำการบุกรุกและทำการโจรกรรมหรือทำการแก้ไขดัดแปลงข้อมูล ซึ่งทำให้เกิดการรั่วไหลของข้อมูลที่มีความสำคัญต่อองค์กรนั้น ๆ โดยผู้ไม่หวังดีสามารถกระทำผ่านทางช่องโหว่หรือจุดอ่อนแอต่าง ๆ ภายในระบบ ดังนั้นในปัจจุบันองค์กรต่าง ๆ จึงมีการให้ความสำคัญในเรื่องของการรักษาความปลอดภัยในระบบสารสนเทศมากขึ้นอย่างเห็นได้ชัด

ซึ่งวิธีการหนึ่งในการจัดการระบบรักษาความปลอดภัยคือการบริหารความเสี่ยง ซึ่งได้แก่การทำ Penetration Testing คือการทดสอบเพื่อหาช่องทางในการเข้าถึงระบบ และการทำ Vulnerability Assessment คือการประเมินหาความเสี่ยงที่เกิดจากช่องโหว่ที่พบ โดยกระบวนการทำงานจะประเมินความเสี่ยงจากค่าความเสี่ยง (Risk) ที่เกิดจากช่องโหว่ที่พบ (Vulnerability) ซึ่งจะทำให้เกิดภัยคุกคาม (Threat) โดยมีวัตถุประสงค์เพื่อการหาข้อมูลคุณลักษณะของระบบในมุมมองด้านการรักษาความปลอดภัยในระบบและแนวทางในการแก้ปัญหาต่าง ๆ

การหาช่องโหว่ของระบบในปัจจุบันมีการใช้เครื่องมือหรือ Vulnerability Scanner Tools เข้ามาใช้ในการตรวจสอบ เพื่อให้ผู้ดูแลระบบสารสนเทศเกิดความตระหนักและเห็นถึงปัญหาที่เกิดจากช่องโหว่ของระบบที่ยังไม่ได้รับการแก้ไข โดยข้อมูลเหล่านี้จะถูกนำเสนอการประเมินความเสี่ยงออกเป็นระดับเช่น High Risk, Medium Risk หรือ Low Risk และข้อมูลจากการวิเคราะห์จะอยู่ในรูปของรายงานหรือเอกสารต่าง ๆ

งานทางด้านการรักษาความปลอดภัยระบบสารสนเทศนั้น ต้องยุ่งเกี่ยวกับการสร้างและแก้ไขเอกสารจำนวนมาก ซึ่งเป็นงานที่ทำวนซ้ำแบบเดิม ๆ และเนื่องจากใช้คนในการทำเอกสารเหล่านี้ทั้งหมด จึงมีโอกาสที่จะเกิดข้อผิดพลาดสูง และมีปัญหาในเรื่องของเวลาในการทำงานเพราะเป็นงานที่มีปริมาณมากทำให้ต้องใช้เวลาในการจัดการค่อนข้างมาก ซึ่งแม้ว่าในปัจจุบันจะมีเครื่องมืออัตโนมัติที่ใช้ในการค้นหาจุดอ่อนแอของระบบมากมาย แต่ก็ยังคงมีข้อติดขัดในเรื่องของการแปลผลจากเครื่องมือค้นหาจุดอ่อนแอเหล่านั้นให้อยู่ในรูปแบบที่ต้องการสำหรับผู้ใช้

โครงงานนี้จึงนำเสนอระบบต้นแบบเพื่อสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยของระบบโดยอัตโนมัติ เพื่อใช้ร่วมกับเครื่องมือค้นหาจุดอ่อนแอดังกล่าวและสามารถปรับแต่งเอกสารให้อยู่ในรูปแบบที่สามารถนำไปใช้งานได้อย่างเหมาะสมและมีประสิทธิภาพ

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อคัดกรองเนื้อหาที่ไม่สำคัญออก เช่น รายงานช่องโหว่เวอร์ชันเก่า ๆ โดยคัดให้เหลือเฉพาะเวอร์ชันปัจจุบัน
2) เพื่อให้ผู้ใช้งานสามารถเลือกรายละเอียดของเนื้อหาที่ต้องการในการใช้งานได้
3) เพื่อจัดรูปแบบของรายงานให้อยู่ในรูปแบบที่สะดวกต่อการใช้งาน
4) เพื่อลดระยะเวลาในการทำงานของผู้ดูแลระบบหรือผู้ตรวจสอบระบบ
5) เพื่อลดข้อผิดพลาดอันเนื่องมาจากการทำงานของคน

1.3 ขอบเขตของโครงงาน

1) โปรแกรมสามารถประมวลผลร่วมกับไฟล์รายงานที่นามสกุล .XML
2) โปรแกรมสามารถกำหนดให้ผู้ใช้งานสามารถเลือกเงื่อนไขของข้อมูลในการออกเอกสารรายงาน
3) โปรแกรมสามารถส่งออกเอกสารรายงานที่มีเงื่อนไขตรงกับความต้องการของผู้ใช้งาน

1.4 วิธีการดำเนินการ

1) การใช้งานและวิธีการทำงานของเครื่องมือค้นหาจุดอ่อนแอ
2) ทดลองใช้งานเครื่องมือค้นหาจุดอ่อนแอในการตรวจสอบระบบ
3) ศึกษาถึงผลลัพธ์ที่ได้จากการใช้งานเครื่องมือค้นหาจุดอ่อนแอ
4) สัมภาษณ์ผู้ที่ทำงานที่เกี่ยวข้องถึงการทำงานและปัญหาที่พบในการทำงาน รวมไปถึงข้อเสนอแนะในการทำโครงงาน
5) ออกแบบโครงสร้างการทำงานของระบบ
6) ศึกษาทฤษฎีที่เกี่ยวข้องและสิ่งที่ต้องใช้กับระบบ
7) ศึกษาไลบรารีของภาษาไพทอนชนิดต่าง ๆ ที่เกี่ยวข้องกับระบบ ได้แก่ ไพมองโก (PyMongo) ในการจัดการฐานข้อมูลมองโกดีบี (MongoDB)
8) ศึกษาโครงสร้างและการทำงานของภาษา XML
9) ทดลองเขียนโปรแกรมดึงค่าจากเอกสารรายงานชนิด XML เพื่อนำเข้าข้อมูลสู่ฐานข้อมูล

10) ทดลองเขียนโปรแกรมเลือกค่าจากฐานข้อมูลเพื่อส่งออกสู่เอกสารรายงานจากความต้องการของผู้ใช้งาน

11) ปรับปรุงประสิทธิภาพของระบบเพื่อการทำงานที่มีประสิทธิภาพและรองรับการทำงานในอนาคต

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้ความเข้าใจเกี่ยวกับการทำงานทางด้านการรักษาความปลอดภัยข้อมูล
2) ช่วยให้ผู้ดูแลระบบหรือผู้ตรวจสอบระบบสามารถทำงานได้ง่าย และรวดเร็วยิ่งขึ้น
3) ได้รับความรู้ความเข้าใจเกี่ยวกับการทำงานของเครื่องมือการค้นหาจุดอ่อนแอของระบบ
4) ช่วยให้ผู้ดูแลระบบหรือผู้ตรวจสอบระบบไม่ต้องทำงานที่วนซ้ำเหมือนเดิม

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาโดยทั่วไปออกเป็น 7 บทด้วยกัน

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 XML (Extensive Markup Language) กล่าวถึงความเป็นมาและทฤษฎีที่ใช้ในโครงงาน ประกอบไปด้วย XML Parser และ XML DOM

บทที่ 3 การใช้งานฐานข้อมูลชนิด NoSQL กล่าวถึงความหมายและความสำคัญของการนำเทคโนโลยีฐานข้อมูลเชิงเอกสารที่มีชื่อว่ามองโกดีบี (MongoDB) มาใช้ในการออกแบบให้รองรับกับการทำงานของระบบ

บทที่ 4 ภาษาโปรแกรมมิ่งไพทอน กล่าวถึงความเป็นมาของภาษา และไลบรารี่ไพมองโก (PyMongo) ที่ใช้ในการออกแบบระบบภายในโครงงาน

บทที่ 5 การออกแบบและพัฒนา กล่าวถึงภาพรวมของระบบ การออกแบบฐานข้อมูล และการออกแบบส่วนติดต่อกับผู้ใช้

บทที่ 6 การทดลองและผลการทดลอง กล่าวถึงรูปแบบการทดลอง ซึ่งประกอบไปด้วยส่วนของการทดลองและส่วนของโปรแกรม

บทที่ 7 แนวทางการพัฒนาต่อ กล่าวถึงบทสรุป ปัญหาอุปสรรคและแนวทางแก้ไข แนวทางการพัฒนาต่อ

บทที่ 2

XML (Extensive Markup Language)

เอกซ์เอ็มแอล (XML) หรือ Extensible Markup Language เป็นภาษาที่ใช้สำหรับการเขียนเอกสารมาร์กอัพ (Markup Document) โดยที่เอกสารมาร์กอัพนั้นมีการใช้เมทาเดตา (Metadata) เพื่อบอกหน้าที่และประเภทของข้อมูลของส่วนต่าง ๆ ของเอกสารนั้นได้โดยชัดเจน การเพิ่มเมทาดาต้า เข้าไปในเอกสารสามารถทำให้โครงสร้างของเอกสารชัดเจนขึ้นและทำให้การประมวลผลเอกสารเป็นไปโดยง่ายและไม่จำเป็นที่จะต้องอาศัยมนุษย์เพื่อตีความเอกสาร

เราใช้เทคโนโลยี เอกซ์เอ็มแอล ในการพัฒนามาตรฐานเพื่อการกระจายข่าวเนื่องจาก เอกซ์เอ็มแอล เป็นภาษาที่เหมาะกับการแลกเปลี่ยนข้อมูลผ่านเครือข่ายคอมพิวเตอร์ เนื่องจาก เอกซ์เอ็มแอล ไม่ได้ขึ้นอยู่กับโปรแกรมประยุกต์หรือระบบปฏิบัติการใด นอกจากนี้ เอกซ์เอ็มแอลยังเป็นภาษาที่มีความยืดหยุ่น เนื่องจากผู้ใช้สามารถที่จะกำหนดและตั้งค่า เมทาเดตา (หรือ Tags) ให้เหมาะกับเอกสารเฉพาะที่ตนต้องการได้อย่างอิสระและยังสามารถเพิ่มเติม เมทาเดตา (หรือ Tags) ได้ในภายหลังโดยไม่มีผลกระทบต่อโปรแกรมที่มีอยู่แล้วด้วย และเอกซ์เอ็มแอลเหมาะสำหรับการแลกเปลี่ยนข้อมูลระหว่าง ฐานข้อมูล (Database) แบบ รีเลชันแนล (Relational) และแบบออบเจ็กต์ (Object) กับแอพพลิเคชันอื่น ๆ ที่ไม่ได้เกี่ยวข้องกับเอกสารโดยตรงอีกด้วย

โดยในโครงงานนี้ได้ศึกษาทฤษฎีที่เกี่ยวข้องกับ เอกซ์เอ็มแอล ในส่วนของ

1) XML Parser
2) XML DOM

2.1 XML Parser

พาร์เซอร์ (Parser) เป็นองค์ประกอบของซอฟต์แวร์ ที่ตั้งอยู่ระหว่างแอพพลิเคชันและ ไฟล์เอกซ์เอ็มแอลโดยมีวัตถุประสงค์เพื่อป้องกันผู้พัฒนาจากความซับซ้อนของความสัมพันธ์ของ เอกซ์เอ็มแอล

การที่จะรวม พาร์เซอร์ และ แอพพลิเคชัน นั้นมี 2 ทาง คือ

1) การใช้ Object-based interfaces
2) การใช้ Event-based interfaces

Object-based interface parser

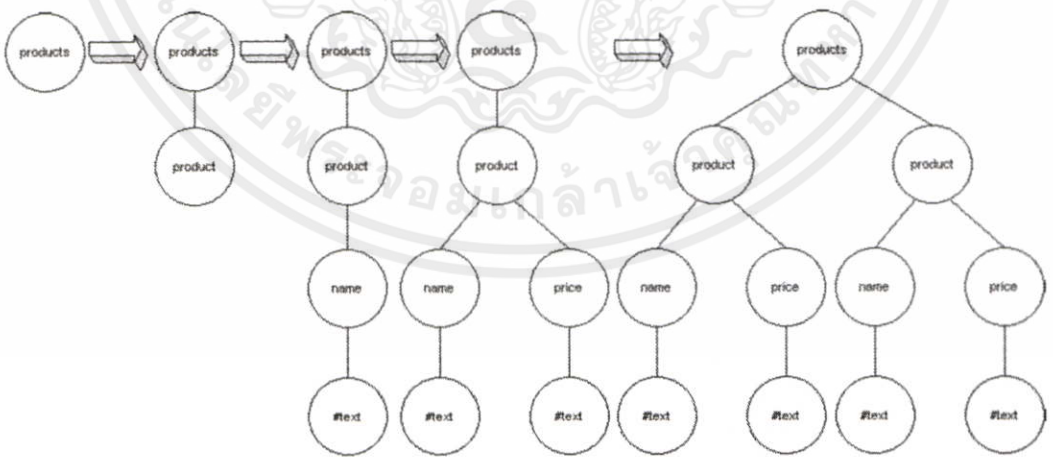
พาร์เซอร์ สร้างแผนผังรูปต้นไม้ของวัตถุอย่างชัดเจน โดยบรรจุ เอเลเมนต์ (Element) ทั้งหมดลงใน XML document

ออบเจ็กต์-เบส อินเตอร์เฟส (Object-based interface) อาจจะเป็นอินเตอร์เฟสที่เป็นธรรมชาติมากที่สุดสำหรับแอพพลิเคชัน เพราะเป็นการตกทอดของแผนผังรูปต้นไม้ในหน่วยความจำที่ตรงกันกับไฟล์บน ดิสก์ (Disk)

ออบเจ็กต์-เบส อินเตอร์เฟส เหมาะสำหรับแอพพลิเคชันที่ทำงานกับแผนผังรูปต้นไม้ในหน่วยความจำ ถ้าหากว่าตัวแอพพลิเคชันนั้นไม่สนใจวากยสัมพันธ์ของเอกซ์เอ็มแอล นอกจากนี้ถ้าใช้ Validating parser แล้ว แผนผังรูปต้นไม้อาจจะมีการตรวจและทำให้ถูกต้อง โดยมี ดีทีดี (DTD) เป็นตัวชี้วัด

โปรแกรมที่ 1.1 ตัวอย่างโค้ดของ XML Parser

```
<?xml version = "1.0"?>

<products>

      <product>

            <name>XML Editor</name>

            <price>499.00</price>

      </product>

      <product>

            <name>XML Book</name>

            <price>199.00</price>

      </product>

      <product>

            <name>XML Training</name>

            <price>699.00</price>

      </product>

</products>
```

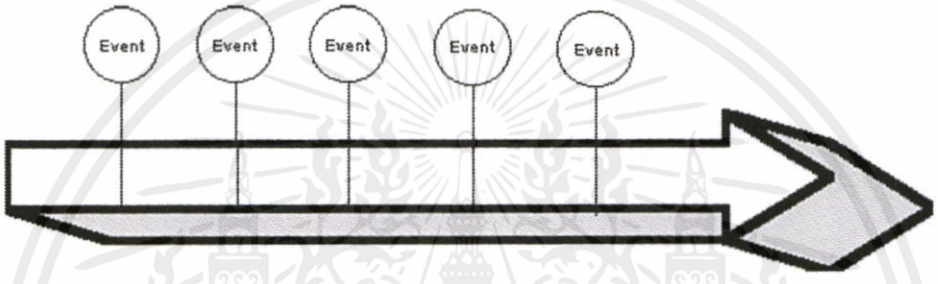


รูปที่ 2.1 แผนผังรูปต้นไม้ของวัตถุ

Event-based interface parser

อีเวนท์-เบส อินเตอร์เฟส (Event-based interface) เป็นธรรมชาติของพาร์เซอร์ แต่มีความซับซ้อนมากกว่า ออบเจ็กต์-เบส อินเตอร์เฟส สำหรับแอพพลิเคชัน แต่มีประสิทธิภาพมากกว่า

พาร์เซอร์ สร้างแผนผังรูปต้นไม้ของวัตถุไม่ชัดเจนเท่า ออบเจ็กต์-เบส อินเตอร์เฟส แต่จะอ่านไฟล์และสร้าง อีเวนท์ (Event) เมื่อเจอ เอเลเมนต์, แอททริบิวต์ (Attributes) หรือ เท็กซ์ (Text) ในไฟล์ แทนตัว อีเวนท์ นี้จะถูกสร้างสำหรับ เอเลเมนต์ สตาร์ท (Element starts) , เอเลเมนต์ เอนด์ (Element ends) , แอททริบิวต์ , เท็กซ์ คอนเทนท์ (Text content), เอนทิตี (Entities) ฯลฯ เป็นต้น



รูปที่ 2.2 การทำงานของ Event-based interface parser

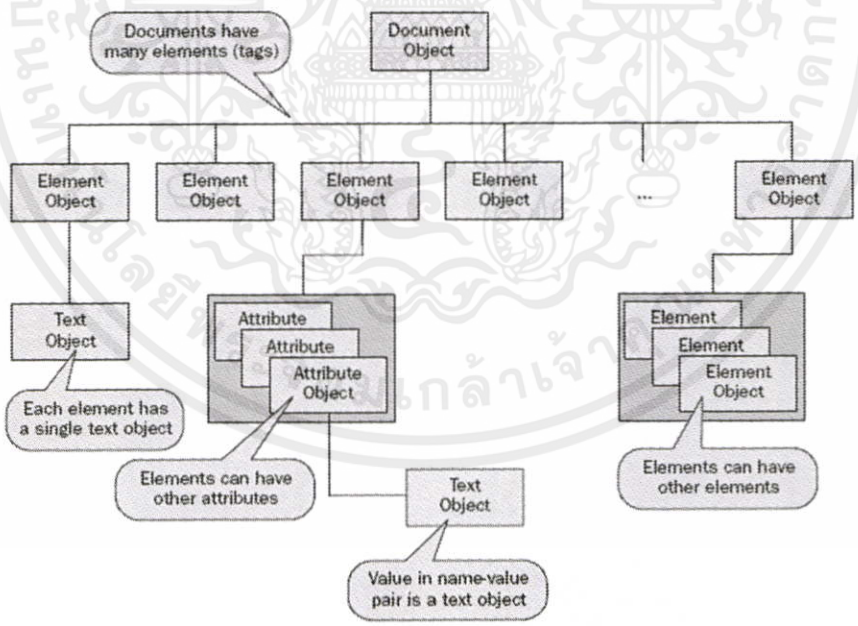
2.2 XML DOM

ในกระบวนการนำข้อมูล เอกซ์เอ็มแอล มาใช้งานใน แอพพลิเคชัน นั้นจะมี เอกซ์เอ็มแอล พาร์เซอร์ เป็นตัวกลางในการดึงข้อมูลจากเอกสารเอกซ์เอ็มแอล และแอพพลิเคชัน ซึ่งเป็น เอพีไอ (API) ชนิดหนึ่ง โดยเอพีไอ ที่นิยมกันมากคือ ดีโอเอ็ม (DOM) ซึ่งมีวิธีในการดึงข้อมูลคือ

DOM ย่อมาจาก Document Object Model มีการมองเอกสารเอกซ์เอ็มแอล ในลักษณะของโครงสร้างต้นไม้ (Tree) มีข้อจำกัดตรงปริมาณหน่วยความจำของเครื่อง แต่มีข้อดีคือในเขียนโค้ดสามารถเขียนได้ง่ายกว่ามาก

โปรแกรมที่ 1.2 ตัวอย่างโค้ดของ XML DOM

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This XML document describes a DVD library -->

      <library>
            <DVD id="1">
                  <title>Breakfast at Tiffany's</title>
                  <format>Movie</format>
                  <genre>Classic</genre>
            </DVD>
            <DVD id="2">
                  <title>Contact</title>
                  <format>Movie</format>
                  <genre>Science fiction</genre>
            </DVD>
            <DVD id="3">
                  <title>Little Britain</title>
                  <format>TV Series</format>
                  <genre>Comedy</genre>
            </DVD>
      </library>
```



รูปที่ 2.3 โครงสร้างของวัตถุ DOM

Class	Description	Child Classes	Example	Method nodeName()	Method nodeValue()	Method nodeType()
Node	Abstraction of other classes	NodeList (Node collection)				
Document	Root node	Element collection	<populationInThousands>			9 (DOCUMENT_NODE)
Element	Element or tag	Text	<country>	country		1 (ELEMENT_NODE)
Attr	An attribute name-value pair	Text	<country name="Germany">	name	Germany	2 (ATTRIBUTE_NODE)
Text	Textual value in element or attribute		<country> <pop>24 mill</pop> </country>	population	24 million	3 (TEXT_NODE)

รูปที่ 2.4 การเก็บค่าตามโครงสร้างของ XML DOM

DOM Node

วัตถุหลักใน ดีโอเอ็ม คือ โหนด (Node) ตัวโหนดเป็นวัตถุทั่วไปในแผนผังรูปต้นไม้ และวัตถุของ ดีโอเอ็ม ส่วนใหญ่จะเป็นการสืบเชื้อสายจาก โหนด ซึ่งเป็นฉบับพิเศษที่ใช้สำหรับ เอเลเมนต์, แอททริบิวต์, เอนทิตี, เท็กซ์ ฯลฯ.

โหนด สามารถจำกัดความของคุณสมบัติที่จะช่วยเดินสำรวจแผนผังรูป

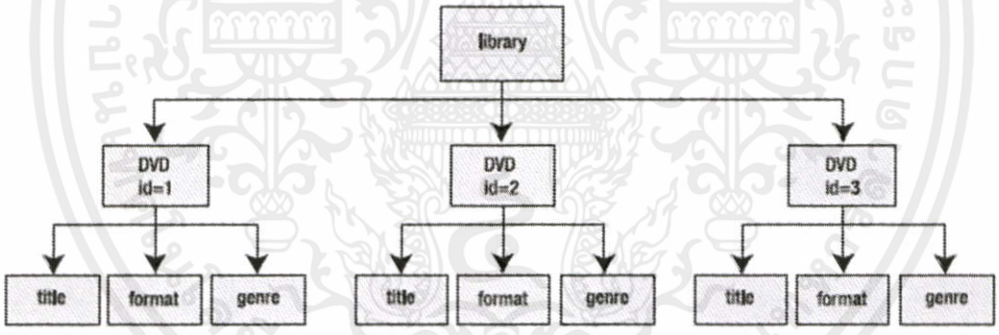
- nodeType คือโค้ดที่ทำหน้าที่แทนประเภทของ โหนด
- parentNode คือผู้ปกครองของวัตถุ โหนด ล่าสุด
- childNode คือรายการของ โหนด ที่เป็นลูกของ โหนด ล่าสุด
- firstChild คือ โหนด ตัวแรกที่เป็นลูก
- lastChild เป็น โหนด ตัวสุดท้ายที่เป็นลูก
- previousSibling คือ โหนด ที่ผ่านมาจาก โหนด ปัจจุบัน
- nextSibling คือ โหนด ที่ถัดมาจาก โหนด ปัจจุบัน
- attributes เป็นรายการของ แอททริบิวต์ ที่ โหนด ปัจจุบันมีอยู่

โหนด กำหนดคุณสมบัติ สองอย่างเพื่อควบคุมวัตถุ ดังข้างล่างนี้

- nodeName คือ แอททริบิวต์ ที่บ่งบอกถึงชื่อของ โหนด, แอททริบิวต์ นี้เทียบเท่ากับ แท็ก (tag) ชื่อของ เอเลเมนต์
- nodeValue เป็น แอททริบิวต์ ที่บ่งบอกถึงค่าที่อยู่ใน โหนด สมมติว่า โหนด นั้นเป็น โหนด จำพวกเท็กซ์ ค่าของ แอททริบิวต์ นี้จะเป็นเท็กซ์ ด้วย

โปรแกรมที่ 1.3 ตัวอย่างโค้ดของ XML DOM Node

```
<?xml version = "1.0"?>
- <customers>
  - <customer>
            <name phone='631-445-2231'>Zachary Smith</name>
       - <address>
                 <street>1 Smith Street</street>
                 <town>Smithtown</town>
                 <state>NY</state>
                 <zip>11723</zip>
            </address>
      </customer>
  +   <customer>
  +   <customer>
   - <customer>
            <name phone='212-123-5566'>James Bloggs</name>
       - <address>
                 <street>25 James Street</street>
                 <town>Manhatton</town>
                 <state>NY</state>
                 <zip>11124</zip>
            </address>
      </customer>
</customers>
```



รูปที่ 2.5 โครงสร้างแผนผังรูปต้นไม้

บทที่ 3
การใช้งานฐานข้อมูลชนิด NoSQL

ฐานข้อมูลชนิดโนเอสคิวแอล (NoSQL) จะมีรูปแบบกลไลสำหรับการจัดเก็บและการดึงข้อมูล โดยใช้วิธีการที่แตกต่างจากฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งแนวคิดในเรื่องของความเรียบง่ายในการออกแบบ การขยายขนาดและการควบคุมการใช้งานเป็นหัวใจหลักในการใช้งาน

ปัจจุบันกระแสของการใช้งานเทคโนโลยีที่โนเอสคิวแอลมีการใช้งานแพร่หลายมากขึ้น โดยสามารถแบ่งกลุ่มของเทคโนโลยีการเก็บข้อมูลออกเป็น 8 ประเภทคือ

- Document Store
- Graph
- Key-Value Store
- Hosted Services
- Multi value Database
- Object Database
- Tabular
- Tuple Store

แต่ละเทคโนโลยีจะมีความเหมาะสมในการใช้งานที่แตกต่างกันไป สำหรับมองโกดีบี (MongoDB) เป็นโนเอสคิวแอลในตระกูลด็อคคิวเม้นท์สโตร์ (Document Store) ที่ได้รับการยอมรับว่าเป็นหนึ่งในด็อคคิวเม้นท์สโตร์ที่ดีที่สุด

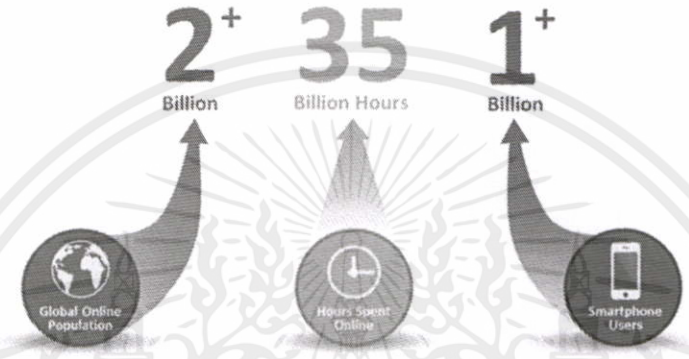
3.1 เหตุผลในการใช้งานโนเอสคิวแอล

3.1.1 ผู้ใช้งานมีจำนวนมากขึ้นเรื่อยๆ

ในปัจจุบันผู้ใช้งานอินเทอร์เน็ตมีแนวโน้มมากขึ้นเรื่อย ๆ ไม่ว่าจะเป็นการใช้งานผ่านเครื่องคอมพิวเตอร์หรืออุปกรณ์สื่อสาร ซึ่งเทคโนโลยีของอุปกรณ์สื่อสารนั้นมีหลากหลายมากขึ้น และสามารถใช้งานได้ง่ายยิ่งขึ้น นอกจากการพัฒนาระบบให้สามารถรองรับปริมาณการเข้าใช้งานของแต่ละอุปกรณ์ ระบบยังต้องรองรับวิธีการป้อนข้อมูลแบบใหม่คือ จากเดิมที่ผู้จัดการเนื้อหาต่างๆ เช่น ผู้ดูแลเว็บไซต์ ผู้ดูแลระบบ จะเป็นผู้ป้อนข้อมูลเข้าระบบ แต่ปัจจุบันผู้ใช้บริการ (User) โดยตรงผ่านอุปกรณ์ต่างๆที่มีหลากหลาย มีส่วนร่วมในการป้อนข้อมูล และการป้อนข้อมูลก็ง่ายกว่าแต่ก่อนมาก

อีกทั้งยังมีปัจจัยอื่น ๆ เช่น เมื่อถึงช่วงเทศกาลสำคัญ การที่จะมีผู้ใช้งานมากเป็นพิเศษ หรือผู้ใช้งานที่ไม่ใช่แค่บางประเทศเท่านั้น เพราะโลกอินเทอร์เน็ตถึงกัน ผู้ดูแลระบบจึงต้องดูว่าระบบมีผู้เข้าใช้งานจากต่างประเทศหรือทั่วโลกหรือไม่

ดังนั้นผู้ดูแลระบบจึงต้องจัดหาการพัฒนาฐานข้อมูล โดยไม่ใช่แค่ทำให้รองรับกับการเข้ามาใช้งานของผู้ใช้บริการเท่านั้น แต่ต้องรองรับการจัดเก็บข้อมูลที่มากขึ้นเรื่อย ๆ อีกด้วย



รูปที่ 3.1 จำนวนของผู้ใช้งานที่มีปริมาณเพิ่มมากขึ้น

3.1.2 ประเภทของข้อมูลมีความหลากหลาย และปริมาณข้อมูลที่ต้องการจัดเก็บมีมากขึ้น

จากตัวแปรของผู้ใช้งานที่มีปริมาณมากขึ้นเรื่อย ๆ อุปกรณ์ในการเข้าใช้งานมีหลากหลายมากยิ่งขึ้น อีกทั้งประเภทของข้อมูลที่ได้จากแต่ละอุปกรณ์มีหลากหลายประเภท เช่น ข้อความ รูปภาพ เสียง วิดีโอ ตำแหน่งสถานที่ (Geo Location) และอื่น ๆ และการป้อนข้อมูลเหล่านี้มีสามารถทำได้ง่ายเพราะเทคโนโลยีของฮาร์ดแวร์และซอฟต์แวร์มีการพัฒนามากขึ้น สามารถใช้งานได้ง่ายขึ้น สะดวกขึ้น และรวดเร็วขึ้น

ดังนั้นการจัดเก็บข้อมูลที่หลั่งไหลเข้ามาจากอุปกรณ์ต่าง ๆ อาจจะต้องทำการวิเคราะห์พฤติกรรมของผู้ใช้บริการ การส่งเสริมการตลาด การทำข้อมูลการตัดสินใจของผู้บริหาร ข้อมูลลูกค้าสัมพันธ์ และอื่น ๆ การนำระบบฐานข้อมูลเชิงสัมพันธ์ อาจจะไม่เหมาะกับลักษณะงานบางอย่างอีกต่อไป

3.1.3 การพัฒนาเทคโนโลยีด้านฮาร์ดแวร์ที่มีประสิทธิภาพดีขึ้น

ภาพรวมของการใช้งานเซิร์ฟเวอร์เริ่มเปลี่ยนไป คือ ใช้งานได้ง่ายขึ้น ราคาถูกลง แต่ประสิทธิภาพดีขึ้น ซึ่งเป็นสิ่งที่สำคัญในการนำมาพิจารณาของเทคโนโลยีด้านฐานข้อมูลคือ ถ้าต้องการจัดเก็บฐานข้อมูลขนาดใหญ่ หรือรองรับปริมาณผู้ใช้งานได้เป็นจำนวนมาก การขยายระบบฐานข้อมูลเป็นเรื่องที่สามารถทำได้ง่ายขึ้น โดยสามารถทำได้โดยการนำเอาเครื่องเซิร์ฟเวอร์มาต่อกันออกไป หรือเรียกว่าการขยายออกแนวราบ (Scale Out) ไม่ใช่การขยายระบบเหมือนแต่ก่อน คือ ขยายออกแนวตั้ง (Scale Up) และต้องใช้เครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง ซึ่งจะมีต้นทุนที่แพงกว่าการขยายแบบแนวราบมาก

ดังนั้นการขยายระบบที่อยู่บนพื้นฐานของโนเอสคิวแอล คือการรองรับการขยายระบบแบบแนวราบ ซึ่งจะกระจายข้อมูลไปเก็บที่เครื่องเซิร์ฟเวอร์หลาย ๆ เครื่อง และใช้เครื่องเซิร์ฟเวอร์ทั่วไปที่เรียกว่า (Commodity Server) โดยไม่จำเป็นต้องใช้เซิร์ฟเวอร์ที่เรียกว่า Enterprise Server ที่มีราคาแพงตามเสป็คที่สูงขึ้นเรื่อยๆ และการบริหารจัดการก็ยาก

3.2 ปัญหาของฐานข้อมูลเชิงสัมพันธ์

การใช้งานฐานข้อมูลเชิงสัมพันธ์ที่ต้องการรองรับการจัดเก็บข้อมูลขนาดใหญ่ จะมีปัญหาเรื่องการแบ่งข้อมูลเก็บเป็นส่วน (Sharding) และการกระจายหน่วยความจำ (Distributed Cache) เพราะเป็นตัวหลักที่ต้องทำเพื่อการขยายระบบฐานข้อมูลของฐานข้อมูลเชิงสัมพันธ์ให้สามารถรองรับปริมาณข้อมูลที่มากขึ้น และรองรับจำนวนการเข้าใช้งานระบบได้มากขึ้น

3.2.1 การแบ่งข้อมูลเก็บเป็นส่วนด้วยตนเอง (Manual Sharding)

การแบ่งตารางฐานข้อมูล (Table) ออกเป็นส่วน ๆ แล้วทำการกระจายไปจัดเก็บในหลาย ๆ เซิร์ฟเวอร์ เพื่อให้แต่ละตารางของฐานข้อมูลไม่จัดเก็บข้อมูลที่ปริมาณเยอะเกินไป เพราะถ้าข้อมูลในแต่ละฐานข้อมูลมีปริมาณเยอะเกินไป จะทำให้การทำงานของระบบฐานข้อมูลช้าไปด้วย และปัญหาที่จะตามมาคือเมื่อต้องกระจายข้อมูลออกไปในแต่ละเซิร์ฟเวอร์ การจะจัดการกับข้อมูล เช่น เพิ่ม แก้ไข ลบ ดึงข้อมูลมาแสดง จะต้องทำผ่านแอพลิเคชันหรือต้องมีเซิร์ฟเวอร์บางตัวที่ต้องทำการดึงข้อมูลแต่ละเซิร์ฟเวอร์มารวมเป็นก้อนเดียวกัน หมายความว่าผู้ดูแลระบบต้องทำด้วยตนเอง ไม่ใช่ระบบฐานข้อมูลจัดการให้

3.2.2 การกระจายหน่วยความจำ (Distributed Cache)

เมื่อต้องการให้ระบบรองรับการเข้าใช้งานในปริมาณที่มาก การจะเข้าถึงข้อมูลต้องทำการอ่านข้อมูลผ่านฐานข้อมูลโดยตรง ทำให้การรองรับการใช้งานข้อมูลเกิดความล่าช้าขึ้น ดังนั้นต้องมีการทำแคชเลเยอร์ (Cache Layer) ขึ้นมา คือแทนที่จะเข้าไปอ่านจากฐานข้อมูลโดยตรง ให้อ่านผ่านหน่วยความจำแคช (Cache) ก่อน ดังนั้นการอ่านข้อมูลจากหน่วยความจำจะเป็นการอ่านจากหน่วยความจำ (Memory) โดยตรง ทำให้รองรับปริมาณการเข้ามาใช้งานได้มากขึ้น

แต่ปัญหาคือการทำแคชเลเยอร์นี้รองรับเฉพาะการอ่านข้อมูลเท่านั้น ไม่รองรับการเขียนข้อมูลได้ ถ้าต้องการรองรับการเขียนข้อมูลปริมาณมาก และอ่านข้อมูลปริมาณมาก จึงเป็นสิ่งที่ฐานข้อมูลเชิงสัมพันธ์ไม่สามารถรองรับงานในลักษณะ อ่าน-เขียน (Write-Read) ข้อมูลปริมาณที่มากได้ดีนัก และประการสำคัญคือการทำแคชเลเยอร์จะต้องมีการดูแลรักษา และใช้เซิร์ฟเวอร์แยกออกไปต่างหาก ด้วยเหตุผลนี้เอง ทั้งการทำการแบ่งข้อมูลเก็บเป็นส่วนและหน่วยความจำแคช เป็นสิ่งที่ถูกพัฒนาขึ้นในเทคโนโลยีโนเอสคิวแอล โดยรองรับการแบ่งข้อมูลเก็บเป็นส่วนโดยอัตโนมัติ (Auto-Sharding) และการฝังหน่วยความจำแคชในตัวเอง (Integrated Caching)

3.3 คุณสมบัติของฐานข้อมูลโนเอสคิวแอล

3.3.1 การปรับเปลี่ยนรูปแบบของโครงสร้างตาราง (Dynamic Schemas)

การจัดเก็บข้อมูลต่าง ๆ ในฐานข้อมูลเชิงสัมพันธ์จะต้องมีการสร้างรูปแบบของโครงสร้างตาราง (Schema) ว่าจะจัดเก็บข้อมูลอะไรบ้าง เมื่อต้องการจัดเก็บข้อมูลเพิ่มเติมต้องเปลี่ยน รูปแบบของโครงสร้างตารางภายหลัง (Alter-Table) ก่อนจะจัดเก็บข้อมูลรูปแบบใหม่ได้

แต่ในปัจจุบันการจัดเก็บข้อมูลมีการเปลี่ยนแปลงตลอดเวลา เพราะความต้องการในการจัดเก็บข้อมูลต่าง ๆ มีหลากหลายมากยิ่งขึ้น การกำหนดโครงสร้างของตารางฐานข้อมูลเป็นอย่างไรนั้นเป็นเรื่องไม่ง่ายนัก หรือการต้องเปลี่ยนโครงสร้างฐานข้อมูลบ่อย ๆ โดยที่มีข้อมูลอยู่แล้วเป็นเรื่องที่ยากหรือทำไม่ได้เลย วิธีการคืออาจต้องแยกออกเป็นตารางใหม่ ซึ่งเป็นวิธีแก้ปัญหาชั่วคราวเท่านั้น

ระบบฐานข้อมูลแบบโนเอสคิวแอลไม่จำเป็นต้องมีรูปแบบของโครงสร้างตารางที่ตายตัว หรือไม่ต้องมีรูปแบบของโครงสร้างตารางก่อนที่จะจัดเก็บข้อมูล ข้อมูลแต่ละแถว สามารถจัดเก็บได้ตามต้องการ การเพิ่มหรือลดสามารถทำได้โดยไม่มีปัญหากับระบบ ทำให้สามารถจัดเก็บข้อมูลได้ตามที่ต้องการหรือมีเปลี่ยนแปลงตลอดเวลาได้สะดวกรวดเร็ว

3.3.2 การแบ่งข้อมูลเก็บเป็นส่วนโดยอัตโนมัติ (Auto-Sharding)

เมื่อข้อมูลมีขนาดใหญ่ หรือต้องการเพิ่มประสิทธิภาพในการอ่าน-เขียนข้อมูลปริมาณมาก การแบ่งข้อมูลเก็บเป็นส่วนในระบบโนเอสคิวแอลจะทำการกระจายข้อมูลไปยังเซิร์ฟเวอร์ต่างๆโดยอัตโนมัติ ผู้ดูแลระบบไม่จำเป็นต้องต้องเขียนโปรแกรมในการกระจายข้อมูลเองเหมือนฐานข้อมูลเชิงสัมพันธ์

3.3.3 การสำเนาข้อมูล (Replication)

การสำเนาข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง (Replication) เมื่อเซิร์ฟเวอร์หนึ่งเสียหาย อีกเครื่องหนึ่งจะขึ้นมาทำงานแทนทันที โดยข้อมูลของแต่ละเครื่องจะมีข้อมูลเหมือนกัน ดังนั้น การสำเนาข้อมูลเป็นคุณสมบัติที่ตอบสนองต่อการใช้งานที่ต้องการความต่อเนื่องได้ตลอดเวลา (High Availability)

3.3.4 การฝังหน่วยความจำแคชในตัวเอง (Integrated Caching)

การจัดเก็บข้อมูลที่ใช้งานบ่อยเข้าไว้ในหน่วยความจำสำรอง เป็นคุณสมบัติเด่นของโนเอสคิวแอล ที่รวมหน่วยความจำแคชไว้ในตัวเอง ไม่จำเป็นต้องทำแคชเลเยอร์ฐานข้อมูลเชิงสัมพันธ์อีกต่อไป ทำให้ไม่จำเป็นต้องดูแลรักษาระบบที่แยกออกไปต่างหากอีกด้วย

3.4 ฐานข้อมูลมองโกดีบี (MongoDB)

รูปที่ 3.2 สัญลักษณ์ของมองโกดีบี

ฐานข้อมูลมองโกดีบี (MongoDB) คือ Document-oriented Database หรือฐานข้อมูลเชิงเอกสาร ซึ่งเป็นการเก็บข้อมูลในลักษณะที่แตกต่างจากเชิงสัมพันธ์ แนวคิดหลักของฐานข้อมูลเชิงเอกสาร คือไม่มีการจัดเก็บข้อมูลลงไปในลักษณะที่เป็นตารางประกอบด้วยแถวและคอลัมน์ (Row/Column) อีกต่อไป เพราะการเก็บเป็นแถวจะมีข้อจำกัดหลายอย่าง เนื่องจากตารางต่าง ๆ จะรูปแบบของโครงสร้างตารางกำกับไว้ชัดเจน ดังนั้นข้อมูลแต่ละแถวจะต้องมีจำนวนคอลัมน์เท่ากัน ถึงแม้ว่าข้อมูลในคอลัมน์ใด ๆ จะไม่มีก็ตาม ผลที่ตามมาคือจะได้ตารางที่มีลักษณะที่เป็นช่องโหว่เป็นจำนวนมาก ดังนั้นการเก็บเอกสารแบบเชิงเอกสารจะช่วยแก้ปัญหานี้ได้เพราะไม่มีรูปแบบของโครงสร้างตาราง เอกสารแต่ละใบจะมีจำนวนแอททริบิวต์ (Attribute) กี่ตัวก็ได้โดยจะเท่ากันหรือไม่เท่ากันก็ได้ ทำให้สามารถเก็บข้อมูลที่ซับซ้อนได้ดี

3.4.1 คุณสมบัติของมองโกดีบี

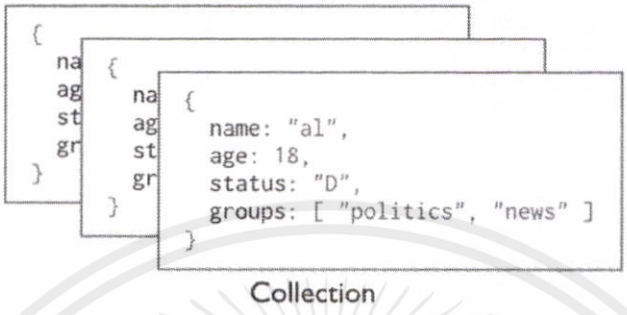
มองโกดีบีได้สนับสนุนคุณลักษณะหลายชนิดด้วยกัน โดยจะประกอบไปด้วย

1) มีการเข้ารหัสในทุกสตริง UTF - 8 และ Non-UTF-8 อย่างสม่ำเสมอ อีกทั้งยังสามารบันทึก, สอบถาม และเรียกข้อมูลไบนารีชนิดพิเศษได้
2) สนับสนุนการทำงานข้ามแพลตฟอร์ม โดยสามารถทำงานร่วมกับระบบปฏิบัติการวินโดวส์ (Windows), ลินุกส์ (Linux), แม็คโอเอส (MacOS), และโซลาริส (Solaris)

3.4.2 แนวคิดพื้นฐานของมองโกดีบี

1) มองโกดีบีมีแนวคิดพื้นฐานเรื่องรูปแบบของโครงสร้างตารางเช่นเดียวกับฐานข้อมูลที่คุ้นเคยกัน ดังนั้นสามารถที่จะสร้างและมีฐานข้อมูลกี่ตัวก็ได้ ซึ่งฐานข้อมูลแต่ตัวจะทำหน้าที่เป็นเหมือนที่เก็บของประเภทต่าง ๆ ที่มีในมองโกดีบี

2) สำหรับฐานข้อมูลหนึ่งตัวสามารถที่จะมีคอลเลคชัน (Collection) กี่ตัวก็ได้ ซึ่งสามารถเปรียบเทียบคอลเลคชันเหมือนกับตาราง (Table) ได้เช่นกัน



รูปที่ 3.3 ลักษณะคอลเลคชันของมองโกดีบี

3) สำหรับคอลเลคชันเองนั้นมีไว้สำหรับเก็บสิ่งที่เรียกว่าด็อคคิวเม้นท์ (Document) ซึ่งสามารถเปรียบเทียบกับแถว (Row)

4) สำหรับด็อคคิวเม้นท์ประกอบไปด้วยฟิลด์ (Field) ต่าง ซึ่งสามารถเปรียบเทียบกับคอลัมน์ (Column)

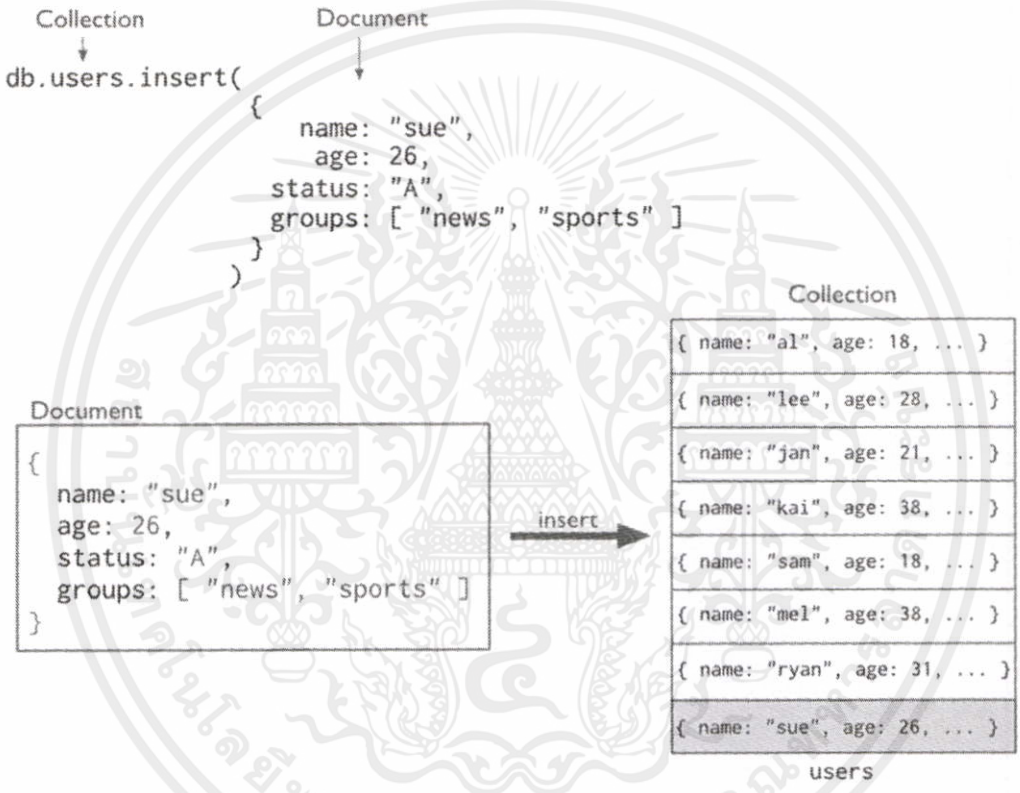
รูปที่ 3.4 ลักษณะด็อคคิวเม้นท์ของมองโกดีบี

5) อินเด็กซ์ (Index) ในมองโกดีบีมีลักษณะเช่นเดียวกับฐานข้อมูลเชิงสัมพันธ์

6) เคอร์ซอร์ (Cursor) มีความสำคัญมากในมองโกดีบี เพราะใช้อ้างอิงแทนข้อมูลที่ต้องการ โดยทุกครั้งที่เรียกหาข้อมูลจากมองโกดีบี สิ่งที่ได้กลับมาคือเคอร์เซอร์ โดยที่ไม่ได้ดึงข้อมูลจริง ๆ ออกมา

3.4.3 การจัดการข้อมูล

การจัดการข้อมูลของมองโกดีบีจะประกอบไปด้วยการดำเนินการต่าง ๆ ได้แก่ การสร้าง (Create), การแก้ไข (Update) และการลบข้อมูล (Delete) โดยในการดำเนินการเหล่านี้สามารถจัดการข้อมูลได้เพียงทีละคอลเล็คชันเดียวเท่านั้น สำหรับการดำเนินการแก้ไขและลบข้อมูลสามารถระบุเงื่อนไขในการดำเนินการกับคอลเล็คชันแบบเจาะจงได้



รูปที่ 3.5 ตัวอย่างการเพิ่มด็อคคิวเม้นท์ไปยังคอลเล็คชัน

ตารางที่ 3.1 การเปรียบเทียบการบัญญัติศัพท์ระหว่างเอสคิวแอล (SQL) และมองโกดีบี (MongoDB)

รูปแบบการบัญญัติศัพท์ของเอสคิวแอล (SQL)	รูปแบบการบัญญัติศัพท์ของมองโกดีบี (MongoDB)
database	database
table	collection
row	document หรือ BSON document
column	field
index	index
table joins	embedded documents และ linking
primary key **หมายเหตุ** เป็นการระบุเฉพาะแถวหรือคอลัมน์เพื่อเป็นไพรมารีย์คีย์ (primary key)	primary key **หมายเหตุ** ไพรมารีย์คีย์จะถูกตั้งให้เป็น _id field โดยอัตโนมัติ
WHERE	$match
GROUP BY	$group
HAVING	$match
SELECT	$project
ORDER BY	$sort
LIMIT	$limit
SUM()	$sum
COUNT()	$sum

3.4.4 ตัวอย่างการใช้งานรูปแบบโครงสร้างข้อมูลระหว่างเอสคิวแอล (SQL) และมองโกดีบี (MongoDB)

ตารางที่ 3.2 การสร้างตารางข้อมูลระหว่างเอสคิวแอลและมองโกดีบี

SQL	MongoDB
CREATE TABLE users (id MEDIUMINT NOT NULL AUTO_INCREMENT, user_id Varchar(30), age Number, status char(1), PRIMARY KEY (id))	db.users.insert({ user_id: "abc123", age: 55, status: "A" })

ตารางที่ 3.3 การเพิ่มข้อมูลระหว่างเอสคิวแอลและมองโกดีบี

SQL	MongoDB
INSERT INTO users(user_id, age, status) VALUES ("bcd001", 45, "A")	db.users.insert({ user_id: "bcd001", age: 45, status: "A" })

ตารางที่ 3.4 การแสดงข้อมูลระหว่างเอสคิวแอลและมองโกดีบี

SQL	MongoDB
SELECT * FROM users WHERE status = "A"	db.users.find({ status: "A" })

ตารางที่ 3.5 การอัพเดทข้อมูลระหว่างเอสคิวแอลและมองโกดีบี

SQL	MongoDB
UPDATE users SET status = "C" WHERE age > 25	db.users.update({ age: { $gt: 25 } }, { $set: { status: "C" } }, { multi: true })

ตารางที่ 3.6 การลบข้อมูลระหว่างเอสคิวแอลและมองโกดีบี

SQL	MongoDB
DELETE FROM users WHERE status = "D"	db.users.remove({ status: "D" })

บทที่ 4
ภาษาโปรแกรมมิ่งไพทอน

ไพทอนเป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูงถูกสร้างขึ้นในปี 1989 โดย Guido van Rossum ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษาไพทอน ได้ทั้งบนระบบ ยูนิกซ์ (Unix), ลินุกซ์ (Linux), วินโดวส์เอนที (Windows NT), วินโดวส์2000 (Windows 2000), วินโดวส์เอกซ์พี (Windows XP) หรือแม้แต่ระบบ ฟรีบีเอสดี (FreeBSD) อีกอย่างหนึ่งภาษานี้เป็นภาษาลักษณะ โอเพนซอร์ซ (Open Source) เหมือนอย่าง พีเอชพี (PHP)

4.1 หลักการทำงานของภาษา ไพทอน (Python)

เมื่อเราได้เขียนโค้ดขึ้นมาตามโครงสร้างของโปรแกรมภาษาใดก็ตาม และการจะให้โค้ดคำสั่งเหล่านั้นทำงานได้ก็จะต้องมีตัวแปรภาษามาจัดการแปลโค้ดคำสั่ง เพื่อให้ทำงานตามที่เราต้องการ โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภทใหญ่ ๆ คือ

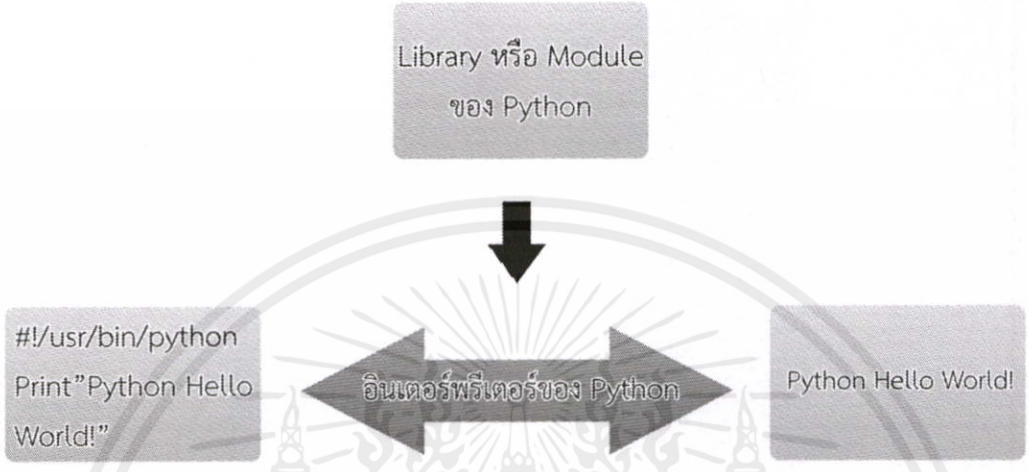
4.1.1 คอมไพเลอร์ (Compiler)

เป็นตัวแปลภาษาสำหรับภาษาซี (C), ซีพลัสพลัส (C++), ปาสคาล (Pascal) การทำงานก็คือจะตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการคอมไพล์ ถ้าไม่มีข้อผิดพลาดก็จะทำการแปลโค้ดคำสั่งให้เป็นไฟล์นามสกุล .obj (object file) จากนั้นก็ทำการแปลงไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ภาษาซี ดังรูปที่ 4.1

รูปที่ 4.1 การแปลภาษาสำหรับภาษา C, C++, Pascal

4.1.2 อินเตอร์พรีเตอร์ (Interperter)

จะทำงานเป็นบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมาเลย ดังรูปที่ 4.2



รูปที่ 4.2 การเรียกใช้ฟังก์ชั่นจากไลบรารี (Library) หรือโมดูล (Module) ของภาษาไพทอน

จากรูปตัวอย่างในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารี (Library) หรือโมดูล (Module) ของภาษาไพทอน อินเตอร์พรีเตอร์ของภาษาไพทอน ก็จะไปทำการเรียกฟังก์ชันเหล่านั้นให้ทำงานแล้วจึงแสดงผลการทำงานออกมา ในส่วนของประสิทธิภาพการทำงานนั้นตัวแปลภาษาแบบคอมไพเลอร์จะทำงานได้เร็วกว่าตัวแปลภาษาแลลอินเตอร์พรีเตอร์ เพราะโค้ดคำสั่งถูกคอมไพล์และลิงค์โดยตัวแปลภาษาแบบคอมไพเลอร์ผ่านแล้วได้เป็นไฟล์ .exe ออกมา จากนั้นก็เป็นขั้นตอนการทำงานอย่างเดียว

4.2 ความสามารถของภาษาไพทอน

ในปัจจุบันภาษาที่ใช้ในการพัฒนา เว็บ แอพพลิเคชัน (Web Application) มีมากมายหลายภาษา อาทิเช่น ภาษาเพิร์ล (Perl), พีเอชพี (PHP), จาวา (JAVA), เอเอสพี (ASP), ทีซีแอล (Tcl), ไพทอน (Python) เป็นต้น สำหรับภาษาไพทอน นับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษาไพทอน ทำให้มีผู้นิยมใช้มากขึ้นเรื่อย ๆ ซึ่งพอสรุปข้อดีของภาษาไพทอนได้ดังนี้

1) ง่ายต่อการเรียนรู้ โดยภาษาไพทอน มีโครงสร้างของภาษาไม่ซับซ้อน เข้าใจง่าย ซึ่งโครงสร้างภาษาไพทอน จะคล้ายกับภาษาซีมาก เพราะภาษาไพทอน สร้างขึ้นมาโดยใช้ภาษาซี ทำให้ผู้ที่คุ้นเคยภาษาซี อยู่แล้วสามารถใช้งานภาษาไพทอนได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความและเท็กซ์ไฟล์ (Text File) ได้เป็นอย่างดี
2) ไม่ต้องเสียค่าใช้จ่ายใด ๆ ทั้งสิ้น เพราะตัวแปรภาษาไพทอน อยู่ภายใต้ลิขสิทธิ์จีเอนยู (GNU)
3) ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษาไพทอน ถูกออกแบบใช้งานกับระบบยูนิกซ์อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษาไพทอน ให้สามารถใช้กับระบบปฏิบัติการอื่น ๆ อาทิเช่น ลินุกซ์ (Linux), วินโดวส์95/98/เอมอี (Windows 95/98/ME), วินโดวส์เอนที (Windows NT), วินโดวส์2000 (Windows 2000), โอเอสทู (OS/2)
4) ภาษาไพทอน ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่าง ๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษาซี, ซีพลัสพลัส, จาวา, เพิร์ล
5) ภาษาไพทอน เป็นภาษาประเภท เซิร์ฟเวอร์-ไซด์ สคริปต์ (Server side Script) คือการทำงานของภาษาไพทอน จะทำงานด้านฝั่ง เซิร์ฟเวอร์ (Server) แล้วส่งผลลัพธ์กลับมายังไคลเอนต์ (Client) ทำให้มีความปลอดภัยสูง
6) โค้ดที่เขียนด้วยไพทอน สามารถนำไปรันบนระบบปฏิบัติการได้หลากหลาย
7) สนับสนุนเทคโนโลยี ซีโอเอม (COM) ของไมโครซอฟต์วินโดวส์ (Microsoft Windows)
8) ไพทอนรวมมาตรฐานการอินเตอร์เฟส Tkinter ซึ่งสนับสนุนบนระบบ เอกซ์-วินโดวส์ (X windows), ไมโครซอฟต์วินโดวส์ (Ms-windows) และ แมคอินทอช (Macintosh) การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่น ๆ
9) เป็น ไดนามิคไทป์ปิง (Dynamic typing) คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
10) มี บิวท์-อิน ออบเจ็กต์ ไทป์ (Built-in Object Types) คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ในไพทอน ประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
11) มีเครื่องมือต่าง ๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนคำ เป็นต้น
12) มีมอดูลสำหรับจัดการ เรกูลาร์-เอกซ์เพรสชัน (Regular Expression)
13) มีมอดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ ซีโอเอม (COM), อิมเมจ (Image), ซีโออาร์บีเอ (CORBA), โออาร์บี (ORBs) และ เอกซ์เอ็มแอล (XML) เป็นต้น

14) จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ

15) อนุญาตให้ฝังชุดคำสั่งของไพทอน เอาไว้ภายในโค้ดภาษาซี หรือ ซีพลัสพลัส ได้

16) อนุญาตให้โปรแกรมเมอร์สร้าง ไดนามิค ลิงค์ ไลบรารี (Dynamic Link Library) หรือ ดีแอลแอล (DLL) เพื่อใช้ร่วมกับไพทอน

17) มีมอดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ค โปรเซส เธรด เรกูลาร์-เอกซ์เพรสชัน, เอกซ์เอ็มแอล, กราฟฟิค ยูสเซอร์ อินเตอร์เฟส (GUI) และอื่น ๆ

18) ประกอบด้วยมอดูลสำหรับสร้าง อินเทอร์เนตสคริปต์ (Internet Script) และติดต่อกับอินเทอร์เน็ตผ่าน ซ็อกเก็ต (Sockets), และทำหน้าที่เป็น ซีจีไอสคริปต์ (CGI Script) ตลอดจนใช้งานคำสั่ง เอฟทีพี (FTP), โกเฟอร์ (Gopher), เอกซ์เอ็มแอล และอื่น ๆ อีกมาก

19) สามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ

20) มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น มายเอสคิวแอล (MySQL), ไซเบส (Sybase), ออราเคิล (Oracle), อินฟอร์มิกซ์ (Informix), โอดีบีซี (ODBC) และอื่น ๆ

21) มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในรูปแบบต่าง ๆ ได้อย่างสะดวกและมีประสิทธิภาพ

22) มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์

23) มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer

24) มีไลบรารีสำหรับสร้าง ช็อคเวฟส์แฟลช (Shockwaves Flash) โดยไม่ต้องติดตั้ง มาโครมีเดียแฟลช (Macromedia Flash)

25) ใช้พัฒนา เว็บเซอร์วิส (Web Service) โดยที่ภาษาไพทอน สามารถนำมาพัฒนาเว็บเซอร์วิส รวมทั้งใช้บริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า ซีเอมเอฟ (Content Management Framework) ตัวอย่าง ซีเอมเอฟ ที่มีชื่อเสียงมากและเบื้องหลังทำงานด้วยไพทอน คือ Plone

4.3 โมดูลสำหรับการจัดการฐานข้อมูล

ภาษาไพทอนมีโมดูลที่มีความสามารถในการจัดการกับฐานข้อมูลเชิงเอกสาร สำหรับโมดูลที่มีหน้าที่ดูแลไฟล์ฐานข้อมูลนี้มีชื่อว่าไพมองโก (PyMongo) โดยมีความสามารถในการทำงานร่วมกับฟอร์แมตบีสัน (BSON) ของภาษาไพทอน

โปรแกรม 4.1 ฟังก์ชันการเชื่อมต่อฐานข้อมูลเข้ากับการทำงานของระบบ

```
from pymongo import MongoClient
client = MongoClient()
client = MongoClient('localhost', 27017)
```

โปรแกรม 4.2 ฟังก์ชันการเชื่อมต่อกับฐานข้อมูล

```
db = client.test_database
```

โปรแกรม 4.3 ฟังก์ชันการเชื่อมต่อกับคอลเล็คชัน

```
collection = db.test_collection
```

โปรแกรม 4.4 ฟังก์ชันการเชื่อมต่อและสร้างด็อคคิวเม้นท์

```
import datetime
post = {"author": "Mike",
"text": "My first blog post!",
"tags": ["mongodb", "python", "pymongo"],
"date": datetime.datetime.utcnow()}
```

บทที่ 5
การออกแบบโครงงาน

5.1 ภาพรวมของระบบ

โครงงานนี้ได้ใช้กระบวนการการประมวลผลร่วมกับเอกสารรายงาน โดยการนำเข้าเอกสารรายงานที่ได้จากเครื่องมือค้นหาจุดอ่อนแอของระบบ หลังจากนั้นโปรแกรมจะทำการเก็บค่ารายละเอียดต่าง ๆ ไว้บนฐานข้อมูลเพื่อรองรับการเลือกค่าต่าง ๆ ในการนำไปใช้งาน โดยผู้ใช้งานสามารถเลือกเงื่อนไขของรูปแบบของเอกสารรายงาน จากนั้นระบบจะทำการออกเอกสารรายงานซึ่งตรงกับความต้องการของผู้ใช้งาน โดยการทำงานของระบบจะแบ่งออกเป็น 4 ขั้นตอนดังนี้

1) **การนำเข้าเอกสารรายงานจากผู้ใช้งาน**

เป็นการนำเข้าเอกสารรายงานจากเครื่องมือค้นหาจุดอ่อนของระบบที่มีชนิดของรายงานเป็น .Nessus เท่านั้น โดยเมื่อผู้ใช้งานทำการนำเข้าเอกสารรายงาน ระบบจะทำการนำข้อมูลต่าง ๆ จากเอกสารรายงานไปเก็บไว้บนฐานข้อมูล

2) **การรับค่าเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงาน**

เป็นการรับค่าเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงานจากผู้ใช้งาน โดยแบ่งออกเป็น

- Result summary
- Vulnerability by host
- Compare with previous report

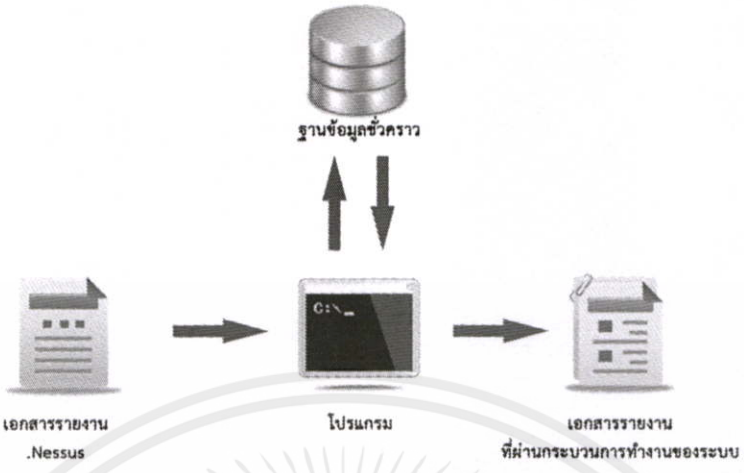
3) **การประมวลผลข้อมูลจากฐานข้อมูล**

ระบบจะนำข้อมูลเงื่อนไขความต้องการจากผู้ใช้งานมาประมวลผลร่วมกับข้อมูลบนฐานข้อมูลเพื่อนำข้อมูลต่าง ๆ ที่ตรงกับเงื่อนไขเพื่อออกเอกสารรายงาน

4) **การส่งออกเอกสารรายงาน**

เป็นการส่งออกเอกสารรายงานที่ตรงกับความต้องการของผู้ใช้งาน โดยมีรูปแบบดังนี้

- .DOC เป็นรูปแบบเอกสารของโปรแกรม Microsoft Word
- .PDF เป็นรูปแบบเอกสารชนิด Portable Document Format



รูปที่ 5.1 ภาพรวมของการทำงานของระบบ

5.2 การออกแบบฐานข้อมูล (Database Design)

ระบบได้มีการใช้งานฐานข้อมูลเชิงเอกสาร โดยจะทำการเก็บค่าต่าง ๆ จากเอกสารรายงานที่ได้จากเครื่องมือค้นหาจุดอ่อนแอเพื่อช่วยในการสร้างเอกสารรายงานตามความต้องการของผู้ใช้งาน

```
{
    ip : 161.246.4.3
    port : 53
    svc_name : "dns"
    protocol : "udp"
    severity : 2
    pluginID : 10539
    pluginName : "DNS Server Recursive Query Cache Poisoning Weakness"
    pluginFamily : "DNS"
    BID : 136 678
    CVE : "CVE-1999-0024"
    cvss base score : 5.0
    cvss temporal score : 4.3
    cvss temporal vector : CVSS2#E:U/RL:U/RC:C
    cvss vector : CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N
    description : "It is possible to query the remote name server for third party names."
    exploit canvas item :
    exploit framework core :
    exploit framework metasploit :
    patch publication date_item :
    risk factor item : "Medium"
    see also : "http://www.nessus.org/u?c4dcf24a"
    solution : "Restrict recursive queries to the hosts that should use this nameserver
            (such as those of the LAN connected to it)."
    synopsis : "The remote name server allows recursive queries to be performed by the host running nessusd."
    vulnerability publication date : 1997/08/01
    XREF : "OSVDB:438 CERT-CC:CA-1997-22"
    plugin_output_item" :
}
```

รูปที่ 5.2 ฐานข้อมูลเชิงเอกสารที่ใช้ภายในระบบ

5.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface)

ในการออกแบบส่วนติดต่อกับผู้ใช้ ได้เลือกการทำงานร่วมกับ Command Prompt เพื่อการทำงานโดยอัตโนมัติ การรับข้อมูลเข้าสู่ระบบจะเป็นคำสั่งที่ได้รับมาจากผู้ใช้งานโดยตรง โดยเมื่อผู้ใช้งานทำการเปิดโปรแกรมจะแสดงด้วยการพิมพ์คำสั่ง Project.py หรือ Project.py –h จะได้หน้าต่างโปรแกรมที่มีการบอกถึงรายละเอียดวิธีการใช้งาน และ ตัวเลือกต่างๆในการใช้งาน ได้แก่

-h หรือ --help เป็นการแสดงหน้าต่างตัวเลือกการทำงานพร้อมคำอธิบาย

-i หรือ --input ตามด้วยชื่อไฟล์นามสกุล .Nessus จะเป็นการระบุชื่อไฟล์ที่จะนำมาสร้างรายงาน

-s หรือ --summary เป็นการเลือกรูปแบบของรายงานแบบ Results summary

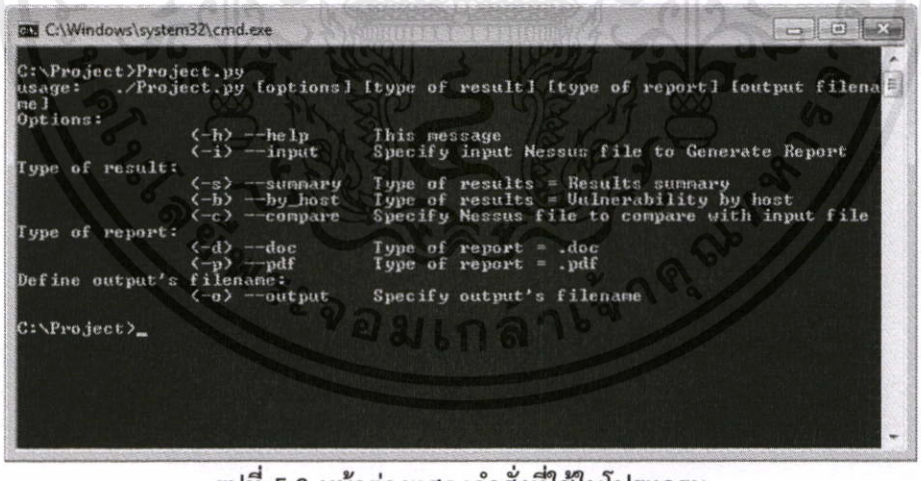
-b หรือ --by_host เป็นการเลือกรูปแบบของรายงานแบบ Vulnerability by host

-c หรือ --compare เป็นการเลือกรูปแบบของรายงานแบบ Compare โดยจะต้องระบุชื่อไฟล์นามสกุล .Nessus ที่จะนำมาเปรียบเทียบกับอินพุทไฟล์

-d หรือ --doc เป็นการเลือกให้รายงานที่สร้างนั้นออกเป็นไฟล์นามสกุล .DOC

-p หรือ --pdf เป็นการเลือกให้รายงานที่สร้างนั้นออกเป็นไฟล์นามสกุล .PDF

-o หรือ --output ตามด้วยชื่อไฟล์รายงานที่ต้องการจะสร้าง เป็นการระบุชื่อไฟล์รายงานตามที่ผู้ใช้ต้องการ



รูปที่ 5.3 หน้าต่างแสดงคำสั่งที่ใช้ในโปรแกรม

โดยสามารถแบ่งเงื่อนไขการจัดเรียงเนื้อหาของเอกสารรายงานได้ดังนี้

1) Result summary

เป็นเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงาน โดยแสดงผลลัพธ์ทั้งหมดของระบบที่ทำการตรวจสอบ

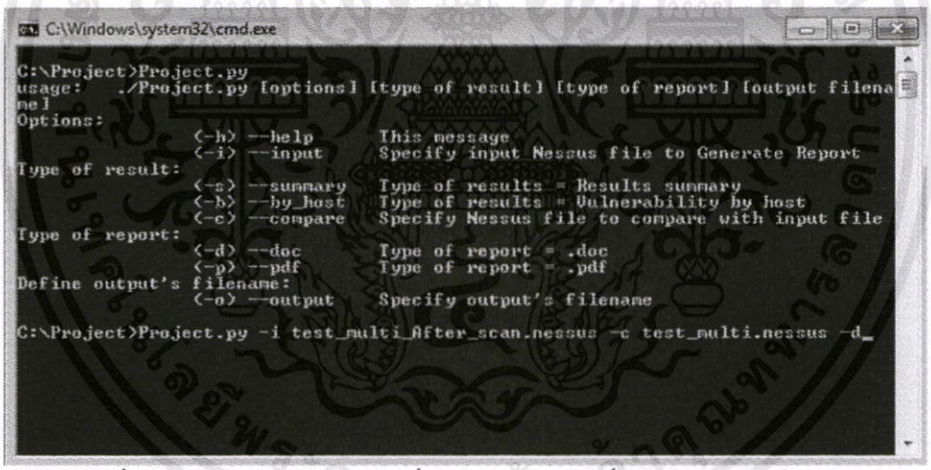
2) Vulnerability by host

เป็นเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงาน โดยแสดงรายละเอียดต่าง ๆ โดยเรียงตามเครื่องของระบบที่ทำการตรวจสอบ

3) Compare with previous report

เป็นเงื่อนไขการเปรียบเทียบเอกสารรายงาน โดยจะทำการเปรียบเทียบเอกสารรายงานและทำการแสดงรายละเอียดของการเปลี่ยนแปลงสู่เอกสารรายงาน

หากเลือกเงื่อนไขเปรียบเทียบ จะต้องระบุชื่อไฟล์นามสกุล .Nessus ที่จะนำมาเปรียบเทียบกับอินพุทไฟล์ โดยระบุหลังจากพารามิเตอร์ –c

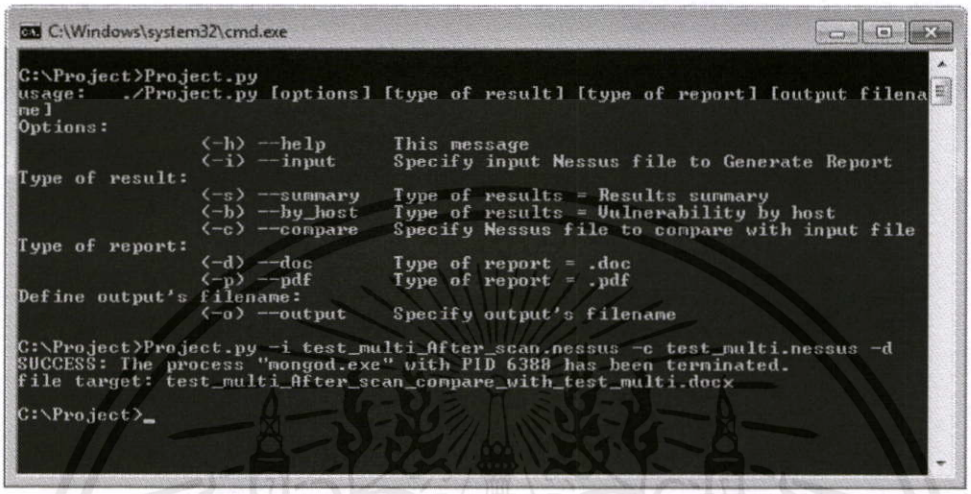


รูปที่ 5.4 หน้าต่างแสดงการใส่ชื่อไฟล์ หากเลือกเงื่อนไขการเปรียบเทียบ

การเลือกเงื่อนไขของรูปแบบของเอกสารรายงาน โดยแบ่งออกเป็น

- .DOC โดยเป็นรูปแบบเอกสารของโปรแกรม Microsoft Word
- .PDF โดยเป็นรูปแบบเอกสารชนิด Portable Document Format

เมื่อทำการสร้างเอกสารรายงานเสร็จสิ้น จะมีการแสดงชื่อของเอกสารที่ได้สร้างขึ้น ในส่วนของ *file target :* ดังรูปที่ 5.5



รูปที่ 5.5 หน้าต่างแสดงชื่อไฟล์เอกสารที่ทำสการสร้างขึ้นมา

5.4 ภาพรวมและรูปแบบในการส่งออกเอกสารรายรายงาน

ภายในเอกสารรายงานได้ถูกแบ่งเนื้อหาออกเป็นส่วน ๆ ตามลักษณะของข้อมูล โดยแบ่งออกเป็น

1) หน้าปกเอกสารรายงาน

ภายในจะมีการแสดงถึงเงื่อนไขของการออกเอกสารรายงาน และมีรายละเอียดของชื่อเอกสารรายงานเดิม อีกทั้งมีเวลาที่ทำการค้นหาจุดอ่อนแอ และเวลาที่ส่งออกเอกสารรายงานฉบับใหม่

2) ภาพรวมของระบบ

เป็นการแสดงรายละเอียดของเครื่องที่ดำทำการตรวจสอบจุดอ่อนแอของระบบ โดยประกอบด้วยเวลาที่ทำการตรวจสอบ รายละเอียดภาพในเครื่อง และจำนวนของจุดอ่อนแอภายในระบบ

3) ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานปัจจุบัน

เป็นการแสดงถึงการสรุปรายการจุดอ่อนแอภายในระบบโดยเรียงตามระกับความเสี่ยงของแต่ละเครื่องซึ่งเป็นรายละเอียดของเอกสารรายงานปัจจุบัน

4) ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานก่อนหน้า

เป็นการแสดงถึงการสรุปรายการจุดอ่อนแอภายในระบบโดยเรียงตามระกับความเสี่ยงของแต่ละเครื่องซึ่งเป็นรายละเอียดของเอกสารรายงานก่อนหน้า

5) ตารางแสดงการเปรียบเทียบจุดอ่อนแอของระบบ

เป็นการแสดงถึงจุดอ่อนแอที่ตรวจพบและทำการเปรียบเทียบจุดอ่อนแอนั้นระหว่างเอกสารรายงาน 2 ฉบับ

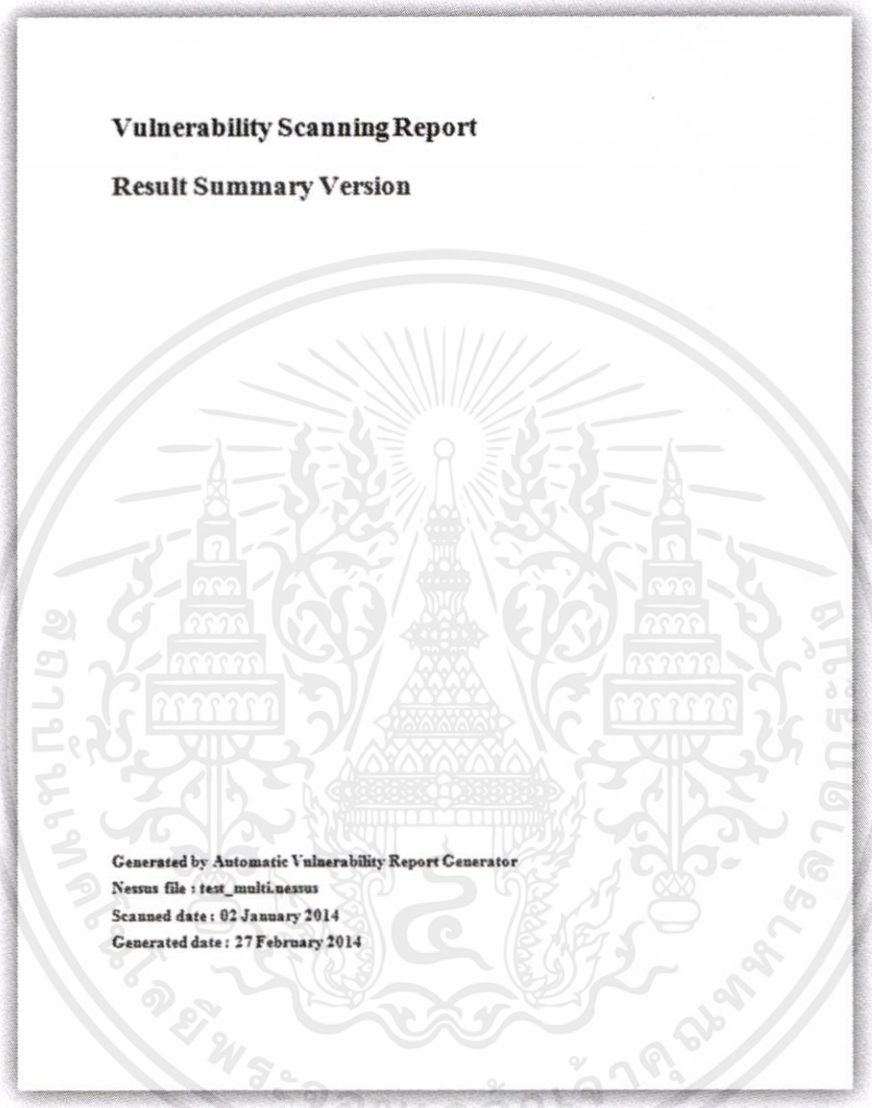
6) ภาพผนวก

เป็นการแสดงรายละเอียดเฉพาะของจุดอ่อนแอที่ตรวจพบในระบบ

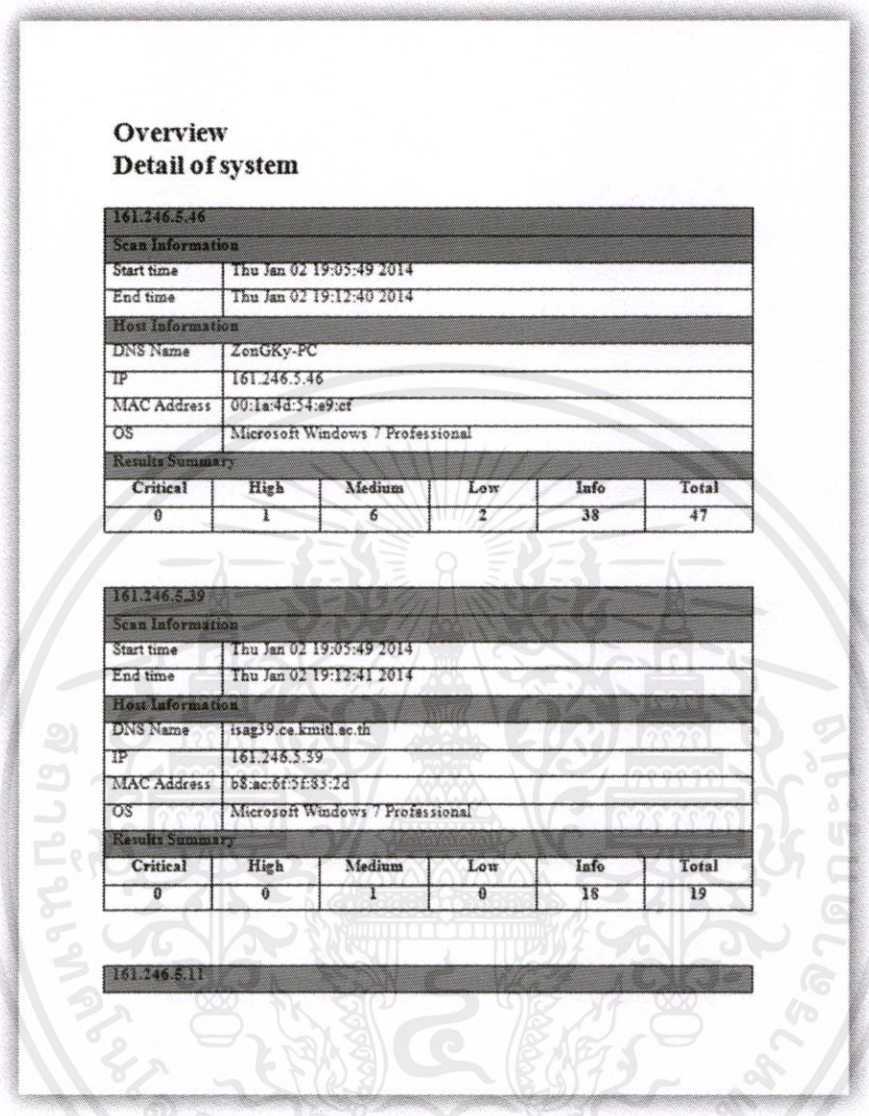
โดยผู้ใช้งานสามารถเลือกรูปแบบในการออกเอกสารรายงานตามความเหมาะสมในการใช้งานได้ 2 รูปแบบด้วยกันคือ

1) .DOC โดยเป็นรูปแบบเอกสารของโปรแกรม Microsoft Word
2) .PDF โดยเป็นรูปแบบเอกสารชนิด Portable Document Format

5.4.1 เอกสารรายงานในรูปแบบของไฟล์ .DOC



รูปที่ 5.6 หน้าปกเอกสารรายงาน



Overview
Detail of system

161.246.5.46	
Scan Information	
Start time	Thu Jan 02 19:05:49 2014
End time	Thu Jan 02 19:12:40 2014
Host Information	
DNS Name	ZonGKy-PC
IP	161.246.5.46
MAC Address	00:1a:4d:54:e9:cf
OS	Microsoft Windows 7 Professional

Results Summary

Critical	High	Medium	Low	Info	Total
0	1	6	2	38	47

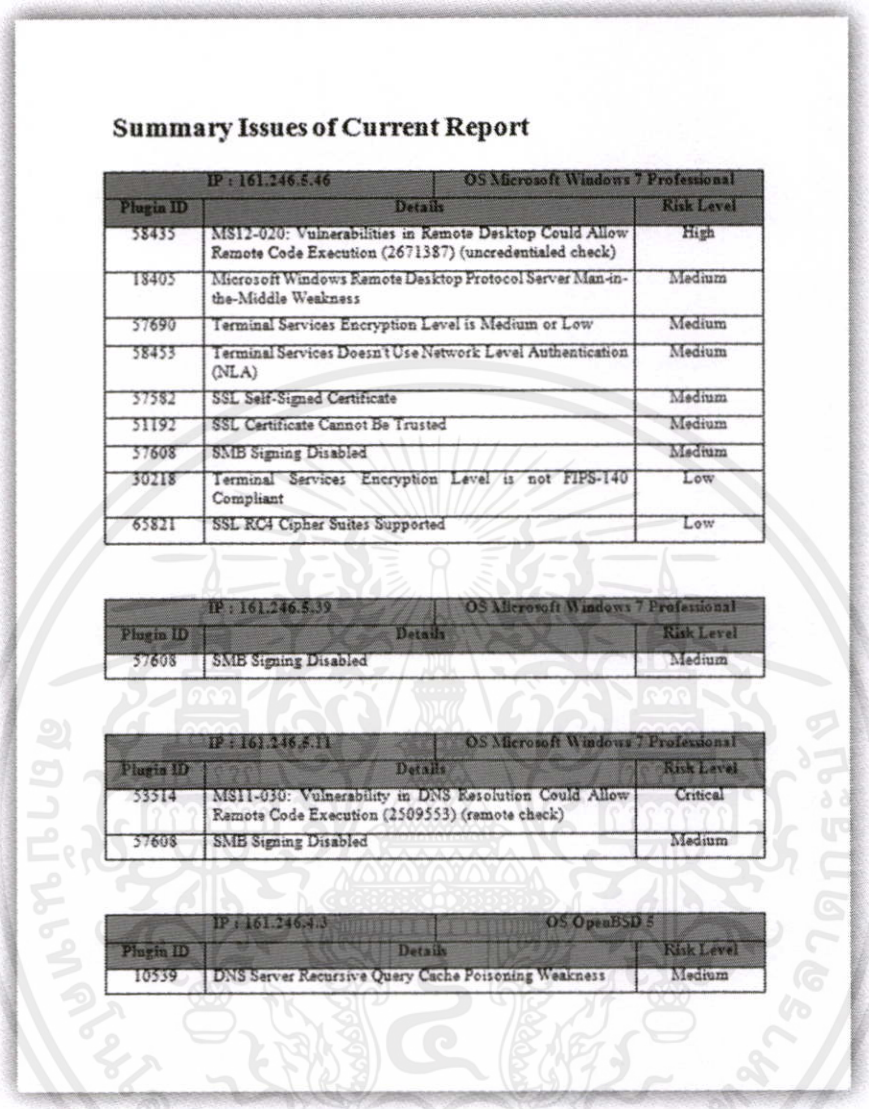
161.246.5.39	
Scan Information	
Start time	Thu Jan 02 19:05:49 2014
End time	Thu Jan 02 19:12:41 2014
Host Information	
DNS Name	isag39.ce.kmitl.ac.th
IP	161.246.5.39
MAC Address	b8:ac:6f:5f:83:2d
OS	Microsoft Windows 7 Professional

Results Summary

Critical	High	Medium	Low	Info	Total
0	0	1	0	18	19

161.246.5.11

รูปที่ 5.7 ภาพรวมของระบบ



Summary Issues of Current Report

IP : 161.246.5.46	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High
18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Medium
57690	Terminal Services Encryption Level is Medium or Low	Medium
58453	Terminal Services Doesn't Use Network Level Authentication (NLA)	Medium
57582	SSL Self-Signed Certificate	Medium
51192	SSL Certificate Cannot Be Trusted	Medium
57608	SMB Signing Disabled	Medium
30218	Terminal Services Encryption Level is not FIPS-140 Compliant	Low
65821	SSL RC4 Cipher Suites Supported	Low

IP : 161.246.5.39	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
57608	SMB Signing Disabled	Medium

IP : 161.246.5.11	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical
57608	SMB Signing Disabled	Medium

IP : 161.246.4.3	OS OpenBSD 5	
Plugin ID	Details	Risk Level
10539	DNS Server Recursive Query Cache Poisoning Weakness	Medium

รูปที่ 5.8 ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานปัจจุบัน

Summary Issues of Previous Report

IP : 161.246.5.46	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High
18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Medium
57690	Terminal Services Encryption Level is Medium or Low	Medium
58453	Terminal Services Doesn't Use Network Level Authentication (NLA)	Medium
57582	SSL Self-Signed Certificate	Medium
51192	SSL Certificate Cannot Be Trusted	Medium
57608	SMB Signing Disabled	Medium
30218	Terminal Services Encryption Level is not FIPS-140 Compliant	Low
65821	SSL RC4 Cipher Suites Supported	Low

IP : 161.246.5.39	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
57608	SMB Signing Disabled	Medium

IP : 161.246.5.11	OS Microsoft Windows 7 Professional	
Plugin ID	Details	Risk Level
53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical
57608	SMB Signing Disabled	Medium

IP : 161.246.4.3	OS OpenBSD 5	
Plugin ID	Details	Risk Level
10539	DNS Server Recursive Query Cache Poisoning Weakness	Medium

รูปที่ 5.9 ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานก่อนหน้า

Vulnerabilities Comparison

Current scan : test_multi_After_scan.nessus
Scanned date : 23 February 2014
Previous scan : test_multi.nessus
Scanned date : 02 January 2014

Number	Nessus Plugin ID	Finding Issue	Risk Level	Previous Scan	Current Scan
1	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical	161.246.5.11	161.246.5.46
2	58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High	161.246.5.46	161.246.5.46
3	57608	SMB Signing Disabled	Medium	161.246.5.46 161.246.5.39 161.246.5.11	161.246.5.6 161.246.5.46 161.246.5.11
4	45411	SSL Certificate with Wrong Hostname	Medium		161.246.5.6
5	71428	PHP 5.5.x < 5.5.7 OpenSSL openssl_x509_parse() Memory Corruption	Medium		161.246.5.6
6	11213	HTTP TRACE / TRACK Methods Allowed	Medium		161.246.5.6
7	51192	SSL Certificate Cannot Be Trusted	Medium	161.246.5.46	161.246.5.6 161.246.5.46
8	57582	SSL Self-Signed Certificate	Medium	161.246.5.46	161.246.5.6 161.246.5.46
9	18405	Microsoft Windows Remote Desktop Protocol Server Man-in-	Medium	161.246.5.46	161.246.5.46

รูปที่ 5.10 ตารางแสดงการเปรียบเทียบจุดอ่อนแอของระบบ

Appendix

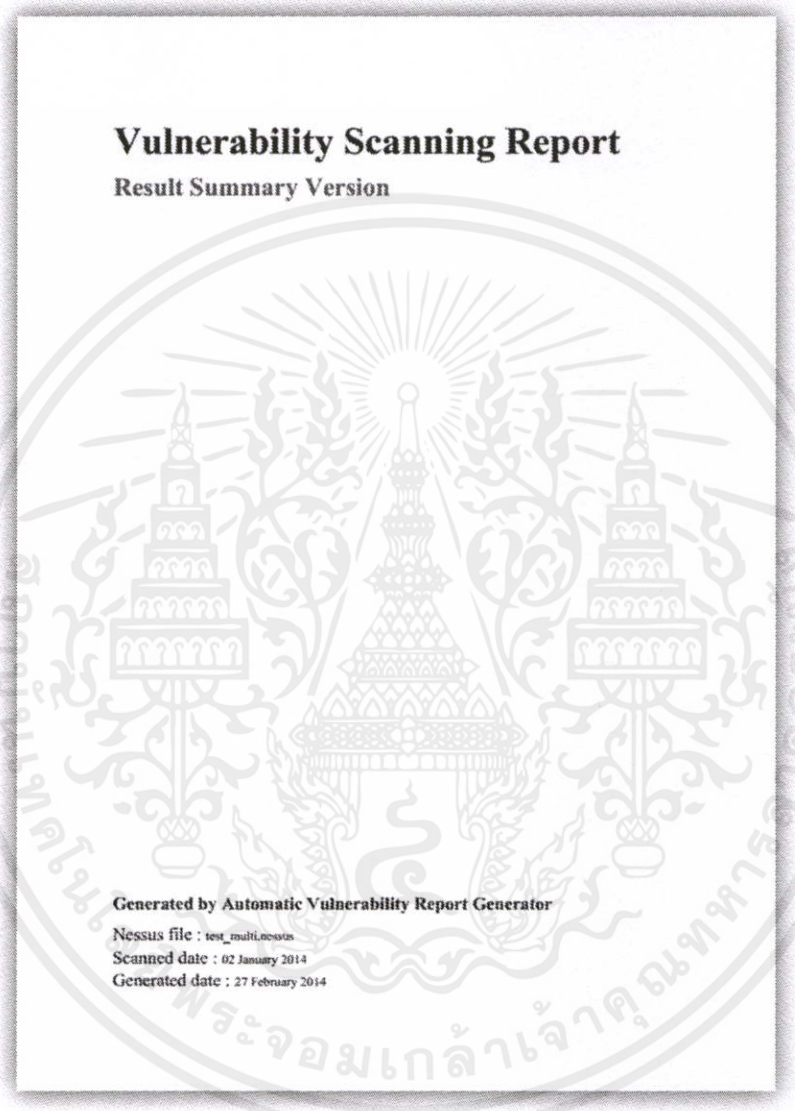
Details of Vulnerabilities Scanning Result

Vulnerability # 1	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
Plugin ID	53514
Risk Level	Critical
Synopsis	Arbitrary code can be executed on the remote host through the installed Windows DNS client.
Description	A flaw in the way the installed Windows DNS client processes Link- local Multicast Name Resolution (LLMNR) queries can be exploited to execute arbitrary code in the context of the NetworkService account. Note that Windows XP and 2003 do not support LLMNR and successful exploitation on those platforms requires local access and the ability to run a special application. On Windows Vista, 2008, 7, and 2008 R2, however, the issue can be exploited remotely.
Solution	Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2.
See Also	http://technet.microsoft.com/en-us/security/bulletin/ms11-030
List of Hosts	161.246.5.11 (udp/5355)
Risk Factor (Reference)	CVE : CVE-2011-0657 BID : BID:47242 Other References : OSVDB:71780 IAVA:2011-A-0039 MSFT:MS11-030
CVSS Score	CVSS Base Score : 10.0 CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C CVSS Temporal Score : 7.8 CVSS2#E:POC/RL:OF/RC:C
Exploit With	Core Impact (true) Metasploit (true)
Vulnerability Publication Date	2011/04/12

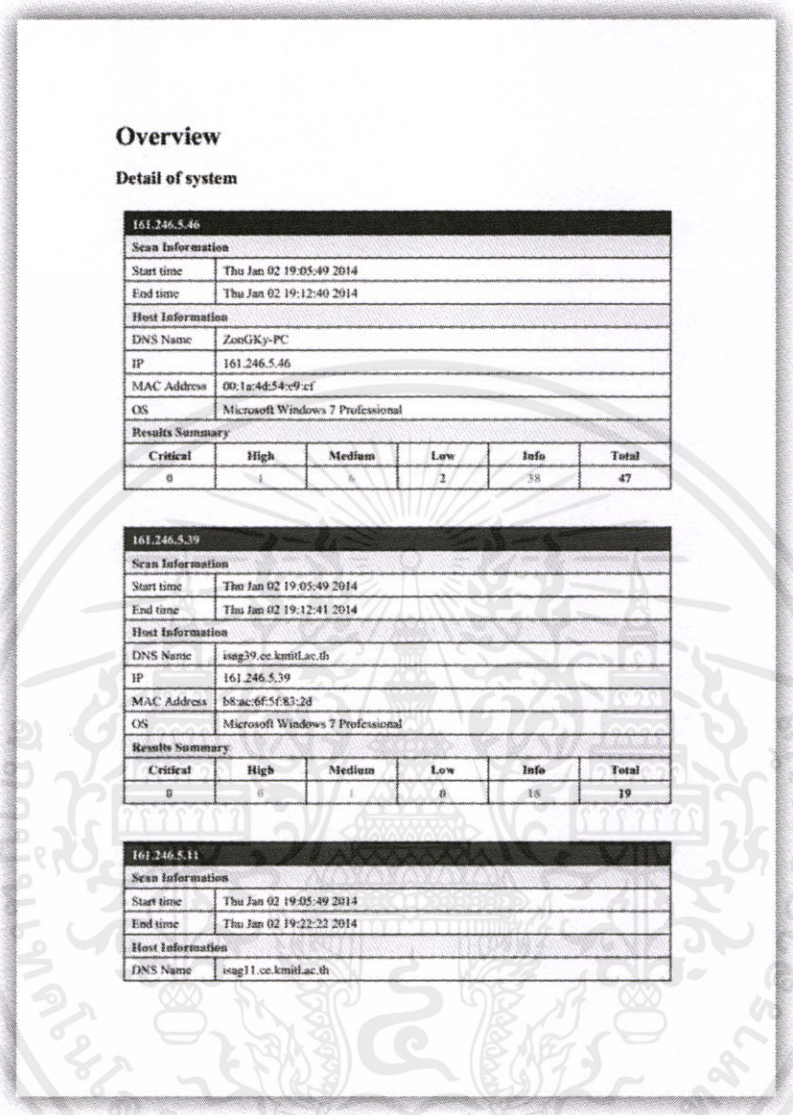
Vulnerability # 2	MS11-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)
Plugin ID	58435
Risk Level	High

รูปที่ 5.11 ภาคผนวก

5.4.2 เอกสารรายงานในรูปแบบของไฟล์ .PDF



รูปที่ 5.12 หน้าปกเอกสารรายงาน



Overview

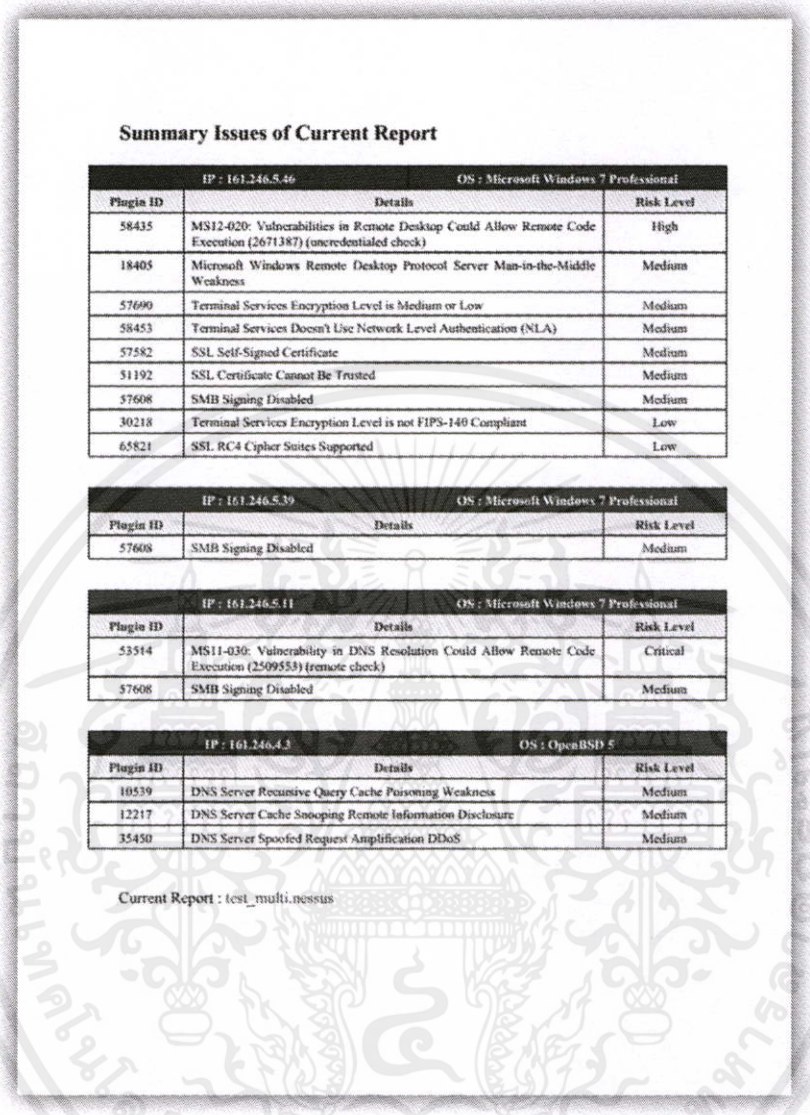
Detail of system

161.246.5.46					
Scan Information					
Start time	Thu Jan 02 19:05:49 2014				
End time	Thu Jan 02 19:12:40 2014				
Host Information					
DNS Name	ZonGKy-PC				
IP	161.246.5.46				
MAC Address	00:1a:4d:54:c9:cf				
OS	Microsoft Windows 7 Professional				
Results Summary					
Critical	**High**	**Medium**	**Low**	**Info**	**Total**
0	1	6	2	38	**47**

161.246.5.39					
Scan Information					
Start time	Thu Jan 02 19:05:49 2014				
End time	Thu Jan 02 19:12:41 2014				
Host Information					
DNS Name	isag39.ce.kmitl.ac.th				
IP	161.246.5.39				
MAC Address	b8:ac:6f:5f:83:2d				
OS	Microsoft Windows 7 Professional				
Results Summary					
Critical	**High**	**Medium**	**Low**	**Info**	**Total**
0	0	1	0	18	**19**

161.246.5.11	
Scan Information	
Start time	Thu Jan 02 19:05:49 2014
End time	Thu Jan 02 19:22:22 2014
Host Information	
DNS Name	isag11.ce.kmitl.ac.th

รูปที่ 5.13 ภาพรวมของระบบ



Summary Issues of Current Report

IP : 161.246.5.46	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High
18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Medium
57690	Terminal Services Encryption Level is Medium or Low	Medium
58453	Terminal Services Doesn't Use Network Level Authentication (NLA)	Medium
57582	SSL Self-Signed Certificate	Medium
51192	SSL Certificate Cannot Be Trusted	Medium
57608	SMB Signing Disabled	Medium
30218	Terminal Services Encryption Level is not FIPS-140 Compliant	Low
65821	SSL RC4 Cipher Suites Supported	Low

IP : 161.246.5.39	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
57608	SMB Signing Disabled	Medium

IP : 161.246.5.11	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical
57608	SMB Signing Disabled	Medium

IP : 161.246.4.3	OS : OpenBSD 5	
Plugin ID	**Details**	**Risk Level**
10539	DNS Server Recursive Query Cache Poisoning Weakness	Medium
12217	DNS Server Cache Snooping Remote Information Disclosure	Medium
35450	DNS Server Spoofed Request Amplification DDoS	Medium

Current Report : test_multi.nessus

รูปที่ 5.14 ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานปัจจุบัน

Summary Issues of Previous Report

IP : 161.246.5.46	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High
18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Medium
57690	Terminal Services Encryption Level is Medium or Low	Medium
58453	Terminal Services Doesn't Use Network Level Authentication (NLA)	Medium
57582	SSL Self-Signed Certificate	Medium
51192	SSL Certificate Cannot Be Trusted	Medium
57608	SMB Signing Disabled	Medium
30218	Terminal Services Encryption Level is not FIPS-140 Compliant	Low
65821	SSL RC4 Cipher Suites Supported	Low

IP : 161.246.5.39	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
57608	SMB Signing Disabled	Medium

IP : 161.246.5.11	OS : Microsoft Windows 7 Professional	
Plugin ID	**Details**	**Risk Level**
53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical
57608	SMB Signing Disabled	Medium

IP : 161.246.4.3	OS : OpenBSD 5	
Plugin ID	**Details**	**Risk Level**
10539	DNS Server Recursive Query Cache Poisoning Weakness	Medium
12217	DNS Server Cache Snooping Remote Information Disclosure	Medium
35450	DNS Server Spoofed Request Amplification DDoS	Medium

Previous Report : test_multi.nessus

รูปที่ 5.15 ตารางสรุปจุดอ่อนแอของระบบของเอกสารรายงานก่อนหน้า

Vulnerabilities Comparison

Current Scan : test_multi_After_scan.nessus

Scanned date : 23 February 2014

Previous Scan : test_multi.nessus

Scanned date : 02 January 2014

Number	Nessus Plugin ID	Finding Issue	Risk Level	Previous Scan	Current Scan
1	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical	161.246.5.11	161.246.5.46
2	58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High	161.246.5.46	161.246.5.46
3	57608	SMB Signing Disabled	Medium	161.246.5.46 161.246.5.39 161.246.5.11	161.246.5.6 161.246.5.46 161.246.5.11
4	45411	SSL Certificate with Wrong Hostname	Medium		161.246.5.6
5	71428	PHP 5.5.x	Medium		161.246.5.6
6	11213	HTTP TRACE / TRACK Methods Allowed	Medium		161.246.5.6
7	51192	SSL Certificate Cannot Be Trusted	Medium	161.246.5.46	161.246.5.6 161.246.5.46
8	57582	SSL Self-Signed Certificate	Medium	161.246.5.46	161.246.5.6 161.246.5.46
9	18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Medium	161.246.5.46	161.246.5.46
10	57690	Terminal Services Encryption Level is Medium or Low	Medium	161.246.5.46	161.246.5.46
11	58453	Terminal Services Doesn't Use Network Level Authentication (NLA)	Medium	161.246.5.46	161.246.5.46
12	12217	DNS Server Cache Snooping Remote Information Disclosure	Medium	161.246.4.3	161.246.4.3
13	10539	DNS Server Recursive Query Cache Poisoning Weakness	Medium	161.246.4.3	161.246.4.3

รูปที่ 5.16 ตารางแสดงการเปรียบเทียบจุดอ่อนแอของระบบ

Appendix
Details of Vulnerabilities Scanning Result

Vulnerability # 1	DNS Server Cache Snooping Remote Information Disclosure(Has not been resolved)
Plugin ID	12217
Risk Level	Medium
Synopsis	The remote DNS server is vulnerable to cache snooping attacks.
Description	The remote DNS server responds to queries for third-party domains that do not have the recursion bit set. This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited. For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of that financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more. Note: If this is an internal DNS server not accessable to outside networks, attacks would be limited to the internal network. This may include employees, consultants and potentially users on a guest network or WiFi connection if supported.
Solution	Contact the vendor of the DNS software for a fix.
See Also	http://www.rootsecure.net/content/downloads/pdf/dns_cache_snooping.pdf
List of Hosts	161.246.4.3 (udp/53)
Risk Factor	CVE : BID : Other References :
CVSS Score	CVSS Base Score : 5.0 CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N CVSS Temporal Score :
Exploit With	
Vulnerability Publication Date	

Vulnerability # 2	DNS Server Recursive Query Cache Poisoning Weakness(Has not been resolved)
Plugin ID	10539
Risk Level	Medium
Synopsis	The remote name server allows recursive queries to be performed by the host running nessusd.
Description	It is possible to query the remote name server for third party names. If this is your internal nameserver, then the attack vector may be limited to employees or guest access if allowed. If you are probing a remote nameserver, then it allows anyone to use it to resolve third party names (such as www.nessus.org). This allows attackers to perform cache poisoning attacks against this nameserver. If the host allows these recursive queries via UDP, then the host can be used to 'bounce' Denial of Service attacks against another network or system.

รูปที่ 5.17 ภาคผนวก

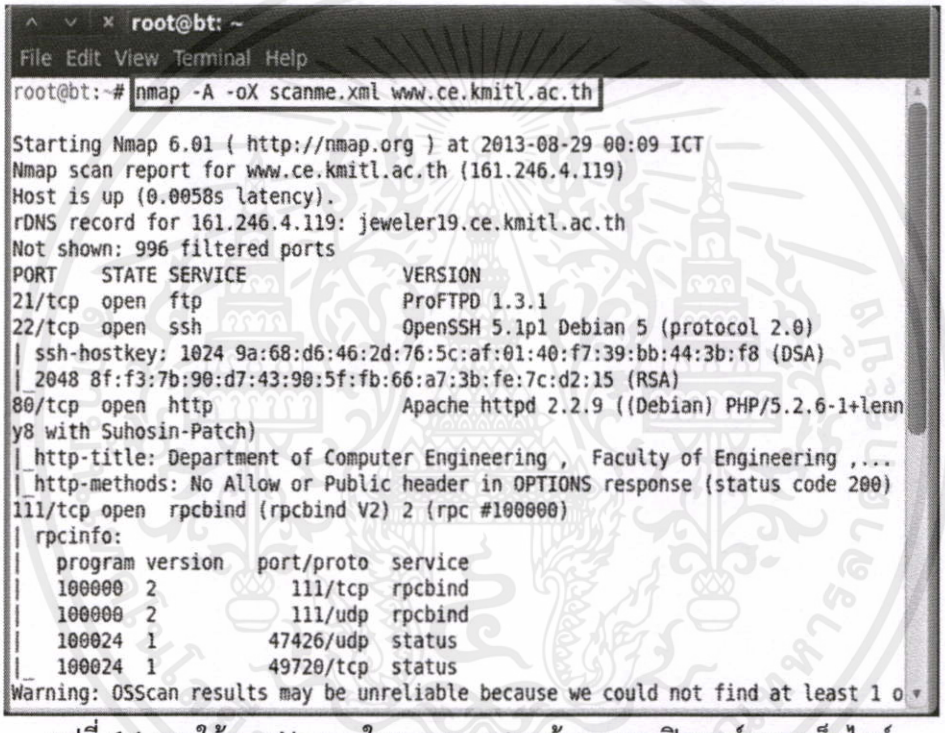
บทที่ 6
การทดลองและผลการทดลอง

ในการพัฒนาระบบการพัฒนาระบบต้นแบบเพื่อการสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยของระบบโดยอัตโนมัติ เพื่อใช้ร่วมกับเครื่องมือค้นหาจุดอ่อนแอ ได้ทำการศึกษาการพัฒนาระบบร่วมกับการทำงานของ Nmap และ Nessus โดยได้ทำการออกแบบการทำงานของระบบในการเก็บค่าข้อมูลจากเอกสารรายงานและการส่งออกค่าข้อมูลบนฐานข้อมูลจากเงื่อนไขความต้องการของผู้ใช้งานเพื่อใช้ในการออกเอกสารรายงาน

6.1 การทดลองร่วมกับ Nmap

6.1.1 การทดลองการออกเอกสารโดยการใช้งาน Nmap

เริ่มต้นด้วยการใช้โปรแกรม Nmap ทำการตรวจสอบข้อมูลการเปิดพอร์ตของเว็บไซต์ www.ce.kmitl.ac.th ด้วยการใช้คำสั่ง "*nmap –A –oX scanme.xml www.ce.kmitl.ac.th*" ซึ่งหมายถึงการตรวจสอบเว็บไซต์ www.ce.kmitl.ac.th โดยให้ทำการตรวจสอบเวอร์ชันของระบบปฏิบัติการของระบบ และให้ทำการส่งออกเอกสารรายงานในรูปแบบของไฟล์ .XML ชื่อ scanme.xml

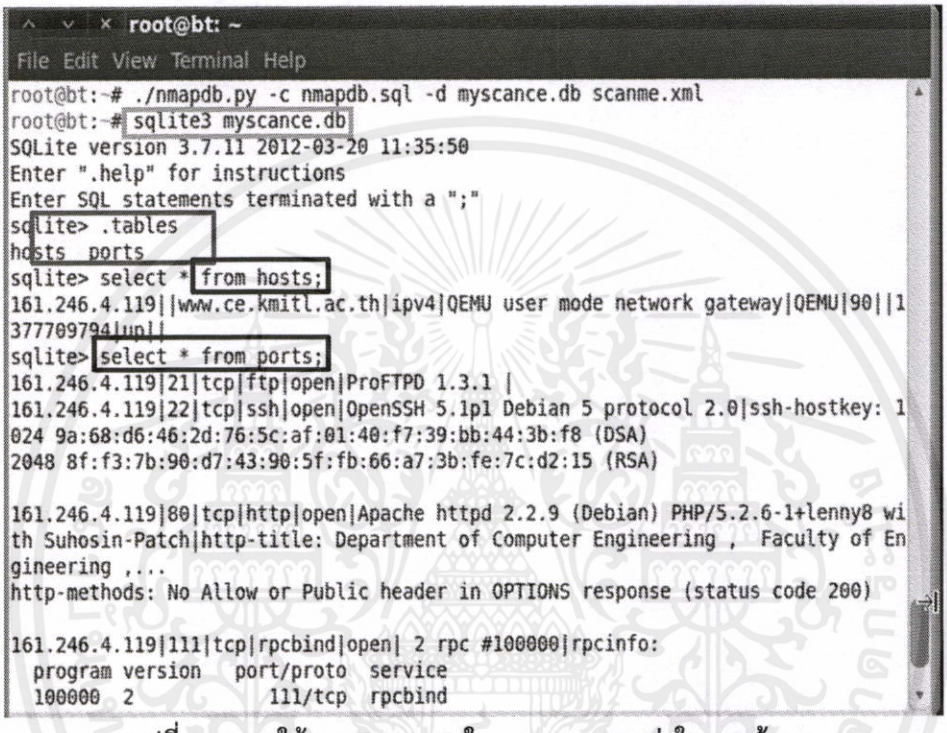


รูปที่ 6.1 การใช้งาน Nmap ในการตรวจสอบข้อมูลการเปิดพอร์ตของเว็บไซต์ www.ce.kmitl.ac.th

ทำการเรียกใช้งานระบบ โดยใช้คำสั่ง *"./nmapdb.py –c nmapdb.sql –d my scance.db scanme.xml"* ซึ่งหมายถึงการสร้างฐานข้อมูล myscance.db โดยใช้คำสั่งจาก nmapdb.sql และทำการนำข้อมูลจากไฟล์ scanme.xml เข้าสู่ฐานข้อมูล myscance.db

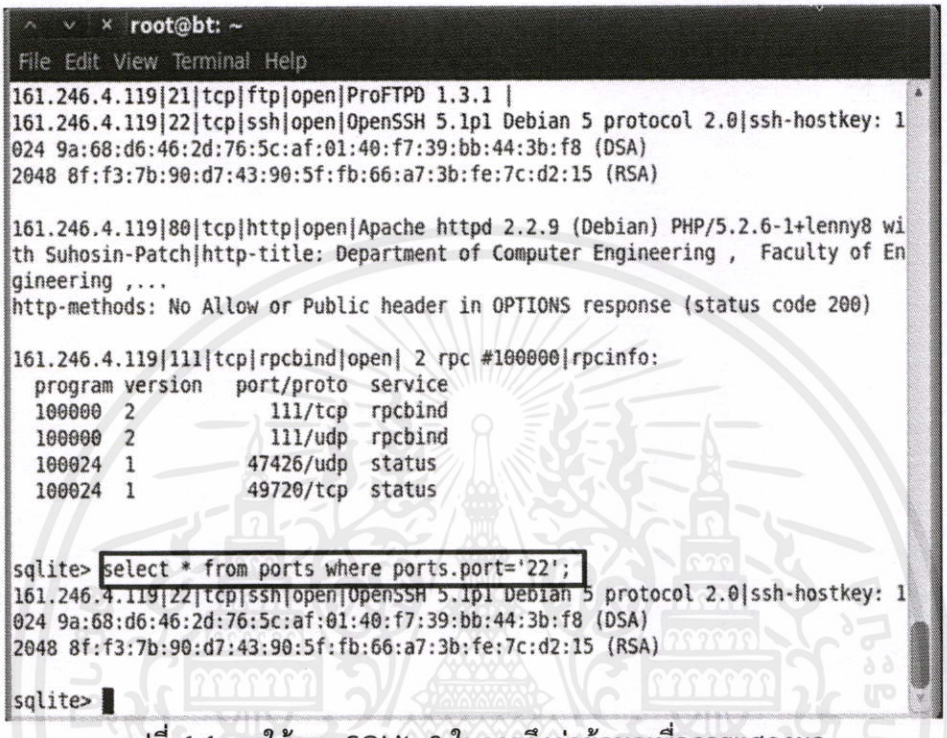
รูปที่ 6.2 การใช้คำสั่งงานเพื่อจะทำการนำเข้าข้อมูลจากไฟล์ XML เข้าสู่ฐานข้อมูล

ทำการเข้าสู่ฐานข้อมูลด้วยคำสั่ง *"sqlite3 myscance.db"* และทำการทดสอบการดูตารางในฐานข้อมูลด้วยการใช้คำ *".tables"* โดยจะได้รับผลลัพธ์ออกเป็นตาราง hosts และตาราง ports จากนั้นทำการทดสอบการดูค่าทั้งหมดภายในตาราง hosts ด้วยคำสั่ง *"select * from hosts;"* และการดูค่าทั้งหมดภายในตาราง ports ด้วยคำสั่ง *"select * from ports;"* ซึ่งจะได้ผลลัพธ์ดังรูปที่ 6.3



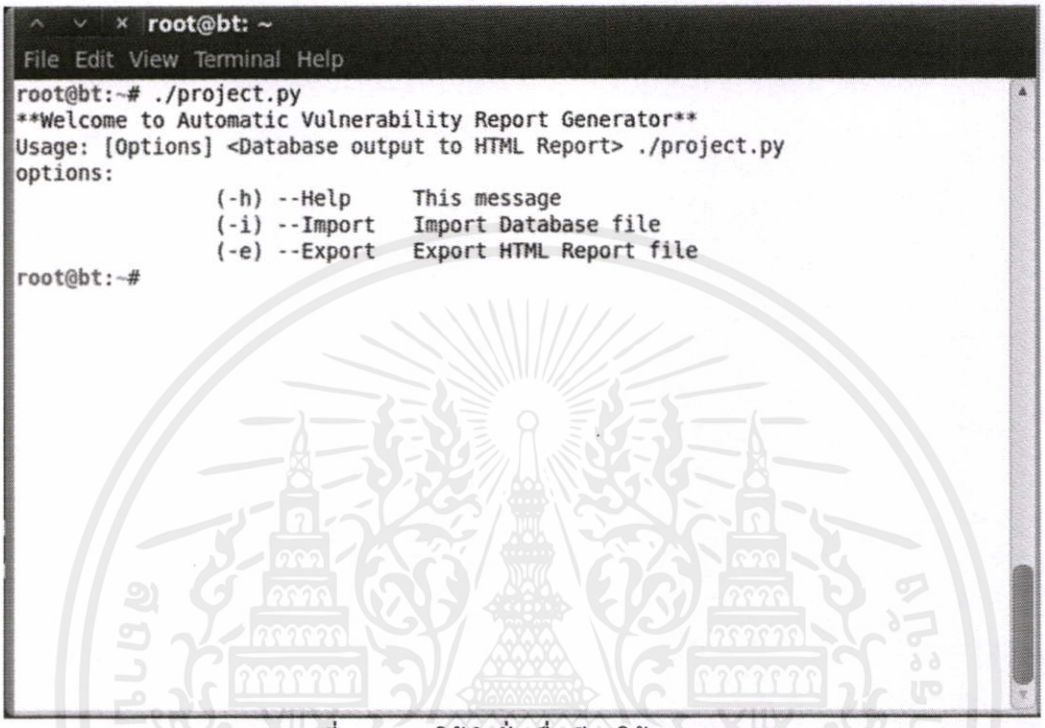
รูปที่ 6.3 การใช้งาน SQLite3 ในการตรวจสอบค่าในฐานข้อมูล

ทำการทดสอบการดูค่าทั้งหมดภายในตาราง ports ซึ่งมีค่าพอร์ตเท่ากับ 22 ด้วยคำสั่ง *"select * from ports where ports.port='22';"* ซึ่งจะได้ผลลัพธ์ดังรูปที่ 6.4



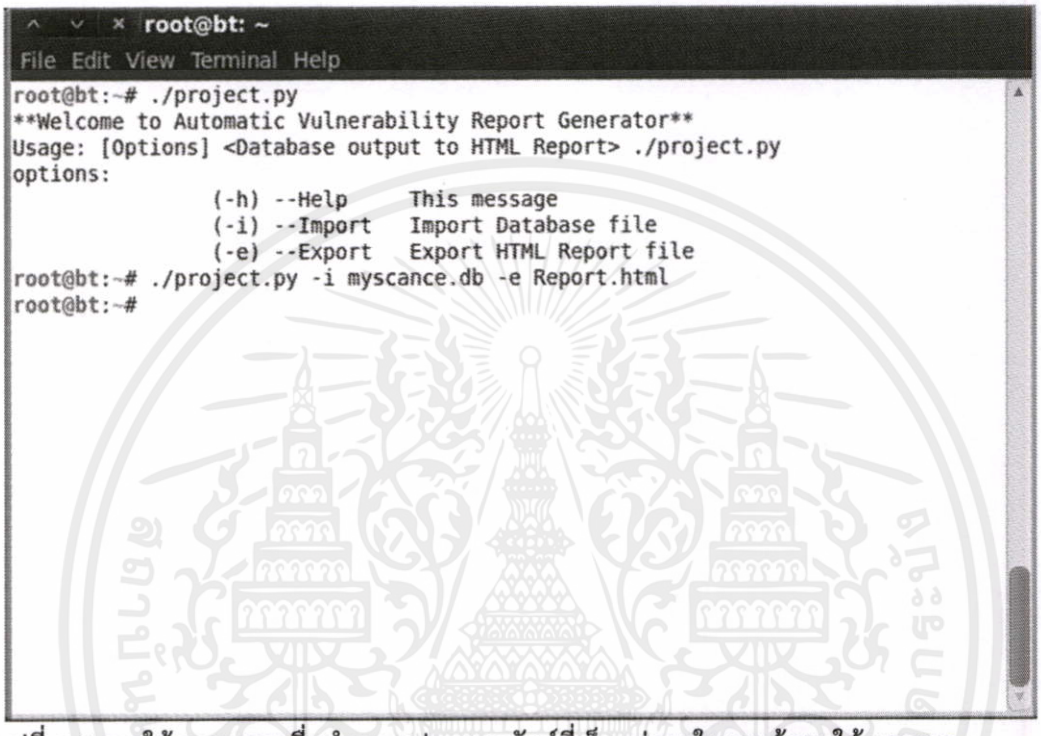
รูปที่ 6.4 การใช้งาน SQLite3 ในการดึงค่าข้อมูลเพื่อการแสดงผล

ทำการทดสอบการเรียกใช้งานระบบ โดยใช้คำสั่ง *"./project.py* ซึ่งระบบจะทำการแสดงข้อความเริ่มต้นของระบบ



รูปที่ 6.5 การใช้คำสั่งเพื่อเรียกใช้งานระบบ

ทำการเรียกใช้งานระบบ โดยใช้คำสั่ง *"./project.py –i myscance.db –e Report.html* ซึ่งหมายถึงการเรียกใช้งานฟังก์ชันการแปรผลลัพธ์ที่เก็บอยู่ภายในฐานข้อมูลให้แสดงผลออกเป็นเอกสารรายงานในรูปแบบของไฟล์ .HTML ในที่นี้คือไฟล์ Report.html



รูปที่ 6.6 การใช้งานระบบเพื่อทำการแปรผลผลลัพธ์ที่เก็บอยู่ภายในฐานข้อมูลให้แสดงผลออกเป็นเอกสารรายงานในรูปแบบของไฟล์ .HTML

เมื่อระบบได้ทำการประมวลผลเสร็จเรียบร้อยแล้วจะส่งออกเอกสารในรูปแบบเอกสารชื่อ Report.html ดังแสดงในรูปที่ 6.7

IP : 161.246.4.119

Hostname : www.ce.kmitl.ac.th

Port	Protocol	Name	State	Service	Info
21	tcp	ftp	open	ProFTPD 1.3.1	-
22	tcp	ssh	open	OpenSSH 5.1p1 Debian 5 protocol 2.0	-ssh-hostkey: 1024 9a:68:d6:46:2d:76:5c:af:01:40:f7:39:bb:44:3b:f8 (DSA) 2048 8f:f3:7b:90:d7:43:90:5f:fb:66:a7:3b:fe:7c:d2:15 (RSA)
80	tcp	http	open	Apache httpd 2.2.9 (Debian) PHP/5.2.6-1+lenny8 with Suhosin-Patch	-http-title: Department of Computer Engineering , Faculty of Engineering .... http-methods: No Allow or Public header in OPTIONS response (status code 200)
111	tcp	rpcbind	open	2 rpc #100000	-rpcinfo: program version port/proto service 100000 2 111/tcp rpcbind 100000 2 111/udp rpcbind 100024 1 47426/udp status 100024 1 49720/tcp status

รูปที่ 6.7 เอกสารรายงานที่อยู่ในรูปของไฟล์ .HTML

6.2 การทดลองร่วมกับ Nessus

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมด้วยการพิมพ์คำสั่ง Project.py หรือ Project.py –h จะได้หน้าต่างโปรแกรมที่มีการบอกถึงรายละเอียดวิธีการใช้งาน และ ตัวเลือกต่างๆในการใช้งาน ได้แก่

-h หรือ --help เป็นการแสดงหน้าต่างตัวเลือกการทำงานพร้อมคำอธิบาย

-i หรือ --input ตามด้วยชื่อไฟล์นามสกุล .Nessus จะเป็นการระบุชื่อไฟล์ที่จะนำมาสร้างรายงาน

-s หรือ --summary เป็นการเลือกรูปแบบของรายงานแบบ Results summary

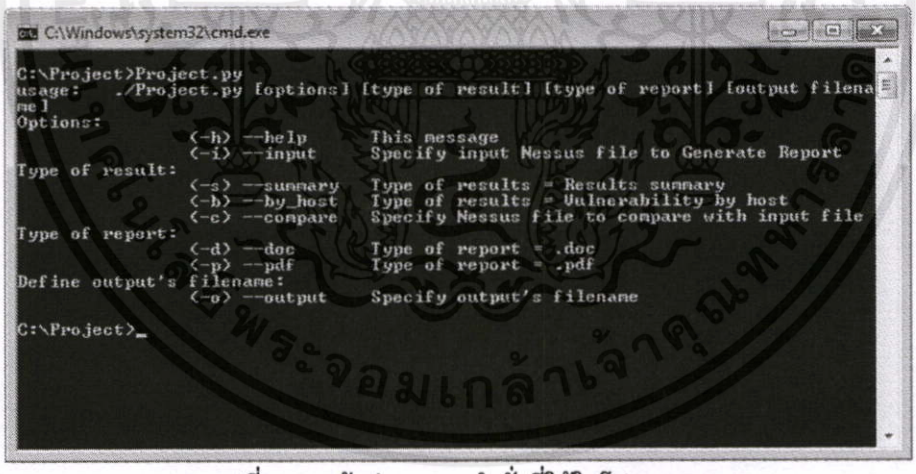
-b หรือ --by_host เป็นการเลือกรูปแบบของรายงานแบบ Vulnerability by host

-c หรือ --compare เป็นการเลือกรูปแบบของรายงานแบบ Compare โดยจะต้องระบุชื่อไฟล์นามสกุล .Nessus ที่จะนำมาเปรียบเทียบกับอินพุทไฟล์

-d หรือ --doc เป็นการเลือกให้รายงานที่สร้างนั้นออกเป็นไฟล์นามสกุล .DOC

-p หรือ --pdf เป็นการเลือกให้รายงานที่สร้างนั้นออกเป็นไฟล์นามสกุล .PDF

-o หรือ --output ตามด้วยชื่อไฟล์รายงานที่ต้องการจะสร้าง เป็นการระบุชื่อไฟล์รายงานตามที่ผู้ใช้ต้องการ

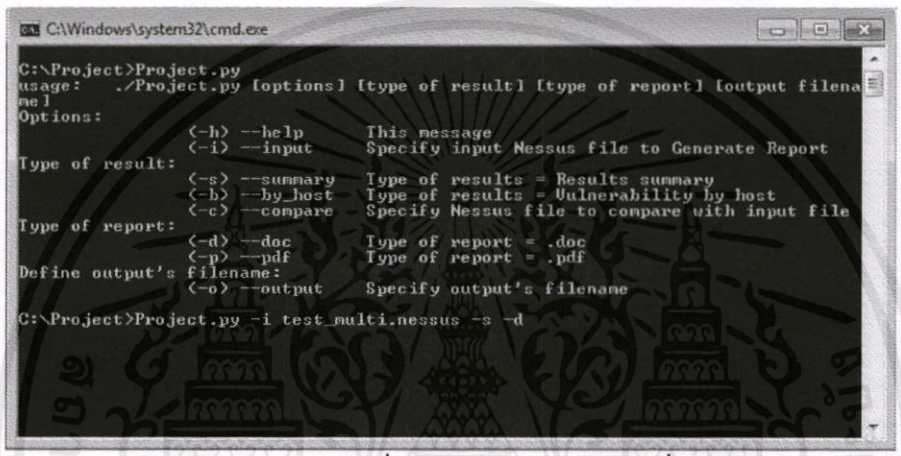


รูปที่ 6.8 หน้าต่างแสดงคำสั่งที่ใช้ในโปรแกรม

6.2.1 การทดลองการออกเอกสารรายงานในรูปแบบของไฟล์ .DOC

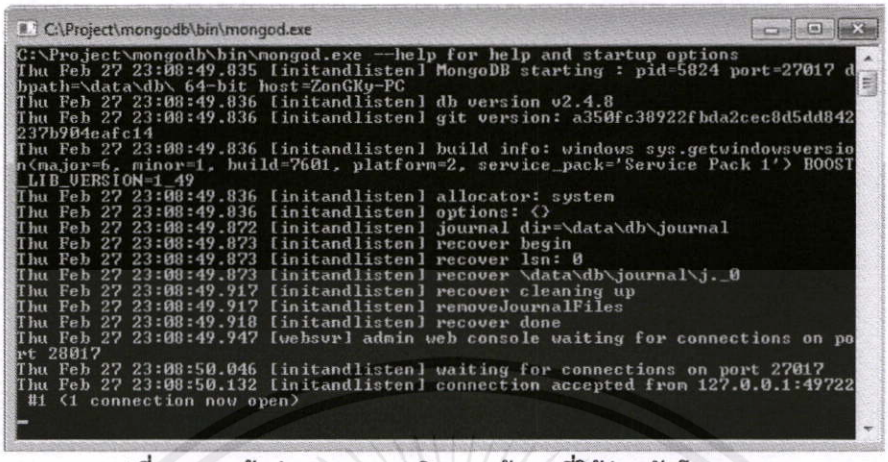
6.2.1.1 การออกเอกสารรายงานด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .DOC

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi.nessus -s -d จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi.nessus โดยเลือกรูปแบบรายงานเป็น Result summary และออกเอกสารในรูปแบบของไฟล์ .DOC



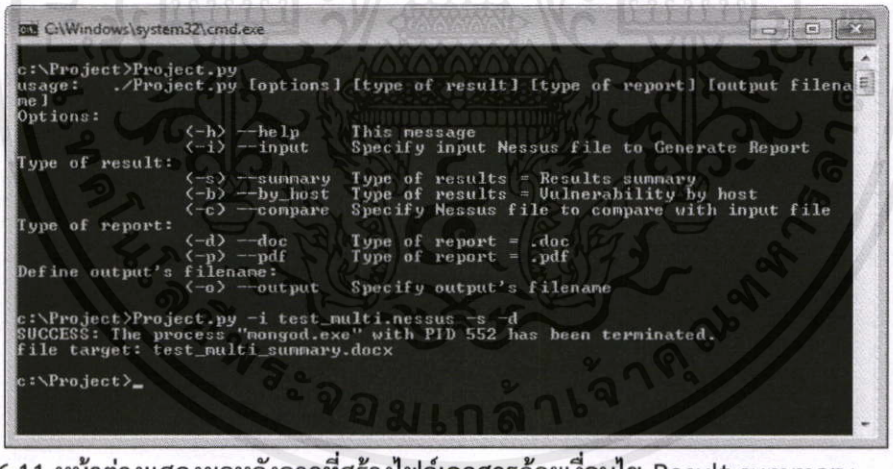
รูปที่ 6.9 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .DOC

เมื่อโปรแกรมเริ่มทำงานจะทำการเปิดฐานข้อมูลมองโกดีบี ซึ่งใช้ในการจัดเก็บข้อมูลที่อ่านได้จากอินพุทไฟล์ เพื่อช่วยในการจัดเรียงข้อมูลในขั้นตอนการสร้างเอกสารรายงาน



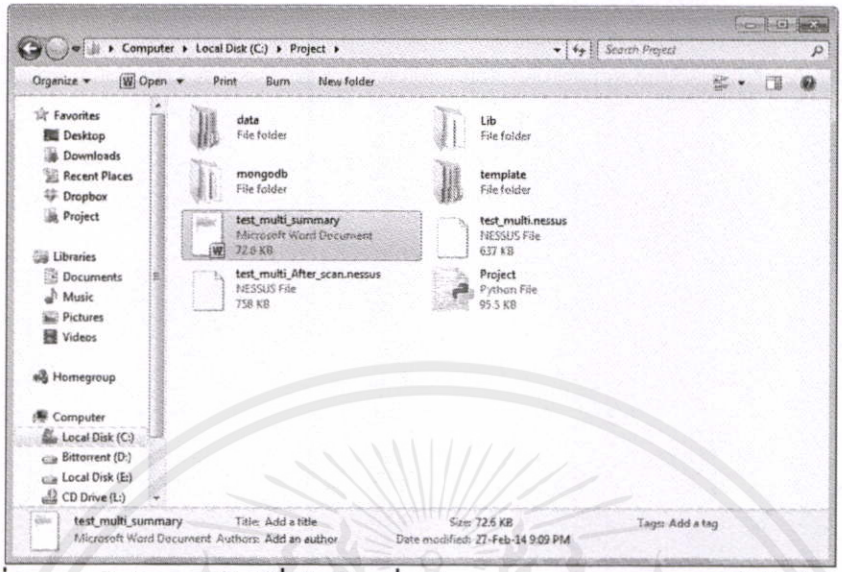
รูปที่ 6.10 หน้าต่างแสดงการเปิดฐานข้อมูลที่ใช้ร่วมกับโปรแกรม

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะทำการปิดฐานข้อมูลมองโกดีบีและระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_summary.docx ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว



รูปที่ 6.11 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .DOC เสร็จสิ้น

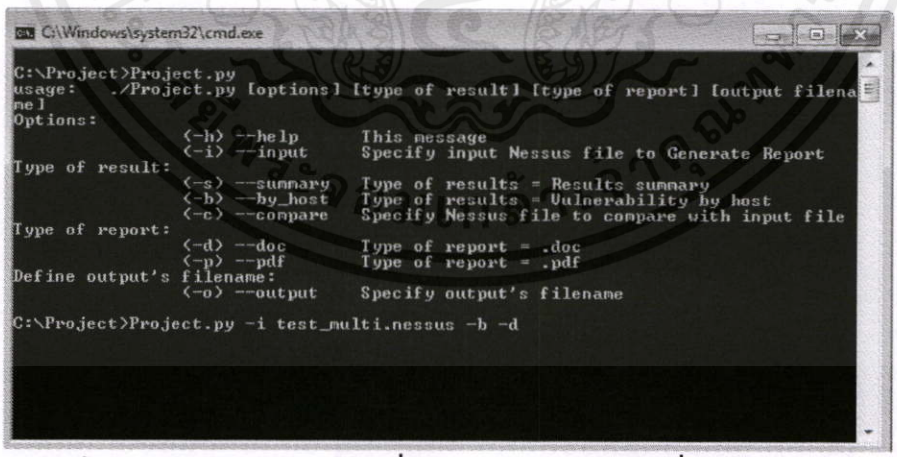
ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_summary.docx”



รูปที่ 6.12 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไข Result summary และเป็นไฟล์ .DOC

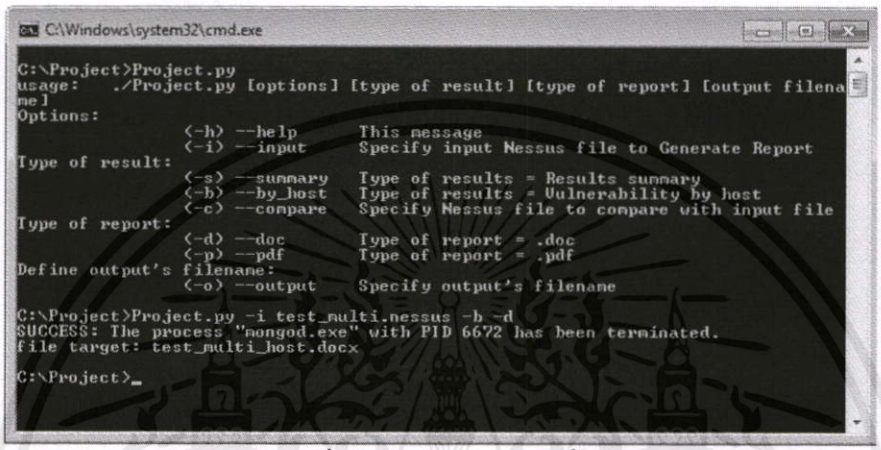
6.2.1.2 การออกเอกสารรายงานด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .DOC

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi.nessus -b -d จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi.nessus โดยเลือกรูปแบบรายงานเป็น Vulnerability by host และออกเอกสารในรูปแบบของไฟล์ .DOC



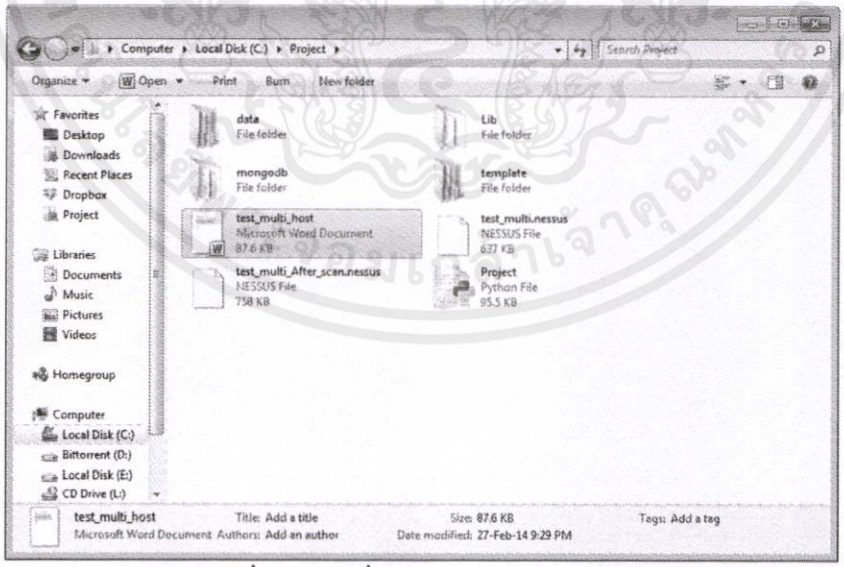
รูปที่ 6.13 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .DOC

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะมีการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_host.docx ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว



รูปที่ 6.14 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .DOC เสร็จสิ้น

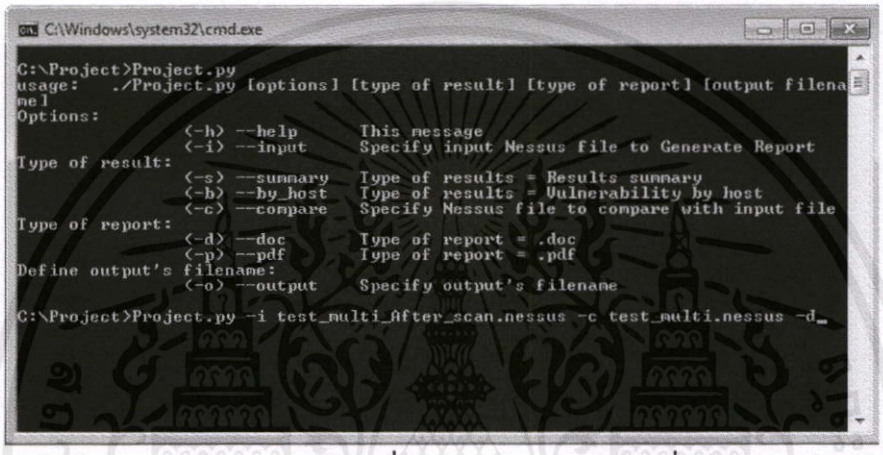
ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_host.docx”



รูปที่ 6.15 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไข Vulnerability by host และเป็นไฟล์ .DOC

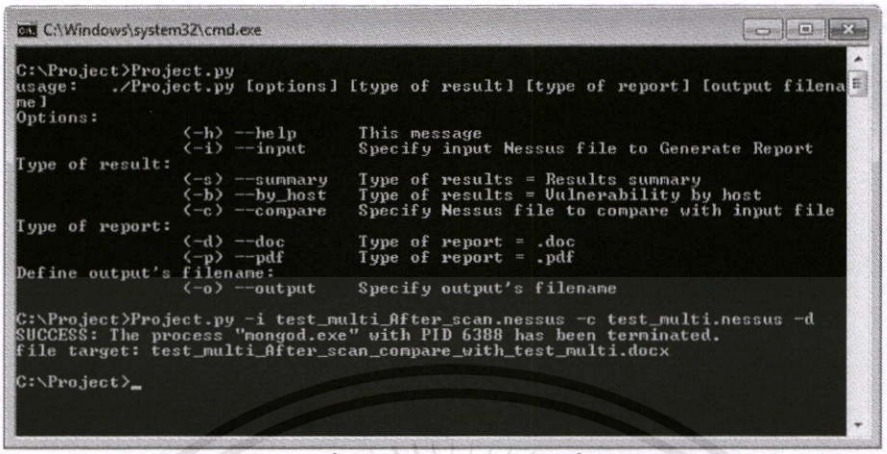
6.2.1.3 การออกเอกสารรายงานด้วยเงื่อนไขการเปรียบเทียบในรูปแบบของไฟล์ .DOC

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi_After_scan.nessus -c test_multi.nessus -d จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi_After_scan.nessus โดยเลือกรูปแบบรายงานเป็นการเปรียบเทียบ โดยจะเปรียบเทียบกับไฟล์ test_multi.nessus และออกเอกสารในรูปแบบของไฟล์ .DOC



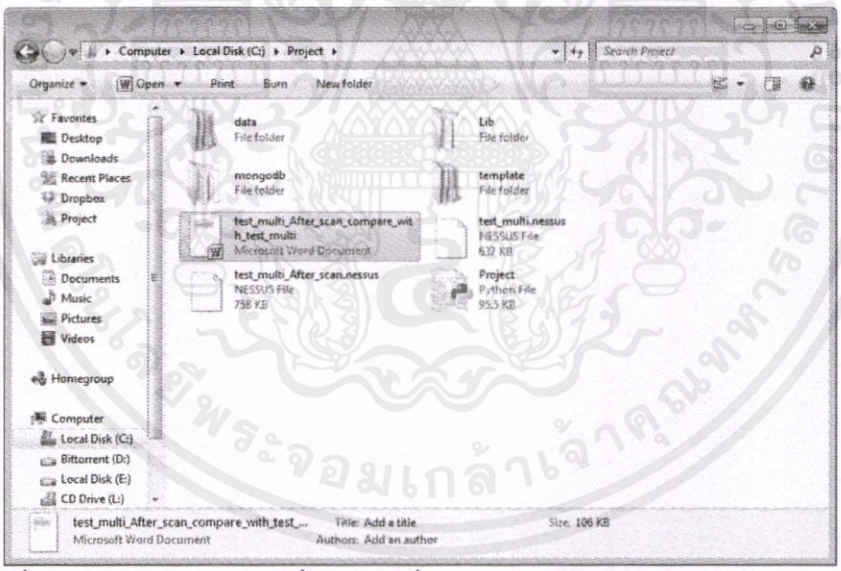
รูปที่ 6.16 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไขการเปรียบเทียบ ในรูปแบบของไฟล์ .DOC

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะมีการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_After_scan_compare_with_test_multi.docx ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว โดยในรูปแบบของการเปรียบเทียบจะมีการบอกว่าเปรียบเทียบกับไฟล์ใด



รูปที่ 6.17 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไขการเปรียบเทียบ ในรูปแบบของไฟล์ .DOC เสร็จสิ้น

ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_After_scan_compare_with_test_multi.docx”

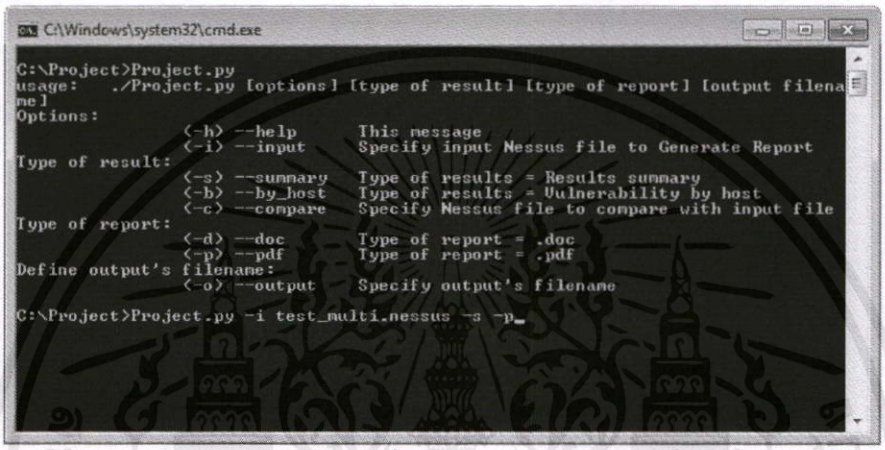


รูปที่ 6.18 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไขการเปรียบเทียบ และเป็นไฟล์ .DOC

6.2.2 การทดลองการออกเอกสารรายงานในรูปแบบของไฟล์ .PDF

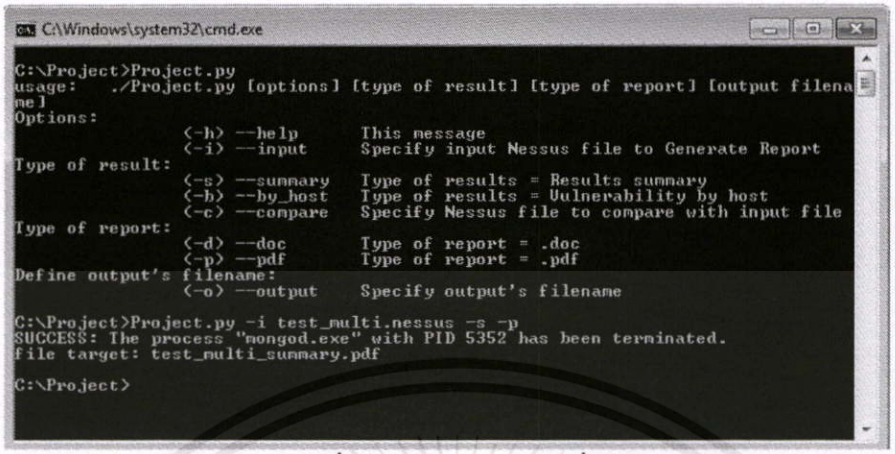
6.2.2.1 การออกเอกสารรายงานด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .PDF

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi.nessus -s -p จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi.nessus โดยเลือกรูปแบบรายงานเป็น Result summary และออกเอกสารในรูปแบบของไฟล์ .PDF



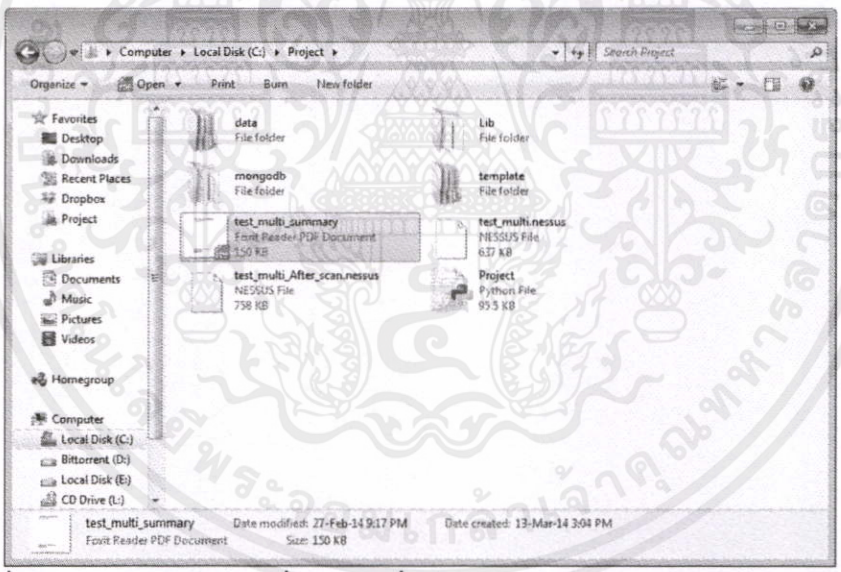
รูปที่ 6.19 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .PDF

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะทำการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_summary.pdf ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว



รูปที่ 6.20 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไข Result summary ในรูปแบบของไฟล์ .PDF เสร็จสิ้น

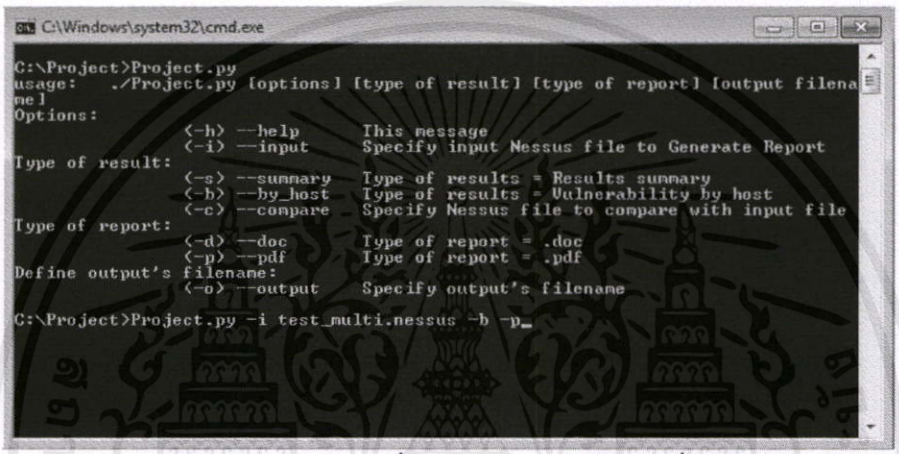
ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_summary.pdf”



รูปที่ 6.21 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไข Result summary และเป็นไฟล์ .PDF

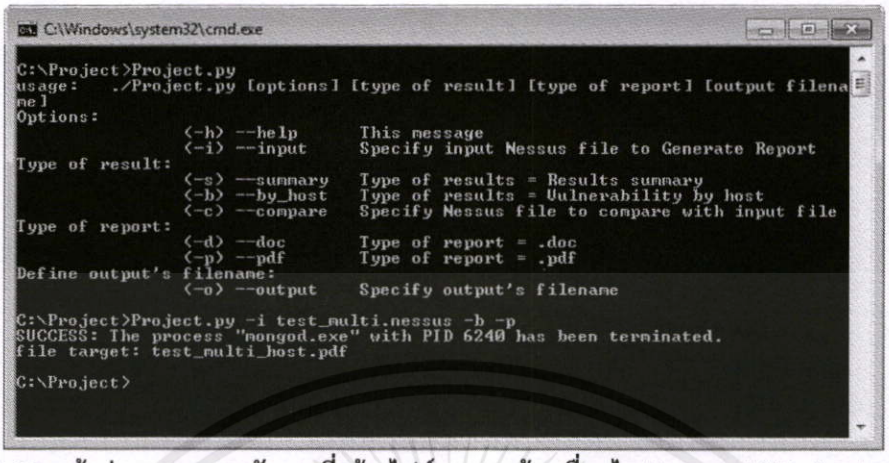
6.2.2.2 การออกเอกสารรายงานด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .PDF

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi.nessus -b -p จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi.nessus โดยเลือกรูปแบบรายงานเป็น Vulnerability by host และออกเอกสารในรูปแบบของไฟล์ .PDF



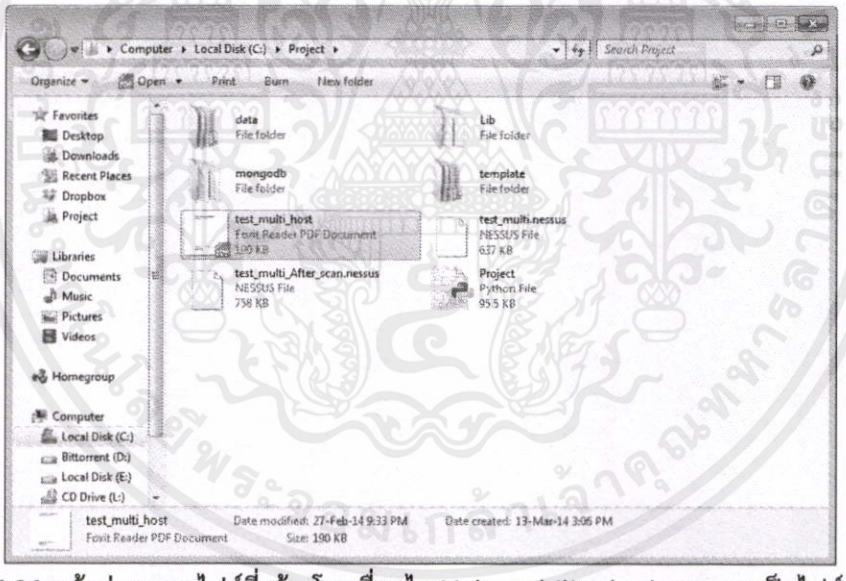
รูปที่ 6.22 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .PDF

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะมีการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_host.pdf ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว



รูปที่ 6.23 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไข Vulnerability by host ในรูปแบบของไฟล์ .PDF เสร็จสิ้น

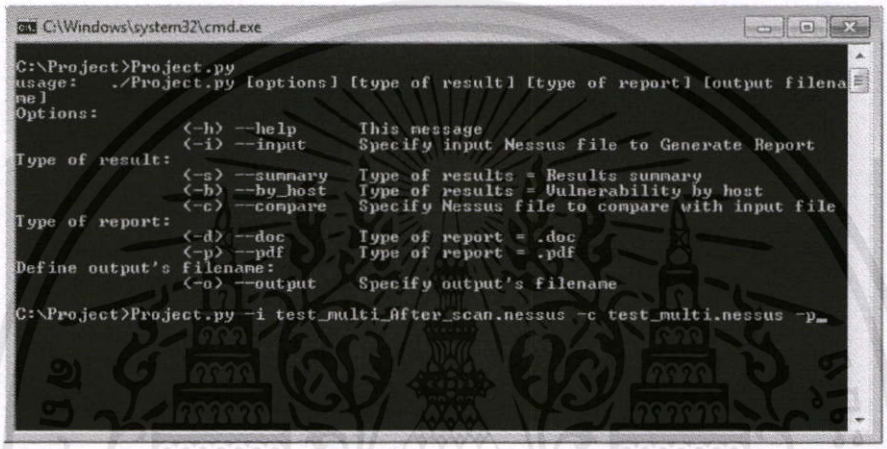
ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_host.pdf”



รูปที่ 6.24 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไข Vulnerability by host และเป็นไฟล์ .PDF

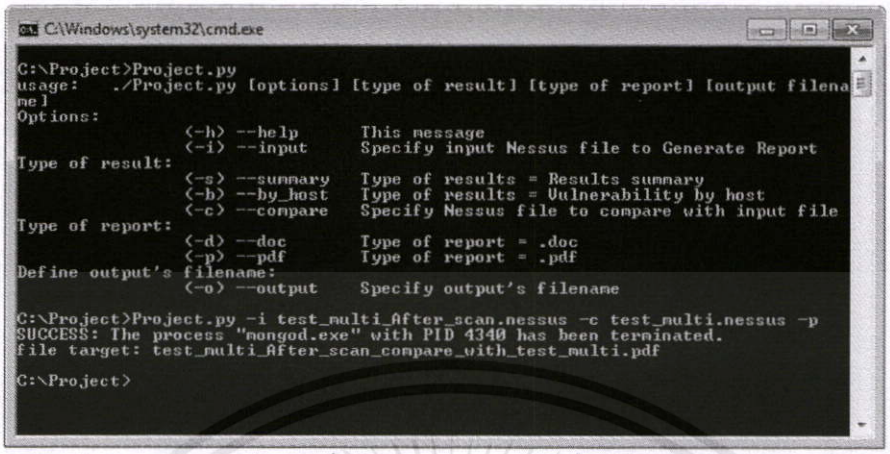
6.2.2.3 การออกเอกสารรายงานด้วยเงื่อนไขการเปรียบเทียบในรูปแบบของไฟล์ .PDF

ทำการเปิดหน้าต่างโปรแกรม Command Prompt เพื่อทำการเริ่มใช้งานโปรแกรมโดยการพิมพ์คำสั่ง Project.py -i test_multi_After_scan.nessus -c test_multi.nessus -p จะเป็นการสร้างเอกสารรายงานโดยใช้ข้อมูลจากไฟล์ test_multi_After_scan.nessus โดยเลือกรูปแบบรายงานเป็นการเปรียบเทียบ โดยจะเปรียบเทียบกับไฟล์ test_multi.nessus และออกเอกสารในรูปแบบของไฟล์ .PDF



รูปที่ 6.25 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารด้วยเงื่อนไขการเปรียบเทียบ ในรูปแบบของไฟล์ .PDF

เมื่อโปรแกรมสร้างเอกสารรายงานเสร็จสิ้น จะมีการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ test_multi_After_scan_compare_with_test_multi.pdf ซึ่งชื่อไฟล์นั้นได้จากการนำชื่อของอินพุทไฟล์ประกอบกับรูปแบบของรายงานที่เลือกในขั้นตอนที่แล้ว โดยในรูปแบบของการเปรียบเทียบจะมีการบอกว่าเปรียบเทียบกับไฟล์ใด

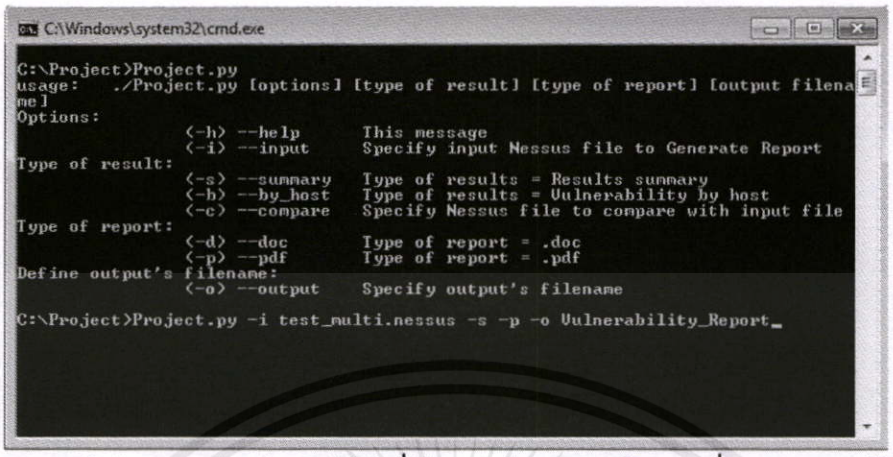


รูปที่ 6.26 หน้าต่างแสดงผลหลังจากที่สร้างไฟล์เอกสารด้วยเงื่อนไขการเปรียบเทียบ ในรูปแบบของไฟล์ .PDF เสร็จสิ้น

ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “test_multi_After_scan_compare_with_test_multi.pdf”

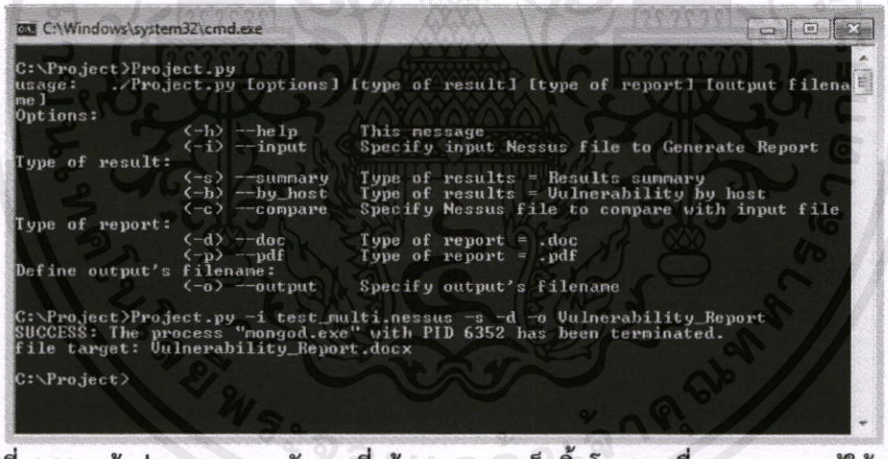
รูปที่ 6.27 หน้าต่างแสดงไฟล์ที่สร้างโดยเงื่อนไขการเปรียบเทียบ และเป็นไฟล์ .PDF

นอกจากนี้ยังสามารถกำหนดชื่อของเอกสารรายงานที่ต้องการจะออกได้ โดยใช้คำสั่ง -o หรือ --output ตามด้วยชื่อไฟล์เอกสารที่ต้องการ โดยในตัวอย่างเป็นการตั้งชื่อไฟล์เอกสารชื่อว่า Vulnerabiliy_Report



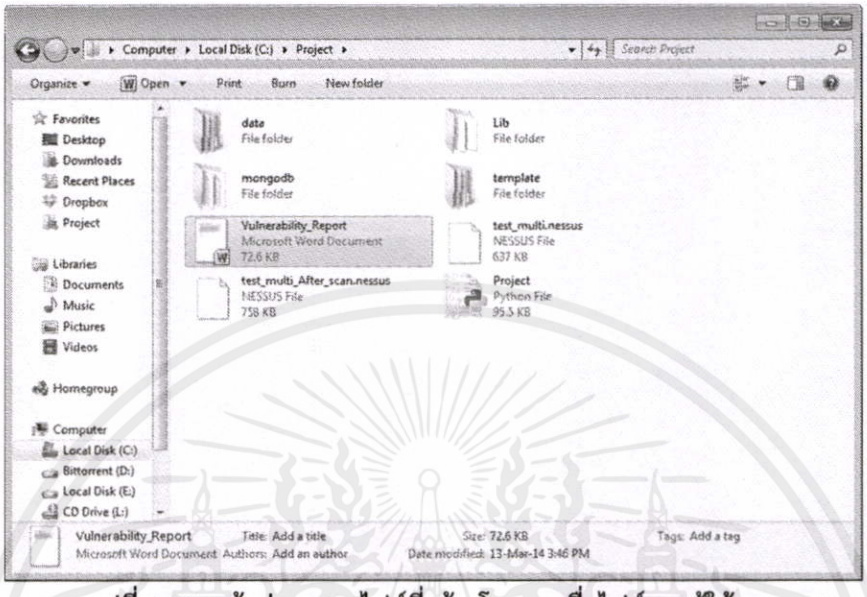
รูปที่ 6.28 หน้าต่างแสดงตัวอย่างการใช้คำสั่งในการสร้างเอกสารโดยระบุชื่อเอกสารจากผู้ใช้งาน

เมื่อโปรแกรมสร้างไฟล์เอกสารรายงานเสร็จสิ้น จะมีการระบุชื่อไฟล์เอกสารรายงานที่สร้างขึ้นจากโปรแกรม โดยในที่นี้จะได้ไฟล์เอกสารรายงานชื่อ Vulnerabiliy_Report.docx ซึ่งได้กำหนดไว้



รูปที่ 6.29 หน้าต่างแสดงผลหลังจากที่สร้างเอกสารเสร็จสิ้นโดยระบุชื่อเอกสารจากผู้ใช้งาน

ผลลัพท์ที่ได้คือ ไฟล์เอกสารรายงานที่มีชื่อว่า “Vulnerabiliy_Report.docx”



รูปที่ 6.30 หน้าต่างแสดงไฟล์ที่สร้างโดยระบุชื่อไฟล์จากผู้ใช้งาน

6.2.3 สรุปผลการทดลอง

จากการทดลองสามารถสรุปข้อดีของการออกเอกสารรายงานแต่ละรูปแบบได้ดังนี้

1) Result Summary

สามารถออกเอกสารรายงานโดยการสรุปข้อมูลจุดอ่อนแอของจำนวนเครื่องทั้งหมดที่ได้รับการตรวจสอบและสามารถบอกได้ว่าจุดอ่อนชนิดหนึ่งมีเครื่องไหนในระบบที่มีจุดอ่อนแอนี้ในระบบ

Vulnerability # 41	DCE Services Enumeration
Plugin ID	10736
Risk Level	None
Synopsis	A DCE/RPC service is running on the remote host.
Description	By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution	n/a
See Also	
List of Hosts	161.246.5.46 (tcp/49152) 161.246.5.46 (tcp/49153) 161.246.5.46 (tcp/49165) 161.246.5.46 (tcp/49161) 161.246.5.46 (tcp/49181) 161.246.5.46 (tcp/49154) 161.246.5.46 (tcp/49157) 161.246.5.46 (tcp/445) 161.246.5.46 (tcp/135)
Risk Factor	CVE : BID : Other References :
CVSS Score	CVSS Base Score : CVSS Temporal Score :
Exploit With	
Vulnerability Publication Date	

รูปที่ 6.31 ตารางแสดงรายละเอียดจุดอ่อนแอของการออกเอกสารรายงานด้วยเงื่อนไข Result Summary

2) Vulnerability by host

มีการจัดเรียงข้อมูลจุดอ่อนแอต่าง ๆ ตามเครื่องที่ได้รับการตรวจสอบ และจัดเรียงผลตามลำดับความเสี่ยง

Vulnerability # 1	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
Plugin ID	53514
Risk Level	Critical
Synopsis	Arbitrary code can be executed on the remote host through the installed Windows DNS client.
Description	A flaw in the way the installed Windows DNS client processes Link- local Multicast Name Resolution (LLMNR) queries can be exploited to execute arbitrary code in the context of the NetworkService account. Note that Windows XP and 2003 do not support LLMNR and successful exploitation on those platforms requires local access and the ability to run a special application. On Windows Vista, 2008, 7, and 2008 R2, however, the issue can be exploited remotely.
Solution	Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2.
See Also	http://technet.microsoft.com/en-us/security/bulletin/ms11-030
List of Hosts	161.246.5.11 (udp/5355)
Risk Factor	CVE : CVE-2011-0657 BID : BID:47242 Other References : OSVDB:71780 IAVA:2011-A-0039 MSFT:MS11-030
CVSS Score	CVSS Base Score : 10.0 CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C CVSS Temporal Score : 7.8 CVSS2#E:POC/RL:OF/RC:C
Exploit With	Core Impact (true) Metasploit (true)
Vulnerability Publication Date	2011/04/12

Vulnerability # 2	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)
Plugin ID	58435
Risk Level	High
Synopsis	The remote Windows host could allow arbitrary code execution.
Description	An arbitrary remote code vulnerability exists in the implementation of the Remote Desktop Protocol (RDP) on the remote Windows host. The vulnerability is due to the way that RDP accesses an object in memory that has been improperly initialized or has been deleted. If RDP has been enabled on the affected system, an unauthenticated, remote attacker could leverage this vulnerability to cause the system to execute arbitrary code by sending a sequence of specially crafted RDP packets to it. This plugin also checks for a denial of service vulnerability in Microsoft Terminal Server. Note that this script does not detect the vulnerability if the 'Allow connections only from computers running Remote Desktop with Network Level Authentication' setting is enabled or the security layer is set to 'SSL (TLS 1.0)' on the remote host.
Solution	Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2. Note that an extended support contract with Microsoft is required to obtain the patch for this vulnerability for Windows 2000.
See Also	http://technet.microsoft.com/en-us/security/bulletin/ms12-020

รูปที่ 6.31 ตารางแสดงรายละเอียดจุดอ่อนแอของการออกเอกสารรายงานด้วยเงื่อนไข Vulnerability by host

3) Compare with previous report

สามารถทำการเปรียบเทียบเอกสารรายงานโดยมีตารางเปรียบเทียบจุดอ่อนแอว่าจุดอ่อนแอนั้นได้รับการแก้ไขแล้วหรือไม่ หรือมีจุดอ่อนแอใดเกิดขึ้นใหม่ในระบบ

Vulnerabilities Comparison

Current Scan : test_multi_After_scan.nessus

Scanned date : 23 February 2014

Previous Scan : test_multi.nessus

Scanned date : 02 January 2014

Number	Nessus Plugin ID	Finding Issue	Risk Level	Previous Scan	Current Scan
1	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)	Critical	161.246.5.11	161.246.5.46
2	58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)	High	161.246.5.46	161.246.5.46
3	57608	SMB Signing Disabled	Medium	161.246.5.46 161.246.5.39 161.246.5.11	161.246.5.6 161.246.5.46 161.246.5.11

รูปที่ 6.32 ตารางแสดงการเปรียบเทียบจุดอ่อนแอของระบบ

บทที่ 7
บทสรุปและข้อเสนอแนะ

7.1 บทสรุป

โครงงานนี้เป็นการพัฒนาระบบต้นแบบเพื่อการสร้างรายงานจุดอ่อนแอด้านการรักษาความปลอดภัยของระบบโดยอัตโนมัติ เพื่อใช้ร่วมกับเครื่องมือค้นหาจุดอ่อนแอและสามารถปรับแต่งเอกสารให้อยู่ในรูปแบบที่สามารถนำไปใช้งานได้อย่างเหมาะสมและมีประสิทธิภาพ ซึ่งใช้ภาษาไพทอน (Python) ในการพัฒนาระบบ โดยมีการใช้มองโกดีบี (MongoDB) ในการพัฒนาระบบฐานข้อมูลเพื่อใช้ในการเก็บข้อมูลสำคัญจากเอกสารรายงานที่ผู้ใช้งานได้นำเข้าสู่ระบบ โดยได้ทำการศึกษาการพัฒนาระบบร่วมกับการทำงานของ Nessus ในส่วนของการทำงานได้ทำการออกแบบการทำงานของระบบ ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

1) การนำเข้าเอกสารรายงานจากผู้ใช้งาน

เป็นการนำเข้าเอกสารรายงานจากเครื่องมือค้นหาจุดอ่อนของระบบที่มีชนิดของรายงานเป็น .Nessus เท่านั้น โดยเมื่อผู้ใช้งานทำการนำเข้าเอกสารรายงาน ระบบจะทำการนำข้อมูลต่าง ๆ จากเอกสารรายงานไปเก็บไว้บนฐานข้อมูล

2) การรับค่าเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงาน

เป็นการรับค่าเงื่อนไขในการจัดเรียงเนื้อหาของเอกสารรายงานจากผู้ใช้งาน

3) การประมวลผลข้อมูลจากฐานข้อมูล

ระบบจะนำข้อมูลเงื่อนไขความต้องการจากผู้ใช้งานมาประมวลผลร่วมกับข้อมูลบนฐานข้อมูลเพื่อนำข้อมูลต่าง ๆ ที่ตรงกับเงื่อนไขเพื่อออกเอกสารรายงาน

4) การส่งออกเอกสารรายงาน

เป็นการส่งออกเอกสารรายงานที่ตรงกับความต้องการของผู้ใช้งาน

7.2 ปัญหาอุปสรรคและแนวทางแก้ไข

1) การหาข้อมูลในการพัฒนาระบบเป็นไปได้ยาก
2) ต้องมีการสอบถามถึงรายละเอียดของการทำงานและรูปแบบของเอกสารรายงานจากผู้ทำงานที่เกี่ยวข้อง เพื่อการออกแบบรูปแบบของเอกสารรายงานที่เป็นที่น่าสนใจของผู้ใช้งาน
3) ความจำกัดของการตรวจสอบระบบหรือโฮสต์ที่สามารถตรวจสอบได้อย่างถูกต้อง
4) การพัฒนาระบบมีการพัฒนาด้วยภาษาไพทอน (Python) ซึ่งเป็นภาษาที่ใหม่ จึงต้องใช้เวลาในการเรียนรู้และพัฒนาระบบ

7.3 แนวทางการพัฒนาต่อ

1) เพิ่มการทำงานของระบบโดยทำการวบรรวมจุดอ่อนแอที่คล้ายกันให้แสดงผลเป็นเวอร์ชันปัจจุบัน
2) พัฒนาการทำงานให้โปรแกรมสามารถรองรับการทำงานร่วมกับเอกสารรายงานจากเครื่องมือชนิดอื่นได้

บรรณานุกรม

[1] Nectec. **"เอกสารประกอบการศึกษาองค์ความรู้ เรื่องการใช้ภาษา XML (Extensive Markup Language)."** [Online]. Available : http://wiki.nectec.or.th/gitiwiki/bin/viewfile/Pub/PoliceShareKnowledge?rev=1;filename=20070824_%E0%CD%A1%CA%D2%C3%BB%C3%D0%A1%CD%BA_XML_v6_Revise.doc. 2007.

[2] โชติพันธุ์ หล่อเลิศสุนทร. **คู่มือเรียน เขียนโปรแกรม Python (ภาคปฏิบัติ)** กรุงเทพมหานคร : สำนักพิมพ์คอร์ฟังก์ชัน. 2554

[3] Joel Shprentz. **"Creating HTML Documents with Python"** [Online]. Available : http://www.python.org/workshops/1995-12/papers/shprentz.html. 1995

[4] MongoDB. **"Inc. The MongoDB 2.4 Manual"** [Online]. Available : http://docs.mongodb.org/manual. 2012

[5] Reportlab. **"Reportlab User Giude"** www.reportlab.com/docs/reportlab-userguide.pdf. 2014