การประเมินความเสี่ยงต่อคอมมานด์อินเจคชันและการปฏิเสธการให้บริการโดยอิงสกีมาข้อมูลของเว็บเซอร์วิซ

นางสาววันวิษา โพธิ์เจริญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์มหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

A RISK ASSESSMENT AGAINST COMMAND INJECTION AND DENIAL OF SERVICE BASED ON DATA SCHEMAS OF WEB SERVICES

Miss Wanwisa Phocharoen

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

หัวข้อวิทยานิพนธ์	การประเมินความเสี่ยงต่อคอมมานด์อินเจคชันและการปฏิเสธการให้บริการโดยอิงสกีมาข้อมูลของเว็บเซอร์วิซ
โดย	นางสาววันวิษา โพธิ์เจริญ
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

……คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…….. ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

……อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

……กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ชวลิต ศรีสถาพรพัฒน์)

วันวิษา โพธิ์เจริญ : การประเมินความเสี่ยงต่อคอมมานด์อินเจคชันและการปฏิเสธการให้บริการโดยอิงสกีมาข้อมูลของเว็บเซอร์วิซ. (A RISK ASSESSMENT AGAINST COMMAND INJECTION AND DENIAL OF SERVICE BASED ON DATA SCHEMAS OF WEB SERVICES)
อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา, 75 หน้า.

ปัญหาด้านความมั่นคงเป็นปัญหาที่ผู้ให้บริการและผู้ใช้เว็บเซอร์วิซมีความกังวล เพราะเว็บเซอร์วิซมีโอกาสเสี่ยงที่จะถูกโจมตีจากผู้ประสงค์ร้ายต่อระบบ ซึ่งประกอบไปด้วยการปลอมแปลงข้อมูล การเปิดเผยข้อมูล การแก้ไขดัดแปลงข้อมูล การหยุดให้บริการข้อมูล และการละเมิดข้อมูล โดยเฉพาะหากข้อมูลนำเข้าในสกีมามีการออกแบบที่ไม่เข้มงวด อาจจะเป็นช่องโหว่ทำให้เว็บเซอร์วิซมีความเสี่ยงต่อการโจมตีจากคอมมานด์อินเจคชันและการปฏิเสธการให้บริการ งานวิจัยนี้จึงได้นำเสนอการประเมินความเสี่ยงต่อการโจมตีสำหรับเว็บเซอร์วิซในเบื้องต้น การประเมินจะเริ่มต้นด้วยการวิเคราะห์ข้อมูลนำเข้าที่กำหนดไว้ในสกีมาซึ่งได้อธิบายไว้ในเอกสารวิสเดิล เพื่อตรวจสอบว่าข้อมูลเหล่านี้ได้มีการกำหนดข้อบังคับไว้หรือไม่และมีความเสี่ยงต่อการโจมตีแบบคอมมานด์อินเจคชันและแบบการปฏิเสธการให้บริการอย่างไร จากนั้นจะทำการตรวจสอบว่าความเสี่ยงดังกล่าวสามารถทำให้บรรเทาลงได้หรือไม่โดยพิจารณาจากข้อมูลเชิงความหมายที่กำกับไว้ที่ข้อมูลนำเข้าในวิสเดิล หากข้อมูลเชิงความหมายที่กำกับไว้มีความเข้มงวดมากกว่าสกีมาของข้อมูลนำเข้า งานวิจัยจะถือว่าเป็นกรณีของการออกแบบที่หละหลวมของส่วนต่อประสานของเว็บเซอร์วิซ ซึ่งหากทำการออกแบบใหม่ให้สกีมาของข้อมูลมีความเข้มงวดขึ้น จะช่วยลดความเสี่ยงจากการถูกโจมตีได้ นอกจากนี้งานวิจัยนี้ยังนำเสนอแบบจำลองการประเมินความเสี่ยงสำหรับประเมินระดับความเสี่ยงต่อการโจมตีสำหรับเว็บเซอร์วิซ เพื่อเป็นแนวทางให้กับผู้ให้บริการในการปรับสกีมาให้เข้มงวดขึ้น และเป็นแนวทางให้กับผู้ใช้บริการในการเลือกใช้เว็บเซอร์วิซรายต่าง ๆ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ. ที่ปรึกษาวิทยานิพนธ์หลัก .....
ปีการศึกษา 2555 .....

5371443721 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: RISK ASSESSMENT / SECURITY ATTACKS / WEB SERVICES / ONTOLOGY

WANWISA PHOCHAROEN : A RISK ASSESSMENT AGAINST COMMAND INJECTION AND DENIAL OF SERVICE BASED ON DATA SCHEMAS OF WEB SERVICES. ADVISOR : ASSOC. PROF. TWITTIE SENIVONGSE, Ph.D., 75 pp.

Security concerns have been raised by Web services providers and consumers since Web services are vulnerable to various security attacks including counterfeiting, disclosure, tampering, disruption, and breach of information. In particular, Web services can be vulnerable if the schemas of the input data are not strong, giving way to security attacks like command injection and denial of service. This research proposes an initial assessment of security attack risks for Web services. The assessment begins with an analysis of the input data schemas that are described in the service WSDL document to determine if they are unconstrained and at risk of command injection and denial of service attacks. Then we determine if such a risk can be mitigated by making use of semantic information that is annotated to the input data elements within the WSDL. If the semantic annotation is stronger than the schema elements themselves, we refer to the case of weak interface design in which a redesign of the service interface with stronger schemas should help reduce attack risks. We also propose a risk assessment model for determining quantitatively the attack risk level of a Web service to guide the provider when considering schema hardening and the consumer when selecting between different services.

Department : Computer Engineering Student's Signature .....

Field of Study : Software Engineering Advisor's Signature .....

Academic Year : 2012 .....

กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ซึ่งเป็นผู้ให้ความช่วยเหลือ คำปรึกษา ข้อคิด แนวทางในการทำวิจัย ตลอดจนเป็นผู้ตรวจทานแก้ไขและเสียสละเวลาให้ความอดทนและใส่ใจแก่ผู้วิจัย เป็นผลอันทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณอาจารย์ ดร.ยรรยง เต็งอำนวย และผู้ช่วยศาสตราจารย์ ดร.ชวลิต ศรีสถาพรพัฒน์ คณะกรรมการสอบวิทยานิพนธ์ที่ให้คำแนะนำในการปรับปรุงงานวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านที่ อบรม สั่งสอน ให้ความรู้ต่าง ๆ มากมาย

ขอกราบขอบพระคุณคุณพ่อ คุณแม่ และสมาชิกในครอบครัวที่ให้ความรัก ความห่วงใย และเป็นกำลังใจแก่ผู้วิจัยในการดำเนินชีวิตมาโดยตลอด

ขอบคุณนางสาวนันทพรรณ เป็นสุข ที่มีส่วนช่วยในการติดต่อประสานงานให้งานวิจัยสำเร็จลุล่วง พร้อมด้วยเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกคนที่เป็นกำลังใจให้กันเสมอมา ที่ให้คำแนะนำความช่วยเหลือต่างๆ ตลอดระยะเวลาที่ดำเนินการวิจัย

สารบัญ

หน้า

หน้า

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากการใช้งานระบบเว็บเซอร์วิซ (Web Services) เป็นที่แพร่หลายและกว้างขวางเป็นอย่างมาก ซึ่งเว็บเซอร์วิซเองมักจะมีการติดต่อสื่อสารกันโดยใช้ภาษาเอกซ์เอ็มแอล (XML) เป็นรูปแบบข้อมูลในการติดต่อ และใช้วิสเดิล (WSDL - Web Services Description Language) ในการอธิบายวิธีการใช้งาน ซึ่งวิสเดิลก็คือเอกสารที่เขียนโดยภาษาเอกซ์เอ็มแอล ที่อธิบายรายละเอียดในการติดต่อกับเว็บเซอร์วิซ เพื่อให้แอพพลิเคชันที่ต้องการเรียกใช้เว็บเซอร์วิซรู้ว่าเว็บเซอร์วิซนั้นให้บริการอะไรบ้างและจะติดต่อได้อย่างไร การใช้งานเว็บเซอร์วิซมีโอกาสเสี่ยงที่จะถูกโจมตีจากผู้ประสงค์ร้ายต่อระบบ เพราะเว็บเซอร์วิซส่วนใหญ่มักจะเป็นเป้าหมายของการโจมตี และการออกแบบวิสเดิลที่ไม่ดีนั้นอาจทำให้เกิดผลกระทบต่อเว็บเซอร์วิซเอง เพราะวิสเดิลมีจุดอ่อนที่ทำให้ผู้ประสงค์ร้ายรู้ช่องโหว่ของเว็บเซอร์วิซ

จากการใช้เอกซ์เอ็มแอลสกีมา (XML Schema) ในการระบุรูปแบบข้อมูลที่แลกเปลี่ยนกับเว็บเซอร์วิซไว้ในวิสเดิล จะพบว่าในสกีมาอาจจะมีการใส่ข้อบังคับ (Restriction) เพื่อบอกถึงขอบเขตของข้อมูลที่แลกเปลี่ยน และข้อบังคับนี้เองจะเป็นสิ่งที่ระบุว่าวิสเดิลนั้นมีความเข้มงวดมากเพียงใดในเรื่องของข้อมูล ซึ่งวิสเดิลที่ไม่เข้มงวดมากหรือถูกออกแบบไว้ไม่ดีจะส่งผลให้การโจมตีเว็บเซอร์วิซทำได้ง่าย เช่น วิสเดิลที่ไม่ได้ระบุขนาดของข้อมูลนำเข้าที่เป็นสายอักขระ (String) ว่ามีกี่ตัวอักษร หรือมีการระบุจำนวนเอลิเมนต์ของข้อมูลแบบไม่จำกัด (Unbounded) อาจจะเป็นการเปิดโอกาสให้ผู้ประสงค์ร้ายส่งข้อมูลที่มีความยาวหรือมีจำนวนมากไปยังเว็บเซอร์วิซ เป็นผลทำให้เว็บเซอร์วิซนั้น ๆ ถูกโจมตีจากคอมมานด์อินเจคชัน (Command Injection) หรือการปฏิเสธการให้บริการ (Denial of Service: DoS) ซึ่งเป็นการโจมตีสองประเภทที่เกี่ยวข้องโดยตรงกับการออกแบบสกีมาข้อมูลที่ไม่เข้มงวด และทำให้เกิดการล้วงข้อมูล ทำอันตรายต่อเว็บเซอร์วิซ หรือทำให้เว็บเซอร์วิซนั้นไม่สามารถให้บริการได้ ก่อให้เกิดความเสียหายในแง่ของชื่อเสียงและมูลค่าทางธุรกิจ

ผู้วิจัยจึงมีแนวคิดในการประเมินความเสี่ยงต่อการถูกโจมตีทางความมั่นคงสำหรับเว็บเซอร์วิซอันเป็นผลมาจากการที่ข้อมูลนำเข้าของเว็บเซอร์วิซมีรูปแบบที่ไม่เข้มงวด อย่างไรก็ตามการออกแบบรูปแบบข้อมูลที่ไม่เข้มงวดในบางกรณีอาจเป็นสิ่งที่หลีกเลี่ยงไม่ได้ เช่น เว็บเซอร์วิซจำเป็นต้องนำเข้าข้อมูลที่มีขนาดหรือจำนวนไม่จำกัดมาเพื่อการทำงาน โดยไม่สามารถระบุจำนวนหรือรูปแบบของข้อมูลที่ตายตัวได้ แต่ในบางกรณีอาจเป็นสิ่งที่หลีกเลี่ยงได้ เช่น เว็บเซอร์วิซซื้อ

ขายสินค้ารับข้อมูลประเภทบัตรเครดิตเป็น string โดยไม่มีการระบุรูปแบบที่จำกัด ทั้ง ๆ ที่สามารถระบุลงไปได้ว่าบัตรเครดิตที่รับนั้นมีประเภทอะไรบ้าง ในลักษณะเช่นนี้จะถือว่าการออกแบบรูปแบบข้อมูลนี้ในวิสเดิลยังหละหลวม และหากสามารถแนะนำผู้ออกแบบเว็บเซอร์วิซได้ว่า ข้อมูลใดที่ออกแบบไว้หละหลวม ผู้ออกแบบเว็บเซอร์วิซสามารถใช้เป็นแนวทางในการปรับปรุงการออกแบบให้เข้มงวดขึ้น ซึ่งจะเป็นประโยชน์ต่อการใช้งานเว็บเซอร์วิซต่อไป

ในการพิจารณาว่าการออกแบบข้อมูลที่หละหลวมนั้นหลีกเลี่ยงได้หรือไม่ ผู้วิจัยจะพิจารณาจากโดเมนออนโทโลจี (Domain Ontology) ของเว็บเซอร์วิซ ภายใต้สมมติฐานว่าโดเมนออนโทโลจีจะระบุองค์ความรู้เกี่ยวกับโดเมนการทำงานของเว็บเซอร์วิซ ซึ่งอยู่ในรูปของคำศัพท์หรือเทอมในโดเมน ความสัมพันธ์ระหว่างเทอม และรูปแบบข้อมูลที่เกี่ยวข้องกับเทอมในโดเมน ในปัจจุบันมีการประยุกต์ใช้การกำกับความหมาย (Semantic Annotation) ด้วยเทอมในโดเมนออนโทโลจีให้กับส่วนต่าง ๆ ภายในวิสเดิล รวมทั้งเอลิเมนต์ข้อมูลในสกีมา เพื่อเป็นการระบุความหมายเพิ่มเติมให้กับข้อมูล การกำกับความหมายนี้จะช่วยให้การทำความเข้าใจเว็บเซอร์วิซว่ามีความหมายในการทำงานอย่างไรสามารถทำได้อย่างอัตโนมัติมากขึ้น และเป็นประโยชน์ในหลายทาง เช่น การค้นหาหรือเปรียบเทียบเว็บเซอร์วิซของผู้ให้บริการรายต่าง ๆ ซึ่งออกแบบไว้แตกต่างกันว่ามีความสามารถในการทำงานแบบเดียวกันหรือแลกเปลี่ยนข้อมูลแบบเดียวกันหรือไม่ ผู้วิจัยจะนำการกำกับความหมายให้กับวิสเดิลโดยอิงโดเมนออนโทโลจีมาประยุกต์ใช้เพื่อประเมินการออกแบบวิสเดิล โดยพิจารณาว่าหากเอลิเมนต์ข้อมูลใดในวิสเดิลถูกออกแบบไว้ไม่เข้มงวดแต่เอลิเมนต์นั้นถูกกำกับความหมายด้วยเทอมในโดเมนออนโทโลจีซึ่งระบุรูปแบบของข้อมูลที่เข้มงวดกว่าไว้ จะแสดงว่าการออกแบบเอลิเมนต์ข้อมูลในวิสเดิลเป็นการออกแบบที่ไม่ดีแต่สามารถบรรเทาความเสี่ยงต่อการโจมตีโดยทำการออกแบบให้เข้มงวดขึ้นได้ตามที่โดเมนออนโทโลจีระบุไว้ ผู้วิจัยจะเสนอมาตรวัดเพื่อประเมินความเสี่ยงต่อการโจมตีสำหรับเว็บเซอร์วิซ โดยเน้นที่การโจมตีแบบคอมมานด์อินเจคชันและการปฏิเสธการให้บริการตามที่กล่าวไว้ข้างต้น โดยพิจารณาจากรูปแบบข้อมูลนำเข้าในวิสเดิลซึ่งไม่เข้มงวดและความรุนแรงของการโจมตีแบบต่าง ๆ นอกจากนี้หากข้อมูลในวิสเดิลมีการกำกับความหมายด้วยเทอมในโดเมนออนโทโลจี ซึ่งผู้วิจัยจะใช้กลไกเอสเอวิสเดิล (Semantic Annotations for WSDL) ในการกำกับความหมาย องค์ความรู้จากโดเมนออนโทโลจีมาช่วยในการทวนสอบได้ว่าความเสี่ยงใดเกิดจากการออกแบบที่ไม่ดีของสกีมาในวิสเดิลแต่สามารถบรรเทาความเสี่ยงต่อการโจมตีได้ ทั้งนี้เพื่อเป็นข้อแนะนำให้แก่ผู้ออกแบบเว็บเซอร์วิซในการพิจารณาปรับปรุงการออกแบบเว็บเซอร์วิซต่อไป

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อเสนอแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตีของเว็บเซอร์วิซโดยพิจารณาจากคำอธิบายเซอร์วิซ และรองรับคำอธิบายเซอร์วิซที่มีการกำกับความหมาย
2. เพื่อพัฒนาเครื่องมือสนับสนุนแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตี

1.3 ขอบเขตของการวิจัย

1. พิจารณาความเสี่ยงต่อการโจมตีสองประเภทคือ คอมมานด์อินเจคชันและการปฏิเสธการให้บริการ
2. พิจารณาเฉพาะเอลิเมนต์ข้อมูลนำเข้าของวิสเดิล โดยเฉพาะข้อมูลประเภทสายอักขระซึ่งเป็นเป้าหมายของการโจมตีแบบคอมมานด์อินเจคชัน และการจำกัดจำนวนข้อมูลซึ่งเป็นเป้าหมายของการโจมตีแบบการปฏิเสธการให้บริการ
3. การกำกับความหมายให้กับวิสเดิลตามโดเมนออนโทโลจีทำโดยผู้ออกแบบเว็บเซอร์วิซและถือว่าโดเมนออนโทโลจีเป็นแม่แบบองค์ความรู้ที่เกี่ยวข้องกับการทำงานของเว็บเซอร์วิซ
4. ในกรณีที่วิสเดิลไม่มีการกำกับความหมายหรือออนโทโลจีที่ใช้กำกับความหมายไม่ได้ระบุข้อบังคับที่เฉพาะเจาะจงไว้ งานวิจัยยังสามารถคำนวณดัชนีความเสี่ยงให้กับเว็บเซอร์วิซได้ แต่จะไม่สามารถระบุได้ว่าความเสี่ยงนั้นสามารถบรรเทาได้หรือไม่

1.4 ประโยชน์ที่ได้รับ

1. ได้แบบจำลองในการประเมินดัชนีความเสี่ยงต่อการถูกโจมตีสำหรับเว็บเซอร์วิซ โดยเป็นการประเมินเบื้องต้นจากวิสเดิล
2. ผลการประเมินสามารถใช้เป็นข้อแนะนำแก่ผู้ออกแบบเว็บเซอร์วิซได้ว่าการออกแบบวิสเดิลยังไม่เข้มงวดที่ส่วนใดบ้างและน่าจะบรรเทาความเสี่ยงต่อการโจมตีโดยการปรับปรุงให้เข้มงวดขึ้นได้หรือไม่
3. ผลการประเมินสามารถใช้เป็นข้อมูลเปรียบเทียบระหว่างเว็บเซอร์วิซเพื่อแนะนำผู้ใช้บริการได้ว่า เว็บเซอร์วิซใดมีความเสี่ยงต่อการถูกโจมตีจากการออกแบบที่ไม่เข้มงวดมากกว่ากัน และเว็บเซอร์วิซใดมีความใส่ใจที่จะบรรเทาการถูกโจมตีมากกว่ากัน

1.5 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึง ความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิดและวิธีการดำเนินงานวิจัย บทที่ 4 กล่าวถึงการออกแบบและพัฒนาระบบ บทที่ 5 กล่าวถึงวิธีการทดสอบระบบ และบทที่ 6 กล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “A Security Attack Risk Assessment for Web Services Based on Data Schemas and Semantics” โดย Wanwisa Phocharoen และ Twittie Senivongse, ในงานประชุมวิชาการ “2012 International Conference on Information Technology and Software Engineering (ITSE2012)” ณ มหาวิทยาลัยปักกิ่งเจียวทง (Beijing Jiaotong University - BJTU) เมืองปักกิ่ง ประเทศจีน ระหว่างวันที่ 8-10 ธันวาคม 2555

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 วิสเดิล

วิสเดิล (Web Services Description Language: WSDL) [1] คือ เอกสารที่เขียนโดยภาษาเอกซ์เอ็มแอล เพื่อใช้อธิบายวิธีการเรียกใช้งานของเว็บเซอร์วิซ และบอกถึงบริการของเว็บเซอร์วิซนั้น ๆ ว่ามีให้บริการงานด้านใดบ้าง

1. การกำหนดชนิดข้อมูลที่เว็บเซอร์วิซใช้ในสกีมา

   เมื่อเว็บเซอร์วิซมีการแลกเปลี่ยนข้อมูลที่จะติดต่อกับแอพพลิเคชันภายนอก จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดชนิดข้อมูลที่ใช้ภายในเว็บเซอร์วิซ โดยจะทำการประกาศชนิดข้อมูลต่าง ๆ ที่ใช้ในการนำเข้าและส่งออกจากเว็บเซอร์วิซ ดังภาพที่ 2.1

```
<types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
            xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="TradePrice">
         <complexType>
           <all>
             <element name="price" type="float"/>
           </all>
         </complexType>
      </element>
    </schema>
</types>
```

ภาพที่ 2.1 ชนิดข้อมูลที่เว็บเซอร์วิซใช้ในสกีมา [2]

2. ข้อบังคับในเอกซ์เอ็มแอลสกีมา

   ดังที่กล่าวไว้ในหัวข้อที่ 1.1 ว่างานวิจัยนี้ให้ความสนใจแก่ข้อมูลนำเข้าที่ได้กำหนดไว้โดยเอกซ์เอ็มแอลสกีมาของข้อมูลในวิสเดิล ซึ่งอาจจะถูกออกแบบไว้ไม่เข้มงวดและมีโอกาสเสี่ยงที่จะถูกโจมตีจากผู้ประสงค์ร้ายต่อเซอร์วิซ ข้อบังคับ (Restriction) ใน

เอกซ์เอ็มแอลสกีมาในวิสเดิลคือการกำหนดรูปแบบที่เฉพาะเจาะจงให้กับข้อมูลเอกซ์เอ็กแอล [2] การกำหนดข้อบังคับจะถูกใช้ในการตรวจสอบ (Validate) ว่าข้อมูลมีรูปแบบตามที่เว็บเซอร์วิซต้องการหรือไม่ เป็นการลดโอกาสของข้อมูลที่อยู่ในรูปแบบที่ไม่เหมาะสมและลดโอกาสการทำอันตรายต่อเว็บเซอร์วิซ ตัวอย่างการกำหนดข้อบังคับให้กับข้อมูลเอกซ์เอ็มแอลในวิสเดิลมีดังต่อไปนี้

1) ข้อบังคับเกี่ยวกับค่าข้อมูล (Restriction on Values) คือการกำหนดช่วงค่าของข้อมูลที่เว็บเซอร์วิซจะนำมาใช้งาน ภาพที่ 2.2 เป็นการประกาศเอลิเมนต์ที่ชื่อว่า “age” โดยมีการกำหนดช่วงค่าข้อมูล integer ของเอลิเมนต์นี้เป็น 0-120

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ภาพที่ 2.2 ข้อบังคับในวิสเดิลแบบกำหนดค่าข้อมูล [2]

2) ข้อบังคับเกี่ยวกับกลุ่มของค่าที่เป็นไปได้ (Restriction on a Set of Values) คือการระบุค่าแจงนับ (Enumeration) ของข้อมูลที่เป็นไปได้ที่เว็บเซอร์วิซจะนำมาใช้งาน ภาพที่ 2.3 เป็นการประกาศเอลิเมนต์ที่ชื่อว่า “car” โดยมีการกำหนดกลุ่มของค่าข้อมูล string ของเอลิเมนต์นี้เป็น Audi, Golf หรือ BMW

```
<xs:element name="car">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
      <xs:enumeration value="BMW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ภาพที่ 2.3 ข้อบังคับในวิสเดิลแบบกำหนดกลุ่มของค่าที่เป็นไปได้ [2]

3) ข้อบังคับเกี่ยวกับชุดของค่าข้อมูล (Restriction on a Series of Values) คือการกำหนดรูปแบบ (Pattern) ของค่าข้อมูลที่เป็นไปได้ที่เว็บเซอร์วิซจะนำมาใช้งาน ภาพที่ 2.4 เป็นการประกาศเอลิเมนต์ที่ชื่อว่า “password” โดยกำหนดรูปแบบว่า

มี 8 ตัวอักษร แต่ละตัวอักษรจะเป็น a-z ที่เป็นตัวใหญ่หรือตัวเล็กก็ได้ หรือเป็นตัวเลข 0-9

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ภาพที่ 2.4 ข้อบังคับในวิสเดิลแบบกำหนดรูปแบบชุดของค่าข้อมูล [2]

4) ข้อบังคับเกี่ยวกับความยาว (Restriction on Length) คือการกำหนดความยาวของข้อมูลที่เว็บเซอร์วิซจะนำมาใช้งาน ภาพที่ 2.5 เป็นการประกาศเอลิเมนต์ที่ชื่อว่า “password” ที่ต้องมีความยาว 8 ตัวอักษร ส่วนในภาพที่ 2.6 เอลิเมนต์ที่ชื่อ “password” จะต้องมีความยาวอย่างน้อย 5 ตัวอักษรแต่ไม่เกิน 8 ตัวอักษร

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ภาพที่ 2.5 ข้อบังคับในวิสเดิลแบบกำหนดความยาวของข้อมูล [2]

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ภาพที่ 2.6 ข้อบังคับในวิสเดิลแบบกำหนดช่วงความยาวของข้อมูล [2]

5) ข้อบังคับเกี่ยวกับจำนวนของข้อมูล (Occurrence Indicators) คือการจำกัดจำนวนเอลิเมนต์ข้อมูล โดยกำหนดจำนวนน้อยที่สุด (minOccurs) และจำนวนมากที่สุด (maxOccurs) ของข้อมูลที่จะเว็บเซอร์วิซจะนำมาใช้งาน ค่าโดยปริยาย (Default) คือ minOccurs = “1” และ maxOccurs = “1” กรณีที่ maxOccurs =

“unbounded” แปลว่าจำนวนเอลิเมนต์มีได้ไม่จำกัด ภาพที่ 2.7 แสดงการประกาศเอลิเมนต์ที่ชื่อว่า “childName” โดยมีการกำหนดจำนวนเอลิเมนต์ที่เป็นไปได้ตั้งแต่ 0-10

```
<xs:element name="childName" maxOccurs="10" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string"/>
 <xs:simpleType>
</xs:element>
```

ภาพที่ 2.7 ข้อบังคับในวิสเดิลแบบกำหนดจำนวนของข้อมูล [2]

2.1.2 ออนโทโลจี

ออนโทโลจี (Ontology) [3] คือข้อกำหนด (Specification) ขององค์ความรู้ภายใต้หัวเรื่องหรือโดเมนที่สนใจ ซึ่งอธิบายอยู่ในรูปของคำศัพท์หรือเทอมที่ปรากฏในโดเมน และความสัมพันธ์ระหว่างเทอม ทำให้เกิดการใช้ข้อมูลองค์ความรู้นี้ร่วมกัน (Share) หรือนำมาใช้ซ้ำ (Reuse) และทำให้เครื่องคอมพิวเตอร์สามารถเข้าใจความหมายของเทอมต่าง ๆ ได้ใกล้เคียงกับที่มนุษย์เข้าใจ ในปัจจุบันภาษาที่นิยมนำมาใช้อธิบายออนโทโลจีของโดเมนต่าง ๆ คือ ภาษาอาวล์ (Web Ontology Language: OWL) [4] ซึ่งเป็นภาษาเอกซ์เอ็มแอลที่อธิบายเทอมและความสัมพันธ์ในรูปของคลาส (Class) คุณสมบัติหรือพร็อพเพอร์ตี (Property) ความสัมพันธ์แบบซับคลาส (Subclass) และซับพร็อพเพอร์ตี (Subproperties) อีกทั้งยังมีการแบ่งชนิดของพร็อพเพอร์ตีออกเป็น อ็อบเจกต์พร็อพเพอร์ตี (Object Property) ซึ่งแสดงความสัมพันธ์ระหว่างคลาสที่เป็นเจ้าของพร็อพเพอร์ตีกับคลาสอื่น ๆ และ ดาตาไทป์พร็อพเพอร์ตี (Datatype Property) โดยจะแสดงความสัมพันธ์ระหว่างคลาสที่เป็นเจ้าของพร็อพเพอร์ตีกับข้อมูลประเภทใดประเภทหนึ่ง งานวิจัยนี้จะทำการอธิบายความสัมพันธ์ระหว่างคลาสและพร็อพเพอร์ตีเหล่านี้พร้อมทั้งการกำหนดรูปแบบข้อบังคับ [5] ภายในออนโทโลจีดังต่อไปนี้

1. ข้อบังคับเกี่ยวกับค่าข้อมูล คือการกำหนดช่วงค่าของข้อมูล จากภาพที่ 2.8 จะเห็นว่ามีคลาส “Person” และคลาส “AdultPerson” ซึ่งคลาส “AdultPerson” เป็นซับคลาสของคลาส “Person” และมีพร็อพเพอร์ตี “hasAge” ที่มีชนิดข้อมูลเป็น “int” โดยมีการระบุข้อบังคับของพร็อพเพอร์ตี “hasAge” ภายในคลาส “Person” ให้มีช่วงข้อมูลตั้งแต่ 18-100

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="AdultPerson">
      <rdfs:subClassOf rdf:resource="#Person"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:DatatypeProperty rdf:ID="hasAge"/>
          </owl:onProperty>
          <owl:allValuesFrom>
            <rdfs:Datatype>
              <xsp:base rdf:resource="&xsd;int"/>
              <xsp:minInclusive
rdf:datatype="&xsd;int">18</xsp:minInclusive>
              <xsp:maxExclusive rdf:datatype="&xsd;int">100</xsp:
maxExclusive >
            </rdfs:Datatype>
          </owl:allValuesFrom>
        </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>
```

ภาพที่ 2.8 ข้อบังคับในออนโทโลจีแบบกำหนดค่าข้อมูล [4]

2. ข้อบังคับเกี่ยวกับกลุ่มของค่าที่เป็นไปได้ คือการระบุค่าแจงนับของข้อมูลที่เป็นไปได้ จากภาพที่ 2.9 จะพบว่าคลาส “CreditCard” มีการกำหนดข้อบังคับที่พร็อพเพอร์ตี “hasCardType” เป็นคลาส “cardType” โดยภายในคลาส “cardType” จะกำหนดกลุ่มค่าที่เป็นไปได้คือ Amex, MasterCard และ Visa

```
<owl:Class rdf:about="&Ontology1344438200889;CreditCard">
<owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="&Ontology1344438200889;hasCardType"/>
      <owl:someValuesFrom
rdf:resource="&Ontology1344438200889;cardType"/>
    </owl:Restriction>
</owl:equivalentClass>
</owl:Class>

<rdfs:Datatype rdf:about="&Ontology1344438200889;cardType">
<owl:equivalentClass>
    <rdfs:Datatype>
      <owl:oneOf>
          <rdf:Description>
            <rdf:type rdf:resource="&rdf;List"/>
            <rdf:first>Amex</rdf:first>
            <rdf:rest>
                <rdf:Description>
                  <rdf:type rdf:resource="&rdf;List"/>
                  <rdf:first>MasterCard</rdf:first>
                  <rdf:rest>
                      <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first>Visa</rdf:first>
                        <rdf:rest rdf:resource="&rdf;nil"/>
                      </rdf:Description>
                  </rdf:rest>
                </rdf:Description>
            </rdf:rest>
          </rdf:Description>
      </owl:oneOf>
    </rdfs:Datatype>
</owl:equivalentClass>
</rdfs:Datatype>
```

ภาพที่ 2.9 ข้อบังคับในออนโทโลจีแบบกำหนดกลุ่มค่าที่เป็นไปได้ [4]

3. ข้อบังคับเกี่ยวกับชุดของค่าข้อมูล คือการกำหนดรูปแบบของค่าข้อมูลที่เป็นไปได้ จากภาพที่ 2.10 จะพบว่าคลาส “CustomerID” มีพร็อพเพอร์ตี “id” ซึ่งถูกกำหนดรูปแบบให้มีความยาว 13 ตำแหน่งโดยจะขึ้นต้นด้วยตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ 3 ตัว และหลังจากนั้นจะตามด้วยตัวเลข 10 ตัว

```
<owl:Class rdf:about="&Ontology1344438200889;CustomerID">
<owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Ontology1344438200889;id"/>
      <owl:someValuesFrom>
          <rdfs:Datatype>
            <owl:onDatatype rdf:resource="&xsd;string"/>
            <owl:withRestrictions rdf:parseType="Collection">
                <rdf:Description>
                  <xsd:pattern>[A-Z]{3}\d{10}</xsd:pattern>
                </rdf:Description>
            </owl:withRestrictions>
          </rdfs:Datatype>
      </owl:someValuesFrom>
    </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

ภาพที่ 2.10 ข้อบังคับในออนโทโลจีแบบกำหนดชุดของค่าข้อมูล [4]

4. ข้อบังคับเกี่ยวกับความยาว คือการกำหนดความยาวของข้อมูล จากภาพที่ 2.11 จะพบว่าคลาส “OrderID” มีพร็อพเพอร์ตี “id” ซึ่งถูกกำหนดให้มีความยาวสูงสุดเป็น 10 ตำแหน่งและมีชนิดข้อมูลเป็น “integer”

```
<owl:Class rdf:about="&Ontology1344438200889;OrderID">
<owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Ontology1344438200889;id"/>
      <owl:someValuesFrom>
          <rdfs:Datatype>
            <owl:onDatatype rdf:resource="&xsd;string"/>
            <owl:withRestrictions rdf:parseType="Collection">
                <rdf:Description>
                  <xsd:maxLength
rdf:datatype="&xsd;integer">10</xsd:maxLength>
                </rdf:Description>
            </owl:withRestrictions>
          </rdfs:Datatype>
      </owl:someValuesFrom>
    </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

ภาพที่ 2.11 ข้อบังคับในออนโทโลจีแบบกำหนดความยาว [4]

5. ข้อบังคับเกี่ยวกับจำนวนของข้อมูล คือการจำกัดจำนวนข้อมูล โดยจะทำการกำหนดจำนวนน้อยที่สุด และจำนวนมากที่สุด จากภาพที่ 2.12 จะพบว่าคลาส “Telephone” มีการกำหนดข้อบังคับเกี่ยวกับจำนวนข้อมูลที่พร็อพเพอร์ตี “hasNumber” โดยทำ

การกำหนดจำนวนน้อยที่สุด (minCardinality) เป็น 1 และ จำนวนที่มากที่สุด (maxCardinality) เป็น 2

```
<owl:Class rdf:about="&Ontology1344438200889;Telephone">
<owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="&Ontology1344438200889;hasNumber"/>
            <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="&Ontology1344438200889;hasNumber"/>
            <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxCardinality>
          </owl:Restriction>
      </owl:unionOf>
    </owl:Class>
</owl:equivalentClass>
</owl:Class>
```

ภาพที่ 2.12 ข้อบังคับในออนโทโลจีแบบกำหนดจำนวนของข้อมูล [4]

2.1.3 การกำกับความหมายภายในวิสเดิล

การกำกับความหมายสำหรับวิสเดิลและเอกซ์เอ็มแอลสกีมาหรือเอสเอวิสเดิล (Semantic Annotations for WSDL: SAWSDL) [6] คือการกำกับคอมโพเนนต์ (Component) บนวิสเดิลรวมทั้งเอกซ์เอ็มแอลสกีมาในวิสเดิลด้วยคำศัพท์หรือเทอมจากซีแมนติกโมเดล (Semantic Models) ที่อยู่ภายนอก ซึ่งซีแมนติกโมเดลจะทำหน้าที่อธิบายองค์ความรู้ของโดเมนที่เกี่ยวข้องกับเว็บเซอร์วิซ การกำกับความหมายในวิสเดิลสามารถทำได้โดยการทำระบุ “sawsdl:modelReference” เพิ่มลงในแอททริบิวต์ของแต่ละเอลิเมนต์ เช่น เอลิเมนต์ OrderID หากต้องการระบุเทอมที่ใช้กำกับความหมายสามารถทำได้โดยระบุเพิ่มลงไปแอททริบิวต์ของเอลิเมนต์ ดังภาพที่ 2.13

```
<xsd:element name="orderID"
sawsdl:modelReference="http://org1.example.com/ontologies
/CustomerOrderOntology#OrderID"/>
<xsd:simpleType >
<xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="paymentType" type="tns:payment" />
<xsd:element name="creditCard" type="tns:creditCard" maxOccurs="1"
minOccurs="0"/>
```

ภาพที่ 2.13 การระบุออนโทโลจีที่ใช้ในการกำกับความหมายภายในวิสเดิล [4]

2.1.4 การโจมตีเว็บเซอร์วิซ

การโจมตี (Attack) เรียกอีกอย่างว่าการบุกรุก (Intrusion) หรือการใช้ประโยชน์ (Exploit) เป็นการกระทำที่มุ่งร้ายต่อระบบ ซึ่งเกิดจากเจตนาในการใช้จุดอ่อน (Vulnerability) ของระบบ งานวิจัยนี้จะทำการพิจารณาข้อมูลการโจมตีเว็บเซอร์วิซจาก CAPEC [7] โดยจะกล่าวถึงเฉพาะการโจมตีที่งานวิจัยจะนำมาใช้ดังต่อไปนี้

1. การโจมตีแบบคอมมานด์อินเจคชัน

   วัตถุประสงค์ของการโจมตีแบบคอมมานด์อินเจคชัน (Command Injection) คือ การโจมตีโดยการใส่คำสั่งที่สามารถเข้าไปควบคุมหรือเรียกดูข้อมูลภายในเอกสาร หรือฐานข้อมูลของระบบ ซึ่งการโจมตีประเภทนี้ ได้แก่

- ซีเควลอินเจคชัน (SQL Injection)
- บลายด์ซีเควลอินเจคชัน (Blind SQL Injection)
- ซีเควลอินเจคชันผ่านการแทรกแซงโซปพารามิเตอร์ (SQL Injection through SOAP Parameter Tampering)
- สคริปต์อินเจคชันอย่างง่าย (Simple Script Injection)
- เอกซ์พาธอินเจคชัน (XPath Injection)
- เอกซ์เควียรีอินเจคชัน (XQuery Injection)

2. การโจมตีแบบการปฏิเสธการให้บริการ

   วัตถุประสงค์ของการโจมตีแบบการปฏิเสธการให้บริการ (Denial of Service) คือ การโจมตีสภาพพร้อมใช้งาน (Availability) โดยจะป้องกันไม่ให้ผู้ให้บริการได้รับหรือตอบกลับข้อความจากผู้ใช้บริการและอาจทำให้ไม่สามารถให้บริการได้ เช่น

ทำให้ระบบหยุดทำงาน (Crash) หรือมีการทำงานผิดพลาด การโจมตีประเภทนี้ได้แก่

- การปฏิเสธการให้บริการแบบเอกซ์เอ็มแอล (Violating Implicit Assumptions Regarding XML Content (หรือเรียกว่า XML Denial of Service หรือ XDoS))
- การทำให้หมดทรัพยากรผ่านฟลัดดิง (Resources Depletion through Flooding)
- การทำให้หมดทรัพยากรผ่านดีทีดีอินเจคชันในข้อความโซป (Resource Depletion through DTD Injection in SOAP Message)
- การทำให้หมดทรัพยากรผ่านการจัดสรรทรัพยากร (Resource Depletion through Allocation)
- เอกซ์เอ็มแอลปิงออฟเดธ (XML Ping of Death)

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การตรวจสอบและป้องกันการโจมตีเว็บเซอร์วิซ

ในบางงานวิจัยได้นำเสนอการป้องกันการโจมตีเว็บเซอร์วิซโดยพิจารณาจากการระบุข้อมูลนำเข้าที่มุ่งร้ายต่อระบบ (Malicious Input) เช่นงานวิจัยของ Brinhosa และคณะ [8] ได้ทำการสร้างแบบจำลองสำหรับการตรวจสอบข้อมูลนำเข้าที่ระบุไว้ในวิสเดิล (Web Service Input Validation Model - WSIVM) โดยงานวิจัยนี้ได้ทำการพัฒนาเครื่องมือตามแบบจำลองที่ได้ทำการออกแบบไว้ โดยการทำงานของเครื่องมือนี้จะต้องทำการกำหนดกฎและคุณสมบัติของข้อมูลนำเข้าในวิสเดิลไว้ก่อนและเมื่อผู้ใช้งานเว็บเซอร์วิซทำการส่งค่าข้อมูลมาในระบบ เครื่องมือนี้จะทำการตรวจสอบข้อมูลเหล่านี้ หากพบกว่ามีการส่งค่าข้อมูลมาถูกต้องตามกฎที่ได้ตั้งไว้ก็จะส่งข้อมูลไปประมวลผลต่อไป แต่ถ้าหากไม่เป็นไปตามกฎเครื่องมือนี้จะส่งข้อความที่บ่งบอกถึงความผิดพลาด (Error Message) กลับไปให้ผู้ใช้งาน ซึ่งการตรวจสอบข้อมูลนำเข้านี้ได้เพิ่มความถูกต้องแม่นยำในการประมวลผลและลดการเกิดการโจมตีเว็บเซอร์วิซ แต่ในทางกลับกันก็เป็นการเพิ่มเวลาในการทำงานของเว็บเซอร์วิซด้วยเช่นกัน

งานวิจัยของ Antunes และคณะ [9] ได้เสนอวิธีการในการป้องกันการเกิดช่องโหว่สำหรับการโจมตีเว็บเซอร์วิซประเภทคอมมานด์อินเจคชัน โดยทำการสร้างเครื่องมือสำหรับการศึกษารูปแบบในการเกิดคอมมานด์อินเจคชันซึ่งมีขั้นตอนต่าง ๆ ดังต่อไปนี้

1) ศึกษาจากการส่งข้อมูลเรียกใช้เว็บเซอร์วิซเป็นจำนวนมากโดยเป็นข้อมูลที่ทำให้เว็บเซอร์วิซทำงานได้อย่างปลอดภัยไม่เกิดการโจมตีใด ๆ
2) หลังจากนั้นจะทำการส่งข้อมูลจำนวนมากไปอีกรอบ ซึ่งในข้อมูลจำนวนนี้จะมีข้อมูลที่มุ่งร้ายต่อระบบรวมถึงคำสั่งในการอินเจคชันรวมอยู่ด้วย
3) ระบบจะทำการตรวจสอบข้อมูลโดยทำการเปรียบเทียบคำสั่งที่ทำให้ระบบทำงานได้อย่างปลอดภัยกับคำสั่งมุ่งร้ายต่อระบบ ถ้าหากว่าคำสั่งใดมีรูปแบบไม่ตรงกันกับคำสั่งที่ทำให้ระบบทำงานได้อย่างปลอดภัย ระบบก็จะบันทึกคำสั่งอินเจคชันและจุดที่ซอร์สโคดทำการประมวลผลว่าซอร์สโคดที่จุดนั้นจะเกิดการโจมตีจากคอมมานด์อินเจคชัน

2.2.2 การสร้างแบบจำลองประเมินความมั่นคงสำหรับเว็บเซอร์วิซ

งานวิจัยหลายงานได้นำเสนอแนวทางเกี่ยวกับการประเมินหรือการวัดความมั่นคงสำหรับเว็บเซอร์วิซ ตัวอย่างเช่น งานวิจัยของ Yu และคณะ [10] ได้เสนอแบบจำลองจุดอ่อนของเว็บเซอร์วิซ (Vulnerability Fault Model) ซึ่งกล่าวถึงส่วนต่าง ๆ ของระบบเว็บเซอร์วิซ ทั้งฝั่งตัวกลางในการประกาศและค้นหาเว็บเซอร์วิซ ฝั่งไคลเอนต์ และฝั่งเซิร์ฟเวอร์ที่โฮสต์เว็บเซอร์วิซ ซึ่งอาจมีจุดอ่อนและกลายเป็นเป้าหมายของการโจมตีได้ งานวิจัยนี้ยังได้จัดหมวดหมู่และอธิบายวิธีการโจมตีแบบต่าง ๆ ว่าทำได้อย่างไร เช่น การโจมตีที่เกี่ยวกับการพิสูจน์ตัวจริง การเข้ารหัส การปลอมแปลงแก้ไขข้อมูล การเปิดเผยข้อมูล เป็นต้น พร้อมทั้งได้แนะนำแนวทางในการทดสอบเว็บเซอร์วิซว่ามีจุดอ่อนหรือถูกโจมตีหรือไม่

งานวิจัยของ Pang และ Peng [11] ได้นำเสนอระบบการประเมินความเสี่ยงด้านความมั่นคงของเว็บเซอร์วิซ ซึ่งมีการพิจารณาปัจจัย 3 ด้าน ได้แก่ 1) ด้านการระบุตัวสินทรัพย์ขององค์กรซึ่งต้องการรักษาความมั่นคง (Asset Identify) และมีการคำนวณคะแนนความเสี่ยงของสินทรัพย์ในด้านการรักษาความลับ บูรณภาพ และสภาพพร้อมใช้งาน 2) ด้านการกราดตรวจจุดอ่อน (Vulnerability Scanning) ใช้โปรแกรมเนสซัส (Nessus) ทำการกราดตรวจจุดอ่อนในเว็บเซอร์วิซและประเมินคะแนนความเสี่ยงจากจุดอ่อนเป็น ต่ำ ปานกลาง หรือสูง และ 3) ด้านการสำรวจภัยคุกคาม (Threat Survey) ประเมินภัยคุกคามโดยผู้ดูแลเว็บเซอร์วิซโดยอาศัยแบบสอบถามเพื่อคำนวณค่าความเสี่ยงจากภัยคุกคาม จากนั้นนำคะแนนที่ได้จากทั้งสามด้านมาพิจารณาร่วมกันเทียบกับตารางการคำนวณความเสี่ยงเพื่อให้ได้เป็นเกรดความเสี่ยงด้านความมั่นคงของเว็บเซอร์วิซนั้น

งานวิจัยของ Jiang และคณะ [12] ได้นำเสนอวิธีการประเมินความมั่นคงของเว็บเซอร์วิซ โดยจะพิจารณาจาก 1) ปัจจัยด้านการโจมตี (Attack Factor) ซึ่งหมายถึงความนิยมและวุฒิภาวะของเทคโนโลยีการโจมตีตามภัยคุกคามแต่ละประเภท 2) ระดับความเสี่ยง (Risk Degree) ซึ่งหมายถึงความรุนแรงของภัยคุกคาม และ 3) อัตราความสำเร็จของการโจมตี (Rate of Successful Attack) ปัจจัยทั้งสามด้านนำมาสู่การคำนวณคะแนนความสามารถด้านความมั่นคงของเว็บเซอร์วิซ

งานวิจัยของ Banklongsi และ Senivongse [13] เสนอการคำนวณคะแนนความมั่นคงของเว็บเซอร์วิซโดยพิจารณาจากจำนวนวิธีการรับมือการโจมตีที่ผู้ให้บริการเตรียมการไว้ให้กับเว็บเซอร์วิซ แต่เนื่องจากการโจมตีแต่ละแบบจะมีลักษณะแตกต่างกัน ได้แก่ ความรุนแรง การใช้ประโยชน์จากเครื่องมือโจมตี และผลกระทบต่อการรักษาความลับ บูรณภาพ และสภาพพร้อมใช้งาน ดังนั้นวิธีการรับมือการโจมตีที่แตกต่างกันจึงถือว่ามีคะแนนความมั่นคงที่แตกต่างกันด้วย เช่น เว็บเซอร์วิซที่ใช้วิธีการที่สามารถรับมือการโจมตีที่มีความรุนแรงมากจะได้คะแนนความมั่นคงมากกว่าเว็บเซอร์วิซที่มีการรับมือการโจมตีที่มีความรุนแรงน้อยกว่า เป็นต้น

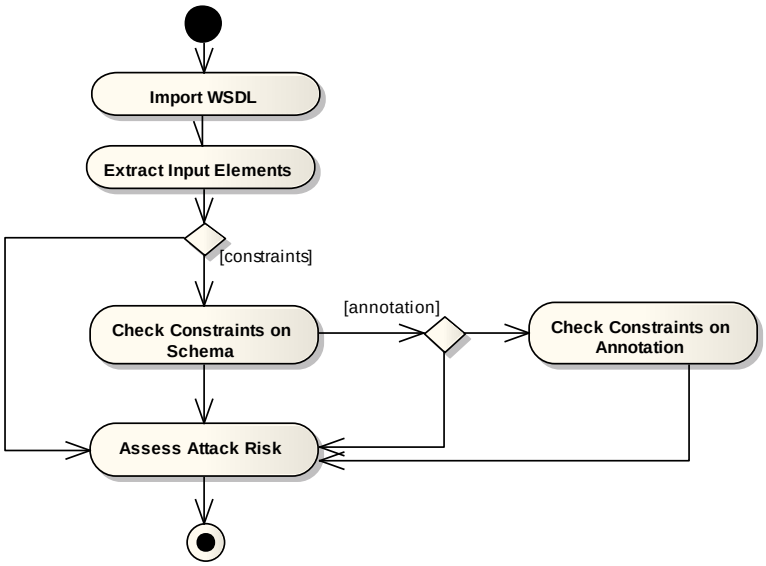
งานวิจัยของ Holgado และคณะ [14] เสนอการประเมินระดับความเสี่ยงของเหตุการณ์ด้านความมั่นคงที่เกิดขึ้นและมีการแจ้งเตือนภายในกลุ่มของระบบจัดการข้อมูลด้านความมั่นคง (Security Information Management Systems) โดยคำนวณความเสี่ยงจากความรุนแรงของผลกระทบจากเหตุการณ์นั้นและความถี่ของการเกิดขึ้น

จากงานวิจัยต่าง ๆ เหล่านี้มีวิธีการประเมินความมั่นคงจากหลากหลายแง่มุม แต่ยังไม่มีงานวิจัยใดที่คำนึงถึงการประเมินความเสี่ยงด้านความมั่นคงจากการออกแบบที่หละหลวมของส่วนต่อประสานของเว็บเซอร์วิซ โดยมีการพิจารณาโดเมนออนโทโลจีของเว็บเซอร์วิซร่วมด้วย ผู้วิจัยขอเสนอแนวทางการประเมินความเสี่ยงในลักษณะดังกล่าวดังรายละเอียดในบทที่ 3

บทที่ 3
การพัฒนาแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตี

งานวิจัยต่าง ๆ ข้างต้นนำเสนอการประเมินคุณภาพด้านความมั่นคงจากหลากหลายแง่มุม โดยพิจารณาปัจจัยภายนอก เช่น การโจมตี ร่วมกับปัจจัยภายใน เช่น ความสามารถด้านความมั่นคงของเว็บเซอร์วิซเอง และทำการประเมินโดยใช้เครื่องมือหรือผู้ดูแลระบบ ผู้วิจัยมีแนวคิดที่คล้ายคลึงกันในแง่การพิจารณาความเสี่ยงต่อการถูกโจมตี แต่มีความแตกต่างในแง่ที่ผู้วิจัยจะพิจารณาที่จุดอ่อนของการออกแบบคำอธิบายเซอร์วิซหรือวิสเดิล ร่วมกับการใช้เทคนิคการกำกับความหมายให้กับวิสเดิลโดยอิงโดเมนออนโทโลจีเพื่อประกอบการพิจารณา การกำกับความหมายให้กับวิสเดิลเป็นเทคนิคที่สามารถนำมาใช้ช่วยประเมินคุณภาพของเว็บเซอร์วิซในด้านต่าง ๆ ได้ เช่น การประเมินคุณภาพด้านการรักษาความเป็นส่วนตัวของข้อมูล [15] การประเมินคุณภาพด้านการออกแบบเว็บเซอร์วิซให้มีขนาดที่เหมาะสมเพื่อการนำไปใช้ซ้ำ [16] เป็นต้น

จากแนวคิดที่จะทำการประเมินความเสี่ยงต่อการโจมตีจากผู้ประสงค์ร้ายต่อเว็บเซอร์วิซ งานวิจัยนี้ให้ความสนใจกับข้อมูลนำเข้าที่ได้กำหนดไว้ในวิสเดิล นอกจากนี้การที่เว็บเซอร์วิซมีการใช้การกำกับความหมายตามออนโทโลจีให้กับวิสเดิลเพื่อวัตถุประสงค์ต่าง ๆ ยังเป็นผลทำให้งานวิจัยนี้สามารถใช้ประโยชน์จากการกำกับความหมายในการตรวจสอบข้อมูลของวิสเดิลว่าได้รับการออกแบบที่หละหลวมแต่สามารถบรรเทาความเสี่ยงโดยออกแบบให้เข้มงวดขึ้นได้หรือไม่ แนวคิดของงานวิจัยนี้สามารถอธิบายภาพรวมได้ดังภาพที่ 3.1



ภาพที่ 3.1 ภาพรวมของงานวิจัย

จากภาพที่ 3.1 ได้อธิบายถึงภาพรวมของการทำงานวิจัยว่า งานวิจัยนี้ให้ความสนใจเอลิเมนต์ที่เป็นข้อมูลนำเข้าในเอกซ์เอ็มแอลสกีมาของวิสเดิล โดยจะทำการพิจารณาเอลิเมนต์ดังกล่าวแต่ละเอลิเมนต์ว่ามีข้อบังคับเรื่องรูปแบบข้อมูลสำหรับการเรียกใช้งานเว็บเซอร์วิซหรือไม่ และถ้าหากเอลิเมนต์นั้น ๆ ไม่มีการระบุข้อบังคับลงไปในสกีมาแต่มีการกำกับความหมายไว้ งานวิจัยก็จะให้ความสนใจที่องค์ความรู้ที่ระบุไว้ในออนโทโลจี โดยพิจารณาว่าข้อมูลนั้นมีการกำกับความหมายแล้วควรมีรูปแบบที่เฉพาะเจาะจงหรือไม่ จากนั้นจะทำการประเมินความเสี่ยงต่อการถูกโจมตีจากความหละหลวมของข้อมูลนำเข้า และประเมินด้วยว่าความเสี่ยงใดน่าจะบรรเทาได้ ซึ่งจะเป็นข้อแนะนำให้ผู้ออกแบบเว็บเซอร์วิซสามารถนำไปปรับปรุงความเข้มงวดของวิสเดิลได้ โดยจะอธิบายรายละเอียดดังหัวข้อย่อยต่อไปนี้

3.1 การวิเคราะห์การกำหนดข้อบังภายในวิสเดิลและการโจมตี

ขั้นตอนแรกตามแนวคิดของงานวิจัยคือ การวิเคราะห์ข้อมูลนำเข้าแต่ละเอลิเมนต์ที่ผู้ออกแบบเว็บเซอร์วิซได้กำหนดไว้ในวิสเดิล และวิเคราะห์การกำหนดข้อมูลออนโทโลจีที่ใช้กำกับความหมายภายในวิสเดิลตามการกำหนดข้อบังคับต่าง ๆ ดังต่อไปนี้

3.1.1 การวิเคราะห์การกำหนดข้อบังคับภายในวิสเดิล

ข้อมูลแต่ละเอลิเมนต์อาจจะมีหรือไม่มีการกำหนดรูปแบบหรือจำนวนข้อมูลที่เฉพาะเจาะจง ซึ่งงานวิจัยนี้ได้ให้ความสนใจเอลิเมนต์ที่เป็นข้อมูลนำเข้าในเอกซ์เอ็มแอลสกีมาของวิสเดิล โดยจะทำการพิจารณาเอลิเมนต์ดังกล่าวว่า แต่ละเอลิเมนต์มีข้อบังคับเรื่องรูปแบบข้อมูลสำหรับการเรียกใช้งานเว็บเซอร์วิซหรือไม่ โดยงานวิจัยได้สนใจการกำหนดข้อบังคับของข้อมูลในวิสเดิลดังตารางที่ 3.1

ตารางที่ 3.1 การกำหนดข้อบังคับภายในวิสเดิล

ข้อบังคับ	การประกาศเอลิเมนต์ในเอกซ์เอ็มแอลสกีมา	คำอธิบาย
None	… <xsd:element name="creditCard" type="xsd:string" /> …	กำหนดเอลิเมนต์ "creditCard" โดยไม่มีการกำหนดข้อบังคับและมีชนิดข้อมูลเป็นสตริง
Enumeration	… <xsd:element name="creditCard"> <xsd:simpleType > <xsd:restriction base="xsd:string"> <xsd:enumeration	กำหนดเอลิเมนต์ "creditCard" และกำหนดกลุ่มของค่าที่เป็นไปได้ที่เอลิเมนต์ "creditCard" ได้แก่ Amex, Mastercard และ Visa

ตารางที่ 3.1 การกำหนดข้อบังคับภายในวิสเดิล (ต่อ)

ข้อบังคับ	การประกาศเอลิเมนต์ในเอกซ์เอ็มแอลสกีมา	คำอธิบาย
	value="Amex"/> <xsd:enumeration value="MasterCard"/> <xsd:enumeration value="Visa"/> …	
Pattern	… <xsd:element name="customerID"> <xsd:simpleType > <xsd:restriction base="xsd:string"> <xsd:pattern value="[A-Z]{3}\d{10}"> …	กำหนดรูปแบบของเอลิเมนต์ "customerID" เป็น "[A-Z]{3}\d{10}" โดยมีความหมายว่า "customerID" ต้องขึ้นต้นด้วยอักษรภาษาอังกฤษตัวใหญ่ 3 ตำแหน่งและตามด้วยอักษรที่เป็นตัวเลข 10 ตำแหน่ง
Length	… <xsd:element name="orderID"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:maxLength value="10"/> …	กำหนดจำนวนตำแหน่งหรือความยาวของ "orderID" ให้มีความยาวสูงสุด 10 ตำแหน่ง
Cardinality	… <xsd:element name="telephone" maxOccurs="2"minOccurs="1"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:maxLength value="14"/> </xsd:restriction> </xsd:simpleType> </xsd:element> …	กำหนดจำนวนข้อมูลที่จะให้ผู้ใช้งานเว็บเซอร์วิซส่งได้ โดย “maxOccurs="2"” มีความหมายว่าส่งได้สูงสุดจำนวน 2 เอลิเมนต์ และ “minOccurs="1"” หมายถึงส่งได้ต่ำสุดจำนวน 1 เอลิเมนต์

3.1.2 ความสัมพันธ์ระหว่างข้อบังคับข้อมูลนำเข้าในวิสเดิลและการโจมตี

จากหัวข้อที่ 3.1.1 จะพบว่าประเภทข้อบังคับข้อมูลนำเข้าในวิสเดิลต่าง ๆ นั้นมีผลกับการถูกโจมตีจากคอมมานด์อินเจคชันและการปฏิเสธการให้บริการ โดยงานวิจัยนี้จะนำเสนอเหตุผลของการเกิดการโจมตีอันเกี่ยวข้องกับการประกาศข้อมูลนำเข้าในเอกซ์เอ็มแอลสกีมาของวิสเดิลตามตารางที่ 3.2

ตารางที่ 3.2 ความสัมพันธ์ระหว่างข้อบังคับข้อมูลนำเข้าในวิสเดิลและการโจมตี [7]

ประเภทข้อบังคับ	เหตุผลในการเกิดคอมมานด์อินเจคชัน	เหตุผลในการเกิดการปฏิเสธการให้บริการ
None	การประกาศข้อมูลนำเข้าในวิสเดิลแบบข้อมูลชนิดอักขระโดยไม่มีการกำหนดข้อบังคับอาจจะทำให้เกิดช่องโหว่สำหรับการโจมตีแบบคอมมานด์อินเจคชัน	การประกาศข้อมูลนำเข้าในวิสเดิลแบบข้อมูลชนิดอักขระ ตัวเลข หรือชนิดข้อมูลอื่น หากไม่มีการกำหนดขนาดความยาว รูปแบบ และจำนวนข้อมูลที่แน่นอน อาจจะทำให้เกิดช่องโหว่สำหรับการโจมตีแบบการปฏิเสธการให้บริการ
Enumeration	การกำหนด “Enumeration” เป็นการกำหนดรายการของค่าหรือข้อมูลที่เอลิเมนต์นั้น ๆ สามารถรับได้ ดังนั้นจะไม่อนุญาตให้มีค่าข้อมูลที่มีเครื่องหมาย ‘ หรือ “ ซึ่งเป็นเหตุทำให้เกิดการอินเจคชันข้อมูล จึงไม่เสี่ยงต่อคอมมานด์อินเจคชัน	ไม่เกี่ยวข้องกับการเกิดการปฏิเสธการให้บริการ
Pattern	การกำหนด “Pattern“ เป็นการระบุรูปแบบตัวอักษรที่เว็บเซอร์วิซจะรับข้อมูลได้ ซึ่งในการกำหนดรูปแบบไม่อนุญาตให้รับหมาย ‘ หรือ “ ซึ่งเป็นเหตุทำให้เกิดการอินเจคชันข้อมูล จึงไม่เสี่ยงต่อคอมมานด์อินเจคชัน	ในการกำหนดรูปแบบข้อมูลในกรณีที่มีการกำหนดจำนวนตัวอักษรหรือความยาวของข้อมูลเช่น "[A-Z]{3}\d{10}" เป็นการกำหนดจำนวนตัวอักษรให้มีขนาดเป็น 13 จึงไม่เป็นเหตุให้เกิดการโจมตีแบบการปฏิเสธการให้บริการ แต่ถ้าหากการกำหนดรูปแบบข้อมูลแบบไม่จำกัดขนาดตัวอักษรเช่น "[A-Z]{3}\d.*" ซึ่งหมายความว่าเอลิเมนต์นี้จะต้องขึ้นต้นด้วยตัวอักษรภาษาอังกฤษตัวใหญ่ 3 ตัวและตามด้วยตัวอักษรที่เป็นตัวเลขจำนวนเท่าได้ก็ได้ จึงอาจทำให้เกิดการโจมตีแบบการปฏิเสธการให้บริการได้
Length	การกำหนดขนาดของเอลิเมนต์ หากมีการกำหนดค่า “maxLength” ที่แน่นอน จะทำให้ไม่เป็นเหตุแก่การเกิดการโจมตีแบบคอมมานด์อินเจคชัน แต่หากกำหนด “minLength” เพียงอย่างเดียวก็อาจจะเป็นเหตุให้เกิดการโจมตีแบบคอมมานด์อินเจคชันได้	การกำหนดขนาดของเอลิเมนต์ หากมีการกำหนดค่า “maxLength” ที่แน่นอน จะทำให้ไม่เป็นเหตุแก่การเกิดการโจมตีแบบการปฏิเสธการให้บริการ แต่หากกำหนด “minLength” เพียงอย่างเดียวก็อาจจะเป็นเหตุให้เกิดการโจมตีแบบการปฏิเสธการให้บริการได้

ตารางที่ 3.2 ความสัมพันธ์ระหว่างข้อบังคับข้อมูลนำเข้าในวิสเดิลและการโจมตี [7] (ต่อ)

ประเภทข้อบังคับ	เหตุผลในการเกิดคอมมานด์อินเจคชัน	เหตุผลในการเกิดการปฏิเสธการให้บริการ
Cardinality	ไม่เกี่ยวข้องกับการเกิดคอมมานด์อินเจคชัน	การกำหนดค่าของจำนวนข้อมูล หากทำการกำหนดจำนวนสูงสุดของข้อมูลที่สามารถส่งมายังเว็บเซอร์วิซได้เป็นจำนวนที่แน่นอนจะไม่ส่งผลให้เกิดการโจมตีแบบการปฏิเสธการ แต่หากมีการกำหนดเป็น “unbounded” ช่องโหว่นี้จะเป็นเหตุให้ผู้ประสงค์ร้ายต่อระบบทำการโจมตีแบบการปฏิเสธการ

แต่ละเอลิเมนต์ข้อมูลนำเข้าในวิสเดิลอาจจะถูกกำกับด้วยเทอมในโดเมนออนโทโลจี ซึ่งเทอมนั้น ๆ อาจจะมีหรือไม่มีข้อบังคับในรูปแบบต่าง ๆ อยู่ด้วย ตารางที่ 3.3 แสดงออนโทโลจีเทอมและรูปแบบข้อบังคับที่สามารถกำกับความหมายให้กับเอลิเมนต์ข้อมูลนำเข้าในวิสเดิล

ตารางที่ 3.3 การกำหนดออนโทโลจีที่ใช้ในการกำกับความหมาย

ข้อบังคับ	การกำหนดข้อมูลออนโทโลจี	คำอธิบาย
None	<owl:Class rdf:about="&Ontology1344438200889;OrderID"> <owl:equivalentClass> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:Class rdf:about="&Ontology1344438200889;CreditCard"> <owl:equivalentClass> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction> </owl:equivalentClass> </owl:Class>	กำหนดคลาส " CreditCard " ซึ่งไม่มีการกำหนดข้อบังคับและมีชนิดข้อมูลเป็นอักขระ
Enumuration	<owl:Class rdf:about="&Ontology1344438200889;CreditCard"> <owl:equivalentClass> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;hasCardType" /> <owl:someValuesFrom rdf:resource="&Ontology1344438200889;cardType"/>	การกำหนดข้อบังคับที่พร็อพเพอร์ตี “hasCardType” ที่คลาส “CreditCard” เป็นคลาส

ตารางที่ 3.3 การกำหนดออนโทโลจีที่ใช้ในการกำกับความหมาย (ต่อ)

ข้อบังคับ	การกำหนดข้อมูลออนโทโลจี	คำอธิบาย
	</owl:Restriction> </owl:equivalentClass> </owl:Class> <rdfs:Datatype rdf:about="&Ontology1344438200889;cardType"> <owl:equivalentClass> <rdfs:Datatype> <owl:oneOf> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>Amex</rdf:first> <rdf:rest> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>MasterCard</rdf:first> <rdf:rest> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>Visa</rdf:first> <rdf:rest rdf:resource="&rdf;nil"/> </rdf:Description> </rdf:rest> </rdf:Description> </rdf:rest> </rdf:Description> </owl:oneOf> </rdfs:Datatype>	“cardType” โดยภายในคลาส “cardType” จะกำหนดกลุ่มค่าที่เป็นไปได้คือ Amex, MasterCard และ Visa
Pattern	<owl:Class rdf:about="&Ontology1344438200889;CustomerID"> <owl:equivalentClass> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> <owl:withRestrictions rdf:parseType="Collection"> <rdf:Description> <xsd:pattern>[A-Z]{3}\d{10}</xsd:pattern> </rdf:Description> </owl:withRestrictions> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction> </owl:equivalentClass> </owl:Class>	กำหนดรูปแบบของคลาส " CustomerID" เป็น "[A-Z]{3}\d{10}" โดยมีความหมายว่า "customerID" ต้องขึ้นต้นด้วยอักษรภาษาอังกฤษตัวใหญ่ 3 ตำแหน่งและตามด้วยอักษรที่เป็นตัวเลข 10 ตำแหน่ง
Length	<owl:Class rdf:about="&Ontology1344438200889;OrderID"> <owl:equivalentClass> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> <owl:withRestrictions rdf:parseType="Collection"> <rdf:Description>	กำหนดจำนวนตำแหน่งหรือความยาวของข้อมูลของคลาส "orderID" ให้มีความยาวสูงสุด 10 ตำแหน่ง

ตารางที่ 3.3 การกำหนดออนโทโลจีที่ใช้ในการกำกับความหมาย (ต่อ)

ข้อบังคับ	การกำหนดข้อมูลออนโทโลจี	คำอธิบาย
	<xsd:maxLength rdf:datatype="&xsd;integer">10</xsd:maxLength> </rdf:Description> </owl:withRestrictions> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction> </owl:equivalentClass> </owl:Class>	
Cardinality	<owl:Class rdf:about="&Ontology1344438200889;Telephone"> <owl:equivalentClass> <owl:Class> <owl:unionOf rdf:parseType="Collection"> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;hasNumber"/> <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:min Cardinality> </owl:Restriction> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;hasNumber"/> <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:max Cardinality> </owl:Restriction> </owl:unionOf> </owl:Class> </owl:equivalentClass> </owl:Class>	กำหนดจำนวนข้อมูลของคลาส “Telephone” โดยให้มีจำนวนสูงสุดเป็น 2 และต่ำสุดเป็น 1

งานวิจัยนี้ให้ความสนใจการกำกับความหมายให้กับสกีมาของข้อมูลนำเข้าด้วย โดยหากเอลิเมนต์มีการกำกับความหมายด้วยออนโทโลจีเทอมซึ่งมีข้อบังคับที่เข้มงวดกว่าในสกีมาของข้อมูลนำเข้า จะถือว่าเอลิเมนต์มีการออกแบบที่ไม่เข้มงวดแต่สามารถบรรเทาความเสี่ยงต่อการโจมตีได้ หากเอลิเมนต์ได้รับการออกแบบใหม่ให้มีสกีมาที่เข้มงวดขึ้นตามข้อบังคับของออนโทโลจีเทอมที่กำกับอยู่ ไม่เช่นนั้นจะถือว่าการออกแบบเอลิเมนต์ที่ไม่เข้มงวดนั้นหลีกเหลี่ยงไม่ได้ ตัวอย่างของการออกแบบเอลิเมนต์ที่ไม่เข้มงวดทั้งที่หลีกเลี่ยงไม่ได้และที่สามารถออกแบบใหม่เพื่อบรรเทาความเสี่ยงได้จะกล่าวถึงในหัวข้อที่ 3.2 การออกแบบที่ไม่เข้มงวดนี้จะเป็นจุดเสี่ยงสำหรับใช้ในการประเมินด้วยแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตีต่อไป

3.2 การวิเคราะห์การกำหนดข้อบังภายในวิสเดิลและการโจมตี

ตามแนวคิดของงานวิจัยคือ วิเคราะห์ข้อมูลนำเข้าแต่ละเอลิเมนต์ที่ผู้ออกแบบเว็บเซอร์วิซได้กำหนดไว้ในวิสเดิล โดยข้อมูลแต่ละเอลิเมนต์อาจจะมีหรือไม่มีการกำหนดรูปแบบหรือจำนวน

ข้อมูลที่เฉพาะเจาะจงและอาจจะมีหรือไม่มีการกำกับความหมายไว้ ในกรณีที่มีการกำกับความหมายด้วยเทอมในออนโทโลจี เทอมนั้นอาจจะมีหรือไม่มีการกำหนดรูปแบบหรือจำนวนไว้อย่างเจาะจงก็ได้ ตัวอย่างการแยกแยะกรณีที่เป็นไปได้ของการกำหนดข้อบังคับให้กับเอลิเมนต์ข้อมูลนำเข้าและการกำกับความหมายเป็นไปดังตารางที่ 3.4

ตารางที่ 3.4 การวิเคราะห์รูปแบบข้อมูลกับการกำกับความหมาย

No.	Schema Rest.	Annotate Rest.	Schema Declaration	Ontology Declaration	Strong Declaration		Weak Declaration			
							Unavoidable Risk		Risk that can be Mitigated	
					Command Injection	DoS	Command Injection	DoS	Command Injection	DoS
1	✘	✘	… **<xs:restriction base="xs:string"/>** …				✓	✓		
2	✓	✘	… <xs:restriction base="xs:string"> <xs:enumeration value="Visa"/> <xs:enumeration value="MasterCard" /> <xs:enumeration value="AMEX" /> </xs:restriction> …		✓	✓				
3	✘	✓	… sawsdl:modelReference="http://purchaseorder/ontology/purchaseorder#CreditCardType"> <xs:restriction base="xs:string"/> …	… <owl:Restriction> <owl:onProperty rdf:resource="#hasType"/> <owl:someValuesFrom> <owl:Class> <owl:one of rdf:parseType="Collection"> <owl:Thing rdf:about="#Visa"/> <owl:Thing rdf:about="#MasterCard"/>			✓	✓	✓	✓

ตารางที่ 3.4 การวิเคราะห์รูปแบบข้อมูลกับการกำกับความหมาย (ต่อ)

No.	Schema Rest.	Annotate Rest.	Schema Declaration	Ontology Declaration	Strong Declaration		Weak Declaration			
							Unavoidable Risk		Risk that can be Mitigated	
					Command Injection	DoS	Command Injection	DoS	Command Injection	DoS
				<owl:Thing rdf:about="#AMEX"/> </owl:oneOf> </owl:Class> </owl:someValuesFrom> </owl:Restriction> …						
4	✓	✓	… sawsdl:modelReference="http://purchaseorder/ontology/purchaseorder#CreditCardType"> <xs:restriction base="xs:string"> <xs:enumeration value="Visa"/> <xs:enumeration value="MasterCard" /> <xs:enumeration value="AMEX" /> </xs:restriction> …	… <owl:Restriction> <owl:onProperty rdf:resource="#hasType"/> <owl:someValuesFrom> <owl:Class> <owl:oneOf rdf:parseType="Collection"> <owl:Thing rdf:about="#Visa"/> <owl:Thing rdf:about="#MasterCard"/> <owl:Thing rdf:about="#AMEX"/> </owl:oneOf> </owl:Class> </owl:someValuesFrom> </owl:Restriction> …	✓	✓				

จากตัวอย่างในตารางที่ 3.4 แบ่งได้เป็นกรณีดังต่อไปนี้

กรณีที่ 1 สกีมาไม่มีการกำหนดข้อบังคับและไม่มีการกำกับความหมาย กรณีตัวอย่างในตารางที่ 3.4 จะถือว่าการออกแบบสกีมาไม่เข้มงวดและไม่มีข้อมูลเชิงความหมายจากออนโทโลจีที่ทำให้ทราบได้ว่าสามารถออกแบบใหม่เพื่อบรรเทาความเสี่ยงได้หรือไม่ จึงอาจเป็นกรณีที่เว็บเซอร์วิซจำเป็นต้องรับข้อมูลนำเข้าที่มีรูปแบบหลากหลาย แต่จะทำให้ไม่สามารถหลีกเลี่ยงการโจมตีที่อาจจะเกิดขึ้นได้

กรณีที่ 2 สกีมามีการกำหนดข้อบังคับแต่ไม่มีการกำกับความหมาย กรณีตัวอย่างในตารางที่ 3.4 จะถือว่าการออกแบบสกีมาเข้มงวดอยู่แล้วเพราะมีการกำหนดค่าข้อมูลที่รับได้ไว้ชัดเจน แม้จะไม่มีข้อมูลเชิงความหมายจากออนโทโลจีกำกับไว้ก็ตาม จึงไม่มีความเสี่ยง

กรณีที่ 3 สกีมาไม่มีการกำหนดข้อบังคับ แต่มีการกำกับความหมายโดยที่ออนโทโลจีมีการกำหนดข้อบังคับไว้ กรณีตัวอย่างในตารางที่ 3.4 จะถือว่าการออกแบบสกีมาไม่เข้มงวด แต่สามารถบรรเทาความเสี่ยงต่อการโจมตีได้หากมีการออกแบบใหม่ให้มีการจำกัดค่าข้อมูลนำเข้าตามข้อมูลเชิงความหมายจากออนโทโลจีซึ่งกำกับอยู่

กรณีที่ 4 สกีมามีการกำหนดข้อบังคับ และมีการกำกับความหมายโดยที่ออนโทโลจีมีการกำหนดข้อบังคับไว้ กรณีตัวอย่างในตารางที่ 3.4 มีการออกแบบสกีมาที่เข้มงวดเท่ากับข้อมูลเชิงความหมายจากออนโทโลจีซึ่งกำกับอยู่ โดยเฉพาะในตัวอย่างนี้เป็นการกำหนดค่าข้อมูลที่รับได้ไว้ชัดเจน จึงถือว่าสกีมามีความเข้มงวดแล้ว ไม่มีความเสี่ยง

ค่าโดยปริยาย (Default) ของเอกซ์เอ็มแอลสกีมาเป็นประโยชน์ต่อการออกแบบวิสเดิลให้เข้มงวดได้เช่นกัน จากตัวอย่างสกีมาในกรณีที่ 1-4 ในตารางที่ 3.4 ข้างต้น ไม่ได้ระบุข้อบังคับเกี่ยวกับจำนวนข้อมูลนำเข้าไว้ จึงเปรียบได้กับการระบุโดยใช้ค่าโดยปริยายคือ minOccurs = “1” และ maxOccurs = “1” ซึ่งทำให้มีความเข้มงวดในแง่ที่สามารถลดความเสี่ยงต่อการโจมตีแบบการปฏิเสธการให้บริการ

3.3 การกำหนดค่าความรุนแรงของการโจมตี

ก่อนที่จะทำการประเมินความเสี่ยงที่พบจากการออกแบบวิสเดิล งานวิจัยจะทำการอธิบายการให้น้ำหนักแก่ประเภทการโจมตีที่จะนำไปใช้ในการประเมิน เนื่องจากที่กล่าวมาในข้างต้นว่าการที่สกีมาในวิสเดิลไม่ได้ระบุข้อบังคับลงไป อาจจะทำให้เสี่ยงต่อการโจมตีได้ งานวิจัยนี้จะพิจารณาการโจมตีประเภทคอมมานด์อินเจคชัน 7 แบบ และประเภทการปฏิเสธการให้บริการ 6 แบบ ตามที่ CAPEC [7] ได้จัดหมวดหมู่ไว้ ซึ่งทั้งหมดเกี่ยวข้องกับข้อมูลนำเข้าของเว็บเซอร์วิซ ผู้วิจัยได้กำหนดคะแนนความรุนแรงของการโจมตีแต่ละประเภทเพื่อใช้ในการคำนวณความเสี่ยง ดังในตารางที่ 3.5 ค่าคะแนนนี้อ้างอิงจากระดับความรุนแรง 5 ระดับที่ CAPEC กำหนดให้กับการโจมตีแต่ละประเภทดังตารางที่ 3.6

ตารางที่ 3.5 การจัดหมวดหมู่การโจมตีและคะแนนความรุนแรง [7]

Type of Attacks	Name of Attacks	Severity Score				
		1	2	3	4	5
Command Injection	SQL Injection				✓	
	Blind SQL Injection				✓	
	SQL Injection through SOAP Parameter Tampering					✓
	Simple Script Injection					✓
	XPath Injection				✓	
	XQuery Injection					✓
DoS	Violating Implicit Assumptions Regarding XML Content (aka XML Denial of Service (XDoS))					✓
	XML Ping of Death			✓		
	Resource Depletion through Allocation			✓		
	Resource Depletion through DTD Injection in SOAP Message			✓		
	Resource Depletion through Flooding			✓		

ตารางที่ 3.6 ค่าความรุนแรงของการโจมตี [7]

Severity Score	Typical Severity Level by CAPEC
5	Very High
4	High
3	Medium
2	Low
1	Very Low

3.4 แบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตี

จากหัวข้อที่ 3.2 เมื่อทำการวิเคราะห์ข้อมูลนำเข้าแล้วจะพบว่าภายในวิสเดิลนั้นอาจจะมีจุดบกพร่องจากการออกแบบที่ไม่เข้มงวด ซึ่งอาจจะเป็นจุดอ่อนต่อการโจมตีได้ ยิ่งมีการออกแบบที่ไม่เข้มงวดเป็นจำนวนมาก ก็ยิ่งมีความเสี่ยงมาก งานวิจัยนี้ขอนำเสนอการคำนวณค่าดัชนีความเสี่ยง (Risk Index) ต่อการโจมตี โดยอ้างอิงจากทฤษฎีการจัดการความเสี่ยง [7] ซึ่งคำนวณดัชนีความเสี่ยงจากผลคูณระหว่างผลกระทบของเหตุการณ์ความเสี่ยง (Impact of Risk Event) กับความน่าจะเป็นในการเกิดเหตุการณ์ความเสี่ยง (Probability of Occurrence of Risk Event) ดังนี้

3.4.1 ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชัน

ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชันสามารถคำนวณได้จากสมการ (1)

$$RI_{CI} = RI_{CI\_U} + RI_{CI\_M} \quad (1)$$

โดยที่

RI_{CI} = ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชัน

RI_{CI_U} = ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชัน ซึ่งหลีกเลี่ยงไม่ได้ และคำนวณได้จากสมการ (2)

RI_{CI_M} = ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชัน ซึ่งบรรเทาได้ และคำนวณได้จากสมการ (3)

และกำหนดให้

$$RI_{Cl\_U} = \sum_{i=1}^{N_{Cl}} S_{Cli} * \frac{NR_{Cl\_U}}{IN} \quad (2)$$

โดยที่

N_{Cl} = จำนวนการโจมตีประเภทคอมมานด์อินเจคชัน ในที่นี้คือ 7 แบบ ตามตารางที่ 3.5

S_{Cli} = คะแนนความรุนแรงของการโจมตีประเภทคอมมานด์อินเจคชันแบบที่ *i* ตามตารางที่ 3.6

NR_{Cl_U} = จำนวนเอลิเมนต์ข้อมูลนำเข้าที่ไม่เข้มงวดต่อการโจมตีประเภทคอมมานด์อินเจคชัน ซึ่งหลีกเลี่ยงไม่ได้

IN = จำนวนเอลิเมนต์ข้อมูลนำเข้าทั้งหมด

และ

$$RI_{Cl\_M} = \sum_{i=1}^{N_{Cl}} S_{Cli} * \frac{NR_{Cl\_M}}{IN} \quad (3)$$

โดยที่

N_{Cl} = จำนวนการโจมตีประเภทคอมมานด์อินเจคชัน ในที่นี้คือ 7 แบบ ตามตารางที่ 3.5

S_{Cli} = คะแนนความรุนแรงของการโจมตีประเภทคอมมานด์อินเจคชันแบบที่ *i* ตามตารางที่ 3.6

NR_{Cl_M} = จำนวนเอลิเมนต์ข้อมูลนำเข้าที่ไม่เข้มงวดต่อการโจมตีประเภทคอมมานด์อินเจคชัน ซึ่งบรรเทาได้

IN = จำนวนเอลิเมนต์ข้อมูลนำเข้าทั้งหมด

3.4.2 ดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการ

ดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการสามารถคำนวณได้จากสมการ (4)

$$RI_{DOS} = RI_{DOS\_U} + RI_{DOS\_M} \quad (4)$$

โดยที่

RI_{DOS} = ดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการ

RI_{DOS_U} = ดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการ ซึ่งหลีกเลี่ยงไม่ได้ และคำนวณได้จากสมการ (5)

RI_{DOS_M} = ดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการ ซึ่งบรรเทาได้ และคำนวณได้จากสมการ (6)

และกำหนดให้

$$RI_{DOS\_U} = \sum_{i=1}^{N_{DOS}} S_{DOSi} * \frac{NR_{DOS\_U}}{IN} \quad (5)$$

โดยที่

N_{DOS} = จำนวนการโจมตีประเภทการปฏิเสธการให้บริการ ในที่นี้คือ 6 แบบ ตามตารางที่ 3.5

S_{DOSi} = คะแนนความรุนแรงของการโจมตีประเภทการปฏิเสธการให้บริการแบบที่ *i* ตามตารางที่ 3.6

NR_{DOS_U} = จำนวนเอลิเมนต์ข้อมูลนำเข้าที่ไม่เข้มงวดต่อการโจมตีประเภทการปฏิเสธการให้บริการ ซึ่งหลีกเลี่ยงไม่ได้

IN = จำนวนเอลิเมนต์ข้อมูลนำเข้าทั้งหมด

และ

$$RI_{DOS\_M} = \sum_{i=1}^{N_{DOS}} S_{DOSi} * \frac{NR_{DOS\_M}}{IN} \quad (6)$$

โดยที่

N_{DOS} = จำนวนการโจมตีประเภทการปฏิเสธการให้บริการ ในที่นี้คือ 6 แบบ ตามตารางที่ 3.5

S_{DOSi} = คะแนนความรุนแรงของการโจมตีประเภทการปฏิเสธการให้บริการแบบที่ *i* ตามตารางที่ 3.6

NR_{DOS_M} = จำนวนเอลิเมนต์ข้อมูลนำเข้าที่ไม่เข้มงวดต่อการโจมตีประเภทการปฏิเสธการให้บริการ ซึ่งบรรเทาได้

IN = จำนวนเอลิเมนต์ข้อมูลนำเข้าทั้งหมด

3.4.3 ดัชนีความเสี่ยงต่อการโจมตีโดยรวม

ดัชนีความเสี่ยงต่อการโจมตีโดยรวมสามารถคำนวณได้จากผลรวมของสมการ (1) และ (4) ตามสมการ (7)

$$RI = RI_{CI} + RI_{DOS} \tag{7}$$

เมื่อกำหนดแนวคิดและกระบวนการในการดำเนินงานวิจัยแล้ว งานวิจัยนี้จึงทำการพัฒนาเครื่องมือสนับสนุนแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตี ดังรายละเอียดในบทที่ 4

บทที่ 4
การพัฒนาเครื่องมือสนับสนุนแบบจำลอง

เพื่อทำให้งานวิจัยมีความสมบูรณ์มากขึ้น งานวิจัยนี้จึงได้จัดทำเครื่องมือเพื่อสนับสนุนการทำงานของแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตี โดยทำการพัฒนาด้วยภาษาจาวา (Java Programming) ร่วมกับการใช้งานไลบรารี (Library) ที่มีการพัฒนาไว้ใช้งานอยู่แล้วมาประยุกต์เพื่อให้การทำงานของเครื่องมือทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยขั้นตอนในการพัฒนาเครื่องมือมีดังต่อไปนี้

4.1 ความต้องการด้านหน้าที่

ก่อนที่จะทำการพัฒนาเครื่องมือเพื่อสนับสนุนงานวิจัย จะต้องมีการกำหนดความต้องการด้านหน้าที่ (Functional Requirement) เพื่อใช้ในการวางแผน กำหนดโครงสร้าง และคัดเลือกภาษาที่ใช้ในการพัฒนา โดยความต้องการด้านหน้าที่สามารถแจกแจงได้ดังตารางที่ 4.1

ตารางที่ 4.1 ความต้องการของระบบด้านหน้าที่

รหัส	ชื่อ	คำอธิบาย
F01	การจัดการเรียกไฟล์วิสเดิล	การจัดการเรียกไฟล์วิสเดิลมีหน้าที่ดังนี้ 1. เรียกไฟล์วิสเดิลจากไดเรกทอรี (Directory) ที่กำหนด 2. นำไฟล์วิสเดิลเข้าสู่ขั้นตอนการค้นหาข้อมูลนำเข้า
F02	การค้นหาข้อมูลนำเข้าในวิสเดิล	การค้นหาข้อมูลนำเข้าในวิสเดิลมีหน้าที่ดังนี้ 1. ค้นหาเอลิเมนต์ที่เป็นข้อมูลนำเข้าทั้งหมดภายในวิสเดิล 2. ระบุรายการต่าง ๆ ของเอลิเมนต์ที่ถูกค้นหาได้แก่ ● ชื่อ ● ชนิดข้อมูล ● ข้อบังคับที่ถูกกำหนดไว้ (ถ้ามี) ● การกำกับความหมาย (ถ้ามี)

ตารางที่ 4.1 ความต้องการของระบบด้านหน้าที่ (ต่อ)

รหัส	ชื่อ	คำอธิบาย
F03	การจัดการเรียกไฟล์ออนโทโลจี	การจัดการเรียกไฟล์ออนโทโลจีมีหน้าที่ดังนี้ 1. เรียกไฟล์ออนโทโลจีจากไดเรกทอรีที่กำหนด 2. นำไฟล์ออนโทโลจีเข้าสู่ขั้นตอนการค้นหาข้อมูลที่นำมาใช้กำกับความหมายให้กับวิสเดิล
F04	การวิเคราะห์ข้อมูลนำเข้าในวิสเดิลและข้อมูลที่ใช้กำกับความหมาย	การวิเคราะห์ข้อมูลนำเข้าในวิสเดิลและข้อมูลที่ใช้กำกับความหมายมีหน้าที่ดังนี้ 1. ค้นหาข้อมูลนำเข้าที่ไม่เกี่ยวข้องกับความเสี่ยงต่อการโจมตีเว็บเซอร์วิซ 2. ค้นหาข้อมูลนำเข้าที่ไม่สามารถหลีกเลี่ยงการโจมตีเว็บเซอร์วิซได้ 3. ค้นหาข้อมูลนำเข้าที่สามารถบรรเทาความเสี่ยงต่อการโจมตีเว็บเซอร์วิซได้
F05	การคำนวณดัชนีความเสี่ยงต่อการโจมตี ตามแบบจำลองที่กำหนด	การคำนวณดัชนีความเสี่ยงต่อการโจมตีมีหน้าที่ดังนี้ 1. กำหนดแบบจำลองดัชนีความเสี่ยงต่อการโจมตี ตามหัวข้อ 3.4 2. นำจำนวนข้อมูลนำเข้าทั้งหมดมาประมวลผลตามแบบจำลองดัชนีความเสี่ยงต่อการโจมตี

4.2 ความต้องการด้านไม่ใช่หน้าที่

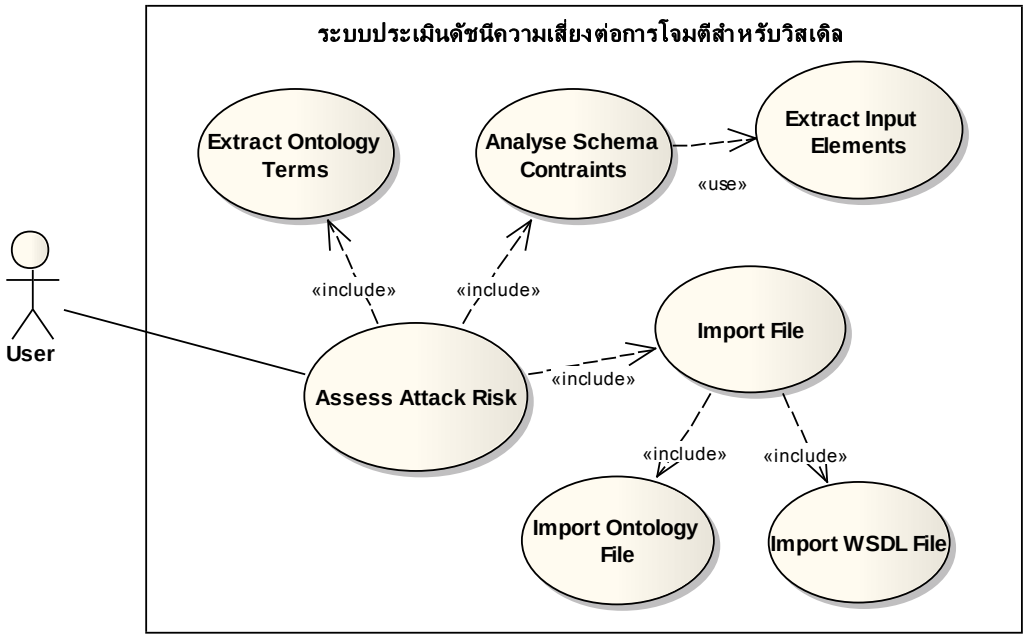
เมื่อได้ความต้องการด้านหน้าที่แล้ว ความต้องการด้านไม่ใช่หน้าที่ (Non-functional Requirement) ยังเป็นส่วนสำคัญทำให้เครื่องมือมีความสมบูรณ์มากยิ่งขึ้น ดังนั้นงานวิจัยนี้จึงได้กำหนดความต้องการที่ไม่ใช่หน้าที่เพิ่มเติม โดยสามารถแจกแจงได้ดังตารางที่ 4.2

ตารางที่ 4.2 ความต้องการของระบบด้านไม่ใช่หน้าที่

รหัส	ชื่อ	คำอธิบาย
NF01	เครื่องมือเว็บแอพพลิเคชัน	การทำงานของเครื่องมือที่เป็นแบบเว็บแอพพลิเคชันมีหน้าที่ดังนี้ 1. เครื่องมือสามารถทำงานบนเบราว์เซอร์ คือ อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer - IE) ไฟร์ฟอกซ์ (Firefox) กูเกิลโครม (Google Chrome)

4.3 การออกแบบระบบ

เมื่อได้ความต้องการด้านหน้าที่และความต้องการที่ไม่ใช่หน้าที่แล้ว งานวิจัยจึงขอนำเสนอแผนภาพยูสเคส เพื่อนำมาใช้อธิบายหน้าที่การทำงานทั้งหมดของระบบดังภาพที่ 4.1



ภาพที่ 4.1 ระบบการประเมินดัชนีความเสี่ยงต่อการโจมตี

4.4 การพัฒนาระบบ

เมื่อการออกแบบระบบเสร็จสิ้นแล้ว ขั้นตอนต่อไปคือการพัฒนาระบบ โดยมีรายละเอียดดังต่อไปนี้

4.4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบมีดังต่อไปนี้

1. ฮาร์ดแวร์ (เครื่องไมโครคอมพิวเตอร์)
   - หน่วยประมวลผล อินเทลเพนเทียมเอ็ม 2.40 กิกะเฮิร์ตซ์ (Core i5 M 2.40 GHz.)
   - หน่วยความจำ 2 กิกะไบต์
   - ฮาร์ดดิสก์ 240 กิกะไบต์
2. ซอฟต์แวร์
   - ระบบปฏิบัติการ วินโดวส์เซเวน อัลติเมท (Windows 7 Ultimate)
   - เครื่องมือพัฒนาโปรแกรมภาษาวาจา ประกอบด้วยรายการต่าง ๆ ดังนี้
     1. โปรแกรมอิคลิปส์เวอร์ชันเฮลิออส (Eclipse Java EE IDE for Web Developers, Version: Helios Release)
     2. จาวาเอพีไอไลบรารี (Java API Library) ประกอบด้วย
        - จาวาซิสเต็มไลบรารี เวอร์ชัน 1.6 (Java JRE System Library Version 1.6)
        - อีซีวิสเดิลไลบรารี (Easy WSDL Library) สำหรับการค้นหาข้อมูลนำเข้าในวิสเดิลและการกำกับความหมายภายในวิสเดิล
        - อาวล์เอพีไอไลบรารี (OWL API Library) สำหรับการค้นหาข้อมูลของออนโทโลจีที่ใช้ในการกำกับความหมาย
3. การติดตั้งซอฟต์แวร์ในการพัฒนาระบบ

   เมื่อได้กำหนดเครื่องมือสำหรับการพัฒนาระบบเรียบร้อยแล้ว ขั้นต่อไปคือการติดตั้งเครื่องมือทั้งหมดลงในเครื่องคอมพิวเตอร์ที่ใช้พัฒนาระบบ โดยมีลำดับการติดตั้งเครื่องมือเป็นไปตามขั้นตอนต่อไปนี้

   - ติดตั้งระบบปฏิบัติการ วินโดวส์เซเวน อัลติเมท
   - ติดตั้งชุดพัฒนาโปรแกรมภาษาวาจา เวอร์ชัน 1.6

- ติดตั้งเอพีไอไลบรารีที่สนับสนุนการค้นหาข้อมูลนำเข้าภายในวิสเดิลและองค์ความรู้ในออนโทโลจี

4.4.2 การพัฒนาส่วนต่อประสาน

เมื่อทำการติดตั้งซอฟต์แวร์เรียบร้อยแล้ว จึงทำการพัฒนาส่วนต่อประสานผู้ใช้เพื่อให้สอดคล้องกับการออกแบบระบบ โดยจะต้องมีการทำงานตรงกับตามความต้องการด้านหน้าที่และความต้องการที่ไม่ใช่หน้าที่ตามตารางที่ 4.1 และ ตารางที่ 4.2 โดยการใช้งานส่วนต่อประสานกับผู้ใช้แสดงไว้ในภาคผนวก ค

เมื่อทำการพัฒนาเครื่องมือสนับสนุนงานวิจัยเรียบร้อยแล้ว ขั้นตอนต่อไปคือการเตรียมข้อมูลเพื่อทำการทดสอบเครื่องมือ และทำการทดสอบ ดังรายละเอียดในบทที่ 5

บทที่ 5
การทดสอบ

เมื่อทำการพัฒนาเครื่องมือเรียบร้อยแล้วจะต้องทำการทดสอบการทำงานของเครื่องมือว่าเครื่องมือนั้นสามารถทำงานได้อย่างถูกต้องตามแบบจำลองที่กำหนดไว้และครบถ้วนตามความต้องการด้านหน้าที่หรือไม่

5.1 การทดสอบความถูกต้องของฟังก์ชันการทำงานของเครื่องมือ

การทดสอบฟังก์ชันการทำงานของเครื่องมือนั้นเป็นการตรวจสอบความต้องการด้านหน้าที่ตามที่ได้ออกแบบไว้ตามตารางที่ 4.1 โดยการทดสอบจะต้องทำการจัดเตรียมข้อมูลที่ใช้ในสกีมาและออนโทโลจีภาษาอาวล์ดังต่อไปนี้

5.1.1 การวิเคราะห์ข้อมูลนำเข้าภายในสกีมา

ในขั้นตอนแรกของงานวิจัย เมื่อได้วิสเดิลที่จะนำมาใช้ในการทดลองแล้ว จะต้องทำการคัดเลือกเฉพาะข้อมูลนำเข้าภายในวิสเดิลเพื่อจะทำการวิเคราะห์ว่าแต่ละเอลิเมนต์ว่ามีชนิดข้อมูลและข้อบังคับเป็นแบบใดบ้าง โดยวิสเดิลที่ใช้ทดสอบ เป็นวิสเดิลที่จัดทำขึ้นตามรูปแบบของเอสเอวิสเดิลจากเว็บไซต์ดับเบิลยูสามซี (W3C) [6] ซึ่งมีรูปแบบตามภาคผนวก ก โดยงานวิจัยสามารถคัดแยกข้อมูลนำเข้าได้ตามตารางที่ 5.1 ดังต่อไปนี้

ตารางที่ 5.1 การกำหนดชนิดข้อมูลสกีมาเพื่อใช้ในการทดสอบเครื่องมือ

No.	Element Name	Schema Declaration	Restriction Description	Semantic Annotation
1	customerName	type="xsd:string"	มีชนิดข้อมูลเป็นแบบสายอักขระ	#CustomerName
2	customerAddress	type="xsd:string"	มีชนิดข้อมูลเป็นแบบสายอักขระ	
3	customerTelephone	minOccurs="1" maxOccurs="unbounded" <xsd:restriction base="xsd:string"> <xsd:minLength value="14"/> </xsd:restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระและมีความยาวอย่างน้อย 14 ตำแหน่ง โดยต้องส่งจำนวนข้อมูลเข้ามาในระบบอย่างน้อย 1 จำนวนและไม่จำกัดจำนวนสูงสุดในการส่งเข้ามาในระบบ โดยจะสามารถส่ง เข้ามาเป็นจำนวนเท่าใดก็ได้	#Telephone

ตารางที่ 5.1 การกำหนดชนิดข้อมูลสกีมาเพื่อใช้ในการทดสอบเครื่องมือ (ต่อ)

No.	Element Name	Schema Declaration	Restriction Description	Semantic Annotation
4	customerID	<xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:pattern value="[A-Z]{3}\d{10}"/> </xsd:restriction> </xsd:simpleType>	มีชนิดข้อมูลเป็นแบบสายอักขระ และมีความยาวจำนวน 13 ตำแหน่งโดยมีรูปแบบดังนี้คือ จะต้องขึ้นต้นด้วยอักษรภาษาอังกฤษตัวใหญ่ 3 ตำแหน่งและตามด้วยอักษรที่เป็นตัวเลข 10 ตำแหน่ง	#CustomerID
5	itemCode	maxOccurs="unbounded" type="xsd:string"	มีชนิดข้อมูลเป็นแบบสายอักขระ และกำหนดจำนวนข้อมูลสูงสุดที่สามารถส่งเข้ามาในระบบเป็นแบบไม่จำกัดจำนวน โดยจะสามารถส่งเข้ามาเป็นจำนวนเท่าใดก็ได้	
6	itemName	maxOccurs="unbounded" type="xsd:string"	มีชนิดข้อมูลเป็นแบบสายอักขระ และกำหนดจำนวนข้อมูลสูงสุดที่สามารถส่งเข้ามาในระบบเป็นแบบไม่จำกัดจำนวน โดยจะสามารถส่งเข้ามาเป็นจำนวนเท่าใดก็ได้	
7	quantity	maxOccurs="unbounded" type="xsd:integer"	มีชนิดข้อมูลเป็นจำนวนเต็มและกำหนดจำนวนข้อมูลสูงสุดที่สามารถส่งเข้ามาในระบบเป็นแบบไม่จำกัดจำนวน โดยจะสามารถส่งเข้ามาเป็นจำนวนเท่าใดก็ได้	
8	orderID	<xsd:restriction base="xsd:string"> <xsd:maxLength value="10" /> </xsd:restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระ และมีความยาวที่สามารถรับสูงสุด 10 ตำแหน่ง	#OrderID

ตารางที่ 5.1 การกำหนดชนิดข้อมูลสกีมาเพื่อใช้ในการทดสอบเครื่องมือ (ต่อ)

No.	Element Name	Schema Declaration	Restriction Description	Semantic Annotation
9	payment	<xsd:restriction base="xs:string"> <xsd:enumeration value="Cash" /> <xsd:enumeration value="Card" /> </xsd:restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระ และกำหนดกลุ่มค่าที่เป็นได้ 2 ค่าคือ “Cash“ และ “Card”	
10	creditCard	<xsd:restriction base="xs:string"> </xsd:restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระ	#CreditCard

5.1.2 การวิเคราะห์ข้อมูลออนโทโลจีที่ใช้กำกับความหมายภายในวิสเดิล

เมื่อได้ชนิดและข้อบังคับของข้อมูลนำเข้าภายในวิสเดิลแล้ว จะพบว่าบางเอลิเมนต์ในวิสเดิลตามภาคผนวก ก มีการกำกับความหมายไว้โดยใช้โมเดลเรฟเฟอร์เรนซ์ซึ่งมีไว้สำหรับการกำกับความหมายดังที่ได้กล่าวไว้ในหัวข้อ 2.1.3 การกำกับความหมายเหล่านี้จะอ้างอิงถึงคลาสต่าง ๆ ภายในออนโทโลจีตามภาคผนวก ข โดยแต่ละคลาสในออนโทโลจีสามารถวิเคราะห์ได้ดังตารางที่ 5.2 ดังต่อไปนี้

ตารางที่ 5.2 การกำหนดชนิดข้อมูลออนโทโลจีที่เพื่อใช้ในการทดสอบเครื่องมือ

No.	Anotation Name	Annotation Declaration	Restriction Description
1	CustomerName	<owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> <owl:withRestrictions rdf:parseType="Collection"> <rdf:Description> <xsd:maxLength rdf:datatype="&xsd;integer">50</xsd:maxLength> </rdf:Description> </owl:withRestrictions> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระ และมีความยาวสูงสุดที่รับได้ 50 ตำแหน่ง

ตารางที่ 5.2 การกำหนดชนิดข้อมูลออนโทโลจีที่เพื่อใช้ในการทดสอบเครื่องมือ (ต่อ)

No.	Anotation Name	Annotation Declaration	Restriction Description
2	Telephone	<owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;hasNumber"/> <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1 </owl:minCardinality> </owl:Restriction> <owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;hasNumber"/> <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">2 </owl:maxCardinality> </owl:Restriction>	กำหนดจำนวนข้อมูลที่สามารถส่งเข้ามาในระบบอย่างน้อย 1 ตัว และจำกัดจำนวนสูงสุดที่สามารถส่งได้เป็น 2 ตัว
3	CustomerID	<owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> <owl:withRestrictions rdf:parseType="Collection"> <rdf:Description> <xsd:pattern>[A- Z]{3}\d{10}</xsd:pattern> </rdf:Description> </owl:withRestrictions> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระและมีความยาวจำนวน 13 ตำแหน่งโดยจะต้องขึ้นต้นด้วยอักษรภาษาอังกฤษตัวใหญ่ 3 ตำแหน่งและตามด้วยอักษรที่เป็นตัวเลข 10 ตำแหน่ง
4	OrderID	<owl:Restriction> <owl:onProperty rdf:resource="&Ontology1344438200889;id"/> <owl:someValuesFrom> <rdfs:Datatype> <owl:onDatatype rdf:resource="&xsd;string"/> <owl:withRestrictions rdf:parseType="Collection"> <rdf:Description> <xsd:maxLength rdf:datatype="&xsd;integer">10</xsd:maxLength> </rdf:Description> </owl:withRestrictions> </rdfs:Datatype> </owl:someValuesFrom> </owl:Restriction>	มีชนิดข้อมูลเป็นแบบสายอักขระและมีความยาวสูงสุดที่รับได้ 10 ตำแหน่ง

ตารางที่ 5.2 การกำหนดชนิดข้อมูลออนโทโลจีที่เพื่อใช้ในการทดสอบเครื่องมือ (ต่อ)

No.	Anotation Name	Annotation Declaration	Restriction Description
5	CreditCard	<owl:oneOf> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>Amex</rdf:first> <rdf:rest> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>MasterCard</rdf:first> <rdf:rest> <rdf:Description> <rdf:type rdf:resource="&rdf;List"/> <rdf:first>Visa</rdf:first> <rdf:rest rdf:resource="&rdf;nil"/> </rdf:Description> </rdf:rest> </rdf:Description> </rdf:rest> </rdf:Description> </owl:oneOf>	กำหนดกลุ่มค่าที่เป็นไปได้ 3 ค่าคือ “Amex” , “MasterCard” และ “Visa”

5.1.3 ตัวอย่างการวิเคราะห์ข้อมูลนำเข้าและการกำกับความหมายภายในวิสเดิล

เมื่อทำการวิเคราะห์ข้อมูลนำเข้าและข้อมูลที่ใช้กำกับความหมายแล้ว หลังจากนั้นจะต้องทำการพิจารณาแต่ละเอลิเมนต์ในตารางที่ 5.1 ว่ามีความเสี่ยงแบบใดบ้างและสมควรถูกกำหนดว่าเป็นเอลิเมนต์ที่มีความเสี่ยงแบบใดสามารถบรรเทาได้หรือไม่ตามที่ได้อธิบายไว้ในหัวข้อที่ 3.2 ตารางที่ 5.3 จะแสดงตัวอย่างการวิเคราะห์ข้อมูลนำเข้าบางเอลิเมนต์ในตารางที่ 5.1 ร่วมกับข้อมูลการพิจารณาออนโทโลจีที่ใช้กำกับความหมายในตารางที่ 5.2

ตารางที่ 5.3 ตัวอย่างการวิเคราะห์ข้อมูลนำเข้าและการกำกับความหมาย

Element Name	Schema Restriction	Annotation Restriction	DOS Attack Description	Injection Attack Description
customerAddress	ไม่มีการกำหนดข้อบังคับไว้ในสกีมา	ไม่มีการกำกับความหมายไว้ในสกีมา	• ไม่สามารถหลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้เพราะไม่กำหนดจำนวนตำแหน่งที่แน่นอนให้กับเอลิเมนต์ • ไม่สามารถทำให้บรรเทาลงได้	• ไม่สามารถหลีกเลี่ยงการโจมตีในรูปแบบของคอมมานด์อินเจคชันได้เพราะไม่กำหนดรูปแบบที่แน่นอนให้กับเอลิเมนต์ • ไม่สามารถทำให้บรรเทาลงได้

ตารางที่ 5.3 ตัวอย่างการวิเคราะห์ข้อมูลนำเข้าและการกำกับความหมาย (ต่อ)

Element Name	Schema Restriction	Annotation Restriction	DOS Attack Description	Injection Attack Description
customerTelephone	มีการกำหนดข้อบังคับไว้ในสกีมา	มีการกำกับความหมายไว้ในสกีมา	• ไม่สามารถหลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้เพราะมีการกำหนดข้อบังคับเป็น “minLength” และกำหนด “maxOccur” เป็น “unbouded” ซึ่งไม่จำกัดจำนวนข้อมูล • สามารถทำให้บรรเทาลงได้หากออกแบบใหม่ให้มีการกำหนดจำนวนข้อมูลสูงสุดเป็น 2 ตัว	• ไม่สามารถหลีกเลี่ยงต่อการโจมตีในรูปแบบของคอมมานด์อินเจคชันได้เพราะไม่กำหนดรูปแบบที่แน่นอนให้กับเอลิเมนต์ • ไม่สามารถทำให้บรรเทาลงได้
customerID	มีการกำหนดข้อบังคับไว้ในสกีมา	มีการกำกับความหมายไว้ในสกีมา	• สามารถหลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้เพราะมีการกำหนดความยาวไว้ 13 ตำแหน่ง • ไม่จำเป็นต้องทำการพิจารณาว่าสามารถทำให้บรรเทาลงได้หรือไม่เพราะเอลิเมนต์นี้หลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้	• สามารถหลีกเลี่ยงการโจมตีในรูปแบบของคอมมานด์อินเจคชันได้เพราะมีการกำหนดรูปแบบและอักขระที่แน่นอน • ไม่จำเป็นต้องทำการพิจารณาว่าสามารถทำให้บรรเทาลงได้หรือไม่เพราะเอลิเมนต์นี้หลีกเลี่ยงการโจมตีได้ในรูปแบบของคอมมานด์อินเจคชันได้
payment	มีการกำหนดข้อบังคับไว้ในสกีมา	ไม่มีการกำกับความหมายไว้ในสกีมา	• สามารถหลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้เพราะมีการกำหนดจำนวนสูงสุดโดยปริยายไว้เป็น 1 และมีการกำหนดกลุ่มค่าที่เป็นไปได้	• สามารถหลีกเลี่ยงการโจมตีในรูปแบบของคอมมานด์อินเจคชันได้เพราะมีการกำหนดกลุ่มค่าที่เป็นไปได้
creditCard	ไม่มีการกำหนดข้อบังคับไว้ในสกีมา	มีการกำกับความหมายไว้ในสกีมา	• ไม่สามารถหลีกเลี่ยงการโจมตีในรูปแบบของการปฏิเสธการให้บริการได้เพราะไม่กำหนดจำนวนตำแหน่งที่แน่นอนให้กับเอลิเมนต์ • สามารถบรรเทาได้หากออกแบบใหม่ให้มีการกำหนดกลุ่มค่าที่เป็นไปได้	• ไม่สามารถหลีกเลี่ยงการโจมตีในรูปแบบของคอมมานด์อินเจคชันได้เพราะไม่มีการกำหนดรูปแบบที่แน่นอน • สามารถบรรเทาได้หากออกแบบใหม่ให้มีการกำหนดกลุ่มค่าที่เป็นไปได้

5.1.4 ผลการคำนวณแบบจำลองการประเมินดัชนีความเสี่ยงต่อการถูกโจมตี

จากตัวอย่างการวิเคราะห์ข้อมูลนำเข้าและข้อมูลที่ใช้กำกับความหมายตามตารางที่ 5.3 จะสามารถสรุปได้ดังตารางที่ 5.4 ดังต่อไปนี้

ตารางที่ 5.4 ผลการวิเคราะห์การออกแบบข้อบังคับและการกำกับความหมายในสกีมา

No.	Element Name	Schema Restriction	Annotation Restriction	Strong Declaration		Weak Declaration			
						Unavoidable Risk		Risk that can be Mitigated	
				Injection	DOS	Injection	DOS	Injection	DOS
1	customerName	✗	✓			✓			✓
2	customerAddress	✗	✗			✓	✓		
3	customerTelephone	✓	✓			✓	✓		✓
4	customerID	✓	✓	✓	✓				
5	itemCode	✓	✗			✓	✓		
6	itemName	✓	✗			✓	✓		
7	quantity	✓	✗	✓			✓		
8	orderID	✓	✓		✓	✓			
9	payment	✓	✗	✓	✓				
10	creditCard	✗	✓					✓	✓

จากตารางที่ 5.4 ทำให้ทราบว่าจำนวนเอลิเมนต์ที่เป็นข้อมูลนำเข้าทั้งหมดในวิสเดิลตามภาคผนวก ก นั้นมีทั้งหมด 10 เอลิเมนต์ โดยสามารถสรุปค่าคงที่ต่าง ๆ ที่ใช้ในการคำนวณสำหรับดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชันตามสมการ (1) ได้ดังต่อไปนี้

โดยที่

$Total\ of\ S_{CIi} = 27$

$NR_{CI\_U} = 6$

$NR_{CI\_M} = 1$

$IN = 10$

ดังนั้น

$RI_{CI\_U} = 27 * (6/10) = 16.2$

RI_{CI_M} = 27 * (1/10) = 2.7

และ

RI_{CI} = 18.9

เมื่อได้ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชันแล้วยังสามารถคำนวณดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการตามสมการ (4) ได้ดังต่อไปนี้

โดยที่

$Total\ of\ S_{DOSi}$ = 17

NR_{DOS_U} = 5

NR_{DOS_M} = 3

IN = 10

ดังนั้น

RI_{DOS_U} = 17 * (5/10) = 8.5

RI_{DOS_M} = 17 * (3/10) = 5.1

และ

RI_{DOS} = 13.6

เมื่อได้ดัชนีความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชันและดัชนีความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการ ดังนั้นจะสามารถคำนวณดัชนีความเสี่ยงต่อการโจมตีโดยรวมตามสมการ (7) ได้ดังนี้

RI = 32.5

จากตารางที่ 5.4 และค่าดัชนีความเสี่ยงข้างต้นสามารถสรุปผลได้ดังตารางที่ 5.5

ตารางที่ 5.5 สรุปผลการทดลอง

ค่าดัชนีความเสี่ยง	จำนวนเอลิเมนต์	ผลการคำนวณ	คำอธิบาย
RI_{CI_U}	6	16.2	วิสเดิลที่ใช้ในการทดสอบนี้มี 6 เอลิเมนต์ที่มีการออกแบบที่ไม่เข้มงวด และอาจถูกโจมตีจากคอมมานด์อินเจคชันได้ แต่ไม่มีข้อมูลเชิงความหมาย หรือมีข้อมูลเชิงความหมายแต่ข้อมูลไม่สามารถบอกได้ว่าจะปรับปรุงสกีมาให้เข้มงวดขึ้นได้อย่างไร ดังนั้นจึงขึ้นกับผู้ออกแบบเว็บเซอร์วิซในการพิจารณาปรับปรุงการออกแบบเอง

ตารางที่ 5.5 สรุปผลการทดลอง (ต่อ)

ค่าดัชนีความเสี่ยง	จำนวนเอลิเมนต์	ผลการคำนวณ	คำอธิบาย
RI_{Cl_M}	1	2.7	วิสเดิลที่ใช้ในการทดสอบนี้มี 1 เอลิเมนต์ที่สามารถทำการออกแบบใหม่ให้เข้มงวดขึ้นได้ ซึ่งจะช่วยบรรเทาความเสี่ยงต่อการโจมตีประเภทคอมมานด์อินเจคชันได้โดยการออกแบบใหม่ตามการกำกับความหมาย
RI_{DoS_U}	5	8.5	วิสเดิลที่ใช้ในการทดสอบนี้มี 5 เอลิเมนต์ที่มีการออกแบบที่ไม่เข้มงวด และอาจถูกโจมตีจากการปฏิเสธการให้บริการได้ แต่ไม่มีข้อมูลเชิงความหมาย หรือมีข้อมูลเชิงความหมายแต่ข้อมูลไม่สามารถบอกได้ว่าจะปรับปรุงสกีมาให้เข้มงวดขึ้นได้อย่างไร ดังนั้นจึงขึ้นกับผู้ออกแบบเว็บเซอร์วิซในการพิจารณาปรับปรุงการออกแบบเอง
RI_{Dos_M}	3	5.1	วิสเดิลที่ใช้ในการทดสอบนี้มี 3 เอลิเมนต์ที่สามารถทำการออกแบบใหม่ให้เข้มงวดขึ้นได้ ซึ่งจะช่วยบรรเทาความเสี่ยงต่อการโจมตีประเภทการปฏิเสธการให้บริการได้ โดยการออกแบบใหม่ตามการกำกับความหมาย

5.2 สรุปผลการทดสอบ

จากการทดสอบฟังก์ชันการทำงานข้างต้นสามารถสรุปได้ว่าเครื่องมือสนับสนุนแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตีนั้นสามารถทำงานได้ถูกต้องตามความต้องการด้านหน้าที่และไม่ใช่หน้าที่ของระบบตามที่ได้ออกแบบไว้ และดัชนีต่าง ๆ ที่ได้จากการทดลองนี้ สามารถนำมาใช้ประโยชน์ในการออกแบบวิสเดิลเพื่อทำการตรวจสอบข้อมูลนำเข้าที่อาจเป็นจุดที่ทำให้เว็บเซอร์วิซถูกโจมตีจากการออกแบบที่ไม่เข้มงวดของผู้ออกแบบเว็บเซอร์วิซ ผลการทดลองนี้สามารถช่วยบ่งบอกว่า ผู้ให้บริการเว็บเซอร์วิซแต่ละรายได้คำนึงถึงความมั่นคงเมื่อออกแบบส่วนต่อประสานของเว็บเซอร์วิซ มากน้อยต่างกันอย่างไร จึงเป็นประโยชน์ต่อผู้เรียกใช้บริการในการเลือกเว็บเซอร์วิซมาใช้ในแอพพลิเคชันของตนได้ อีกทั้งผู้ออกแบบเว็บเซอร์วิซควรพิจารณาจำนวนเอลิเมนต์ข้อมูลเข้าที่เสี่ยงต่อการโจมตีแต่ละประเภท ทั้งที่เครื่องมือสามารถแนะนำการปรับปรุงได้และที่ไม่สามารถแนะนำได้ เนื่องจากหากไม่มีการออกแบบสกีมาใหม่ให้มีข้อบังคับรูปแบบใดรูปแบบหนึ่งเพิ่มขึ้นแล้ว เว็บเซอร์วิซจะมีความเสี่ยงต่อการโจมตี

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ผลลัพธ์ที่ได้จากงานวิจัยนี้มีดังนี้

1. ได้แบบจำลองในการประเมินดัชนีความเสี่ยงต่อการโจมตีสำหรับเว็บเซอร์วิซ โดยเป็นการประเมินเบื้องต้นจากข้อมูลนำเข้าที่ระบุอยู่ในสกีมาของวิสเดิล ร่วมกับข้อมูลเชิงความหมายที่กำกับไว้ที่ข้อมูลนำเข้า การประเมินความเสี่ยงทำโดยพิจารณาจุดที่มีการออกแบบข้อมูลนำเข้าอย่างไม่เข้มงวด และความรุนแรงของการโจมตีแบบคอมมานด์อินเจคชันและการปฏิเสธการให้บริการที่อาจจะกระทำกับจุดที่ออกแบบไว้ไม่เข้มงวดนั้น
2. ผลการประเมินที่ได้จากข้อ 1 สามารถนำไปใช้เป็นข้อแนะนำแก่ผู้ออกแบบเว็บเซอร์วิซได้ว่า การออกแบบข้อมูลนำเข้าใดบ้างที่ยังไม่เข้มงวดและเสี่ยงต่อการโจมตี นอกจากนี้ยังสามารถใช้เป็นข้อมูลเปรียบเทียบระหว่างเว็บเซอร์วิซเพื่อแนะนำผู้ใช้บริการได้ว่า เว็บเซอร์วิซใดมีความเสี่ยงต่อการโจมตีจากการออกแบบที่ไม่เข้มงวดมากกว่ากัน และเว็บเซอร์วิซใดใส่ใจที่จะหลีกเลี่ยงการโจมตีมากกว่ากัน
3. งานวิจัยนี้ได้พัฒนาเครื่องมือสนับสนุนการประเมินในรูปแบบของเว็บแอพพลิเคชันที่สามารถทำงานได้ตามความต้องการของแบบการจำลองในการประเมินดัชนีความเสี่ยงต่อการโจมตี โดยสามารถแสดงผลการวิเคราะห์และการคำนวณตามแบบจำลองได้จริง

6.2 ข้อจำกัด

ในการดำเนินงานวิจัยนี้มีข้อจำกัดในเรื่องของการใช้งานสกีมาของข้อมูลนำเข้าและการกำกับความหมาย ดังต่อไปนี้

1. เครื่องมือสนับสนุนแบบจำลองในการประเมินดัชนีความเสี่ยงต่อการโจมตี จะรองรับวิสเดิลเวอร์ชัน 2.0 เป็นต้นไป แต่วิสเดิลที่ใช้งานแพร่หลายทั่วไปจะเป็นวิสเดิลเวอร์ชัน 1.1 ดังนั้นหากต้องการประเมินความเสี่ยงของวิสเดิล 1.1 จะต้องแปลงให้เป็นเวอร์ชัน 2.0 ก่อน
2. ในการกำกับความหมายภายในวิสเดิลนั้นจะต้องเขียนในรูปแบบของโมเดลเรฟเฟอร์เรนซ์ (Model Reference) ตามหลักการกำกับความหมายของภาษาอาวล์เวอร์ชัน 2

เป็นต้นไป เพื่อให้สอดคล้องกับเครื่องมือที่ได้พัฒนาขึ้น แต่อาวล์ออนโทโลจีที่มีอยู่เป็นจำนวนมากเป็นอาวล์เวอร์ชัน 1 ซึ่งยังไม่สนับสนุนการกำหนดข้อบังคับให้กับออนโทโลจีเทอม จึงไม่สามารถใช้กับงานวิจัยได้

3. การกำกับความหมายภายในวิสเดิลจะต้องกำกับที่เอลิเมนต์ที่มีชนิดข้อมูลเป็นซิมเพิลไทป์ (Simple Type) ไม่สามารถกำกับที่คอมเพล็กซ์ไทป์ (Complex Type) ได้ เนื่องจากเครื่องมือสนับสนุนการกำกับความหมายที่ชนิดข้อมูลที่อยู่ระดับล่างสุดเท่านั้น โดยยังไม่สนับสนุนการกำกับความหมายที่ชนิดข้อมูลที่มีความซับซ้อน

6.3 แนวทางการวิจัยต่อไป

จากการนำเสนองานวิจัยนี้จะพบว่า มีการสรุปชนิดของการโจมตีหลักไว้สองชนิดคือ คอมมานด์อินเจคชันและการปฏิเสธการให้บริการ ซึ่งในความเป็นจริงแล้ว สกีมาข้อมูลในวิสเดิลอาจจะเกี่ยวข้องกับการโจมตีประเภทอื่น ๆ อีก จึงสามารถที่จะขยายแบบจำลองให้รองรับการโจมตีประเภทอื่นได้ อีกทั้งแบบจำลองที่นำเสนอควรได้รับการประเมินจากผู้ให้บริการเว็บเซอร์วิซและผู้ใช้บริการจริงว่าสามารถเป็นประโยชน์ต่อการปรับปรุงการออกแบบวิสเดิลและการเลือกใช้เว็บเซอร์วิซได้หรือไม่ อีกทั้งเมื่อเว็บเซอร์วิซได้รับการปรับปรุงให้เข้มงวดขึ้นแล้ว สามารถป้องกันการโจมตีได้เพิ่มขึ้นหรือไม่

รายการอ้างอิง

[1] W3C. Web Services Description Language (WSDL) 1.1 [Online]. Available from : http://www.w3.org/TR/wsdl [2012,March 25].

[2] W3School. XSD Restrictions/Facets [Online]. Available from : http://www.-w3schools.com/schema/schema_facets.asp [2012,March 25].

[3] W3C. OWL Web Ontology Language Guide [Online]. 2004. Available from : http://www.w3.org/TR/owl-guide/ [2012,March 25].

[4] W3C. OWL Web Ontology Language Overview [Online]. 2004. Available from : http://www.w3.org/TR/2004/REC-owl-features-20040210/ [2012,March 25].

[5] Protégé. User-defined datatypes in protégé-owl [Online]. Available from : http://protege.stanford.edu/plugins/owl/xsp.html [2012,March 25].

[6] W3C. Semantic Annotations for WSDL and XML Schema [Online]. 2007. Available from : http://www.w3.org/TR/2007/REC-sawsdl-20070828/ [2012,March 25].

[7] The Mitre Corporation. CAPEC - Common Attack Pattern Enumeration and Classification [Online]. Available from : http://capec.mitre.org/ [2012,-March 25].

[8] Brinhosa, RB., Westphall, CM. and Westphall, CB. Proposal and development of the Web services input validation model. Proceedings of 2012 IEEE Network Operations and Management Symposium (NOMS 2012), pp. 643-646. 2012.

[9] Antunes, N., Laranjeiro, N., Vieira, M. and Madeira, H. Effective detection of SQL/XPath injection vulnerabilities in Web services. Proceedings of 2009 IEEE International Conference on Services Computing (SCC 2009), pp. 260-267. 2009.

[10] Yu, WD., Aravind, D. and Supthaweesuk, P. Software Vulnerability Analysis for Web Services Software Systems. Proceedings of 11th IEEE Symposium on Computers and Communications (ISCC'06), pp. 740-748. 2006.

[11] Jianjing, P. and Xinguang, P. Trustworthy Web Service Security Risk Assessment Research. Proceedings of International Forum on Information Technology and Applications, pp. 417-420. 2009.

[12] Li J., Hao C., Fei, D. and Qiusheng, Z. A Security Evaluation Method Based on Threat Classification for Web Service. Journal of Software, Vol. 6, No. 4, pp. 595-603. April 2011.

[13] Banklongsi, T. and Senivongse, T. A Security Measurement Model for Web Services Based on Provision of Attack Countermeasures. Proceedings of 15th International Annual Symposium on Computational Science and Engineering (ANSCSE15), pp. 593-598. 2011.

[14] Holgado, P., López de Vergara, JE., Villagrá, VA., Sanz, I. and Amaya, A. Sharing information about security alerts using semantic web technologies. Proceedings of the 2010 International Conference on Network and Service Management (CNSM 2010), pp. 270-273. 2010.

[15] Chaiwongsa, P. and Senivongse, T. Web Services Privacy Measurement Based on Privacy Policy and Sensitivity Level of Personal Information. Proceedings of 8th International Conference on Computing and Information Technology (IC2IT 2012), pp. 145-150. 2012.

[16] Muchalintamolee, N. and Senivongse, T. Measuring Granularity of Web Services with Semantic Annotation. Proceedings of 8th International Conference on Computing and Information Technology (IC2IT 2012), pp. 151-156. 2012.

[17] Hughes, B. and Cotterel, M. Software Project Management Fourth Edition, 2006, pp. 155.

[18] McBride, B. Jena: Implementing the RDF Model and Syntax Specification [Online],White Paper. Available from : http://ceur-ws.org/Vol-40/ mcbride.pdf. [2012,March 25].

ภาคผนวก

ภาคผนวก ก
ตัวอย่างวิสเดิลที่กำกับความหมายตามเอสเอวิสเดิล

```
<?xml version="1.0"?>
<description targetNamespace="http://www.example.org/CustomerOrder/"
      xmlns:tns="http://www.example.org/CustomerOrder/"
      xmlns="http://www.w3.org/ns/wsdl"
      xmlns:sawsdl="http://www.w3.org/ns/sawsdl">
      <types>
            <xsd:schema
targetNamespace="http://www.example.org/CustomerOrder/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
                  <xsd:element type="tns:customerInfo"
name="CustomerRegisterRequest" />
                  <xsd:complexType name="customerInfo" >
                        <xsd:sequence>
                              <xsd:element type="xsd:string"
name="customerName"

      sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/on
tology/purchaseorder#CustomerName"/>
                              <xsd:element type="xsd:string"
name="customerAddress" />
                              <xsd:element name="customerTelephone"
minOccurs="1" maxOccurs="unbounded"

      sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/on
tology/purchaseorder#Telephone">
                                    <xsd:simpleType>
                                          <xsd:restriction
base="xsd:string">
                                                <xsd:minLength
value="14" />
                                          </xsd:restriction>
                                    </xsd:simpleType>
                              </xsd:element>
                        </xsd:sequence>
                  </xsd:complexType>
                  <xsd:element name="CustomerRegisterResponse">
                        <xsd:complexType>
                              <xsd:sequence>
                                    <xsd:element type="xsd:string"
name="customerID" />
                              </xsd:sequence>
                        </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="OrderRequest">
                        <xsd:complexType>
                              <xsd:sequence>
                                    <xsd:element name="customerID"

      sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/on
tology/purchaseorder#CustomerID">
                                          <xsd:simpleType>
```

```
                                                      <xsd:restriction
base="xsd:string">
                                                            <xsd:pattern
value="[A-Z]{3}\d{10}" />
                                                      </xsd:restriction>
                                                </xsd:simpleType>
                                          </xsd:element>
                                          <xsd:element type="tns:item"
name="orderItem"
                                                maxOccurs="unbounded"
minOccurs="1" />
                                    </xsd:sequence>
                              </xsd:complexType>
                        </xsd:element>
                        <xsd:complexType name="item">
                              <xsd:all>
                                    <xsd:element type="xsd:string"
name="itemCode" />
                                    <xsd:element type="xsd:string"
name="itemName" />
                              </xsd:all>
                              <xsd:attribute type="xsd:integer"
name="quantity" />
                        </xsd:complexType>
                        <xsd:element name="OrderResponse">
                              <xsd:complexType>
                                    <xsd:sequence>
                                          <xsd:element name="orderID">
                                                <xsd:simpleType>
                                                      <xsd:restriction
base="xsd:string">
                                                            <xsd:maxLength
value="10" />
                                                      </xsd:restriction>
                                                </xsd:simpleType>
                                          </xsd:element>
                                          <xsd:element
type="tns:confirmation" name="orderStatus" />
                                    </xsd:sequence>
                              </xsd:complexType>
                        </xsd:element>
                        <xsd:simpleType name="confirmation">
                              <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="Confirmed" />
                                    <xsd:enumeration value="Pending" />
                                    <xsd:enumeration value="Rejected" />
                              </xsd:restriction>
                        </xsd:simpleType>
                        <xsd:element name="PaymentRequest">
                              <xsd:complexType>
                                    <xsd:sequence>
                                          <xsd:element name="orderID"
sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology
/purchaseorder#OrderID">
                                                <xsd:simpleType>
                                                      <xsd:restriction
base="xsd:string">
```

```
                                                                  <xsd:maxLength
value="14" />
                                                            </xsd:restriction>
                                                      </xsd:simpleType>
                                                </xsd:element>
                                                <xsd:element type="tns:payment"
name="paymentType" />
                                                <xsd:element
type="tns:creditCard" name="creditCard"
                                                      maxOccurs="1" minOccurs="0"
/>
                                    </xsd:sequence>
                              </xsd:complexType>
                        </xsd:element>
                        <xsd:simpleType name="payment">
                              <xsd:restriction base="xs:string">
                                    <xsd:enumeration value="Cash" />
                                    <xsd:enumeration value="Card" />
                              </xsd:restriction>
                        </xsd:simpleType>
                        <xsd:simpleType name="creditCard"
sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology
/purchaseorder#CreditCard">
                              <xsd:restriction base="xsd:string">
                              </xsd:restriction>
                        </xsd:simpleType>
                        <xsd:element type="tns:paymentStatus"
name="PaymentResponse" />
                        <xsd:simpleType name="paymentStatus">
                              <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="Completed" />
                                    <xsd:enumeration value="Waiting" />
                              </xsd:restriction>
                        </xsd:simpleType>
                  </xsd:schema>
            </types>
            <interface name="CustomerOrder">
                  <operation name="register"
pattern="http://www.w3.org/ns/wsdl/in-out">
                        <input element="tns:CustomerRegisterRequest" />
                        <output element="tns:CustomerRegisterResponse" />
                  </operation>
                  <operation name="order"
pattern="http://www.w3.org/ns/wsdl/in-out">
                        <input element="tns:OrderRequest" />
                        <output element="tns:OrderResponse" />
                  </operation>
                  <operation name="payment"
pattern="http://www.w3.org/ns/wsdl/in-out">
                        <input element="tns:PaymentRequest" />
                        <output element="tns:PaymentResponse" />
                  </operation>
            </interface>
            <binding type="http://www.w3.org/ns/wsdl/soap"
name="CustomerOrderSOAP"

            wsoap:protocol="http://www.w3.org/2006/01/soap11/bindings/HTTP/
"
```

```
            wsoap:version="1.1" interface="tns:CustomerOrder"
xmlns:wsoap="http://www.w3.org/ns/wsdl/soap">
            <operation
wsoap:soapAction="http://www.example.org/CustomerOrder/register"
                  ref="tns:register" />
      </binding>
      <service name="CustomerOrder" interface="tns:CustomerOrder">
            <endpoint name="CustomerOrderSOAP"
address="http://www.example.org/"
                  binding="tns:CustomerOrderSOAP" />
      </service>
</description>
```

ภาคผนวก ข
ตัวอย่างออนโทโลจีที่ใช้กำกับความหมาย

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY Ontology1344438200889
"http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.o
wl#" >
]>
<rdf:RDF
xmlns="http://www.semanticweb.org/ontologies/2012/7/Ontology134443820
0889.owl#"

xml:base="http://www.semanticweb.org/ontologies/2012/7/Ontology134443
8200889.owl"

xmlns:Ontology1344438200889="http://www.semanticweb.org/ontologies/20
12/7/Ontology1344438200889.owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology
rdf:about="http://www.semanticweb.org/ontologies/2012/7/Ontology13444
38200889.owl"/>

    <!--

////////////////////////////////////////////////////////////////////
//////////////////
    //
    // Datatypes
    //

////////////////////////////////////////////////////////////////////
//////////////////
     -->
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#CardType -->
    <rdfs:Datatype rdf:about="&Ontology1344438200889;cardType">
        <owl:equivalentClass>
            <rdfs:Datatype>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first>Amex</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first>MasterCard</rdf:first>
```

```
                                <rdf:rest>
                                    <rdf:Description>
                                        <rdf:type
rdf:resource="&rdf;List"/>
                                        <rdf:first>Visa</rdf:first>
                                        <rdf:rest
rdf:resource="&rdf;nil"/>
                                    </rdf:Description>
                                </rdf:rest>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdfs:Datatype>
        </owl:equivalentClass>
    </rdfs:Datatype>
    <!--

//////////////////////////////////////////////////////////////////////
//////////////////
    //
    // Object Properties
    //

//////////////////////////////////////////////////////////////////////
//////////////////
     -->
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#hasNumber -->
    <owl:ObjectProperty rdf:about="&Ontology1344438200889;hasNumber">
        <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
        <rdfs:range>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="&Ontology1344438200889;phone"/>
                <owl:someValuesFrom>
                    <rdfs:Datatype>
                        <owl:onDatatype rdf:resource="&xsd;string"/>
                        <owl:withRestrictions
rdf:parseType="Collection">
                            <rdf:Description>
                                <xsd:maxLength
rdf:datatype="&xsd;integer">14</xsd:maxLength>
                            </rdf:Description>
                        </owl:withRestrictions>
                    </rdfs:Datatype>
                </owl:someValuesFrom>
            </owl:Restriction>
        </rdfs:range>
    </owl:ObjectProperty>
    <!--

//////////////////////////////////////////////////////////////////////
//////////////////
    //
    // Data properties
    //
```

```
//////////////////////////////////////////////////////////////////////
//////////////////
     -->
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#hasCardType -->
    <owl:DatatypeProperty
rdf:about="&Ontology1344438200889;hasCardType">
        <rdfs:range rdf:resource="&Ontology1344438200889;cardType"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#id -->
    <owl:DatatypeProperty rdf:about="&Ontology1344438200889;id"/>
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#phone -->
    <owl:DatatypeProperty rdf:about="&Ontology1344438200889;phone"/>
    <!--

//////////////////////////////////////////////////////////////////////
//////////////////
    //
    // Classes
    //

//////////////////////////////////////////////////////////////////////
//////////////////
     -->

    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#CreditCard -->
    <owl:Class rdf:about="&Ontology1344438200889;CreditCard">
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="&Ontology1344438200889;hasCardType"/>
                <owl:someValuesFrom
rdf:resource="&Ontology1344438200889;cardType"/>
            </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#CustomerID -->
    <owl:Class rdf:about="&Ontology1344438200889;CustomerID">
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="&Ontology1344438200889;id"/>
                <owl:someValuesFrom>
                    <rdfs:Datatype>
                        <owl:onDatatype rdf:resource="&xsd;string"/>
                        <owl:withRestrictions
rdf:parseType="Collection">
```

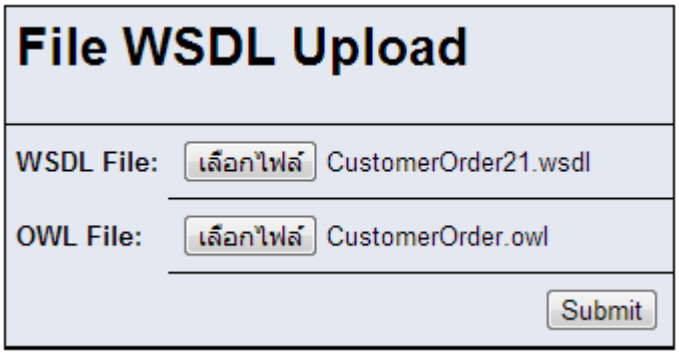
```
                            <rdf:Description>
                                <xsd:pattern>[A-
Z]{3}\d{10}</xsd:pattern>
                            </rdf:Description>
                        </owl:withRestrictions>
                    </rdfs:Datatype>
                </owl:someValuesFrom>
            </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#OrderID -->
    <owl:Class rdf:about="&Ontology1344438200889;OrderID">
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="&Ontology1344438200889;id"/>
                <owl:someValuesFrom>
                    <rdfs:Datatype>
                        <owl:onDatatype rdf:resource="&xsd;string"/>
                        <owl:withRestrictions
rdf:parseType="Collection">
                            <rdf:Description>
                                <xsd:maxLength
rdf:datatype="&xsd;integer">10</xsd:maxLength>
                            </rdf:Description>
                        </owl:withRestrictions>
                    </rdfs:Datatype>
                </owl:someValuesFrom>
            </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#Telephone -->
    <owl:Class rdf:about="&Ontology1344438200889;Telephone">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Restriction>
                        <owl:onProperty
rdf:resource="&Ontology1344438200889;hasNumber"/>
                        <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
                    </owl:Restriction>
                    <owl:Restriction>
                        <owl:onProperty
rdf:resource="&Ontology1344438200889;hasNumber"/>
                        <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxCardinality>
                    </owl:Restriction>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/ontologies/2012/7/Ontology1344438200889.ow
l#CustomerName -->
    <owl:Class rdf:about="&Ontology1344438200889;CustomerName">
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="&Ontology1344438200889;id"/>
                <owl:someValuesFrom>
                    <rdfs:Datatype>
                        <owl:onDatatype rdf:resource="&xsd;string"/>
                        <owl:withRestrictions
rdf:parseType="Collection">
                            <rdf:Description>
                                <xsd:maxLength
rdf:datatype="&xsd;integer">50</xsd:maxLength>
                            </rdf:Description>
                        </owl:withRestrictions>
                    </rdfs:Datatype>
                </owl:someValuesFrom>
            </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>
</rdf:RDF>
```

ภาคผนวก ค
การใช้งานเครื่องมือ

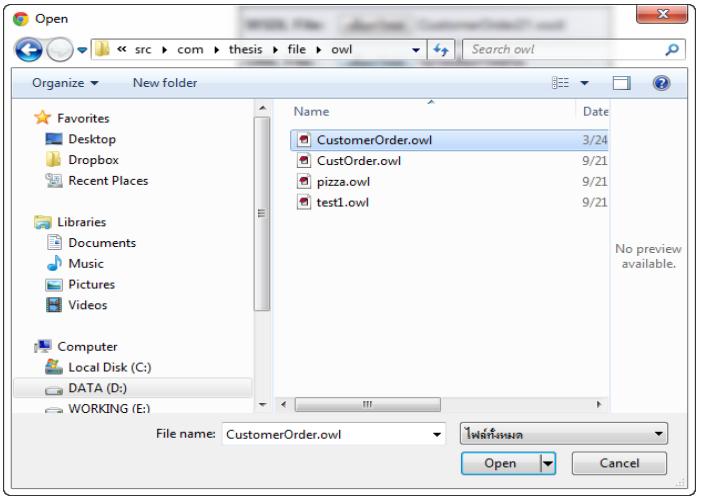
งานวิจัยนี้ได้พัฒนาเครื่องมือสนับสนุนแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตีในรูปแบบของเว็บแอพพลิเคชัน โดยมีขั้นตอนการใช้งานเครื่องมือดังต่อไปนี้

1. เมื่อผู้ใช้ทำการเปิดใช้งานเครื่องมือผ่านเว็บเบราว์เซอร์โดยผู้ใช้งานจะต้องระบุยูอาร์แอล “http://localhost:8080/ThesisWebTools/thesis/showUpload.action” เพื่อทำการค้นหาไฟล์ ซึ่งผู้ใช้งานจะต้องทำการระบุไฟล์ข้อมูล 2 ประเภทคือ ไฟล์วิสเดิลและไฟล์อาวล์ที่ใช้กำกับความหมาย ดังภาพที่ ค.1



ภาพที่ ค.1 หน้าจอการระบุไฟล์ข้อมูล

2. ในการระบุไฟล์ข้อมูลนั้นผู้ใช้สามารถเรียกไฟล์จากไดเรกทอรีที่ต้องการได้จากไฟล์เบราว์เซอร์ ดังภาพที่ ค.2



ภาพที่ ค.2 หน้าจอการระบุไฟล์ข้อมูล

3. เมื่อทำการเลือกไฟล์ทั้งสองประเภทเรียบร้อยแล้ว ให้ผู้ใช้ทำการกดปุ่ม Submit เพื่อทำการประมวลผลตามแบบจำลองการประเมินดัชนีความเสี่ยงต่อการโจมตีตามที่ได้นำเสนอไว้ในงานวิจัย เครื่องมือจะแสดงผลดังภาพที่ ค.3

Evaluate Result

Element Name	Data Type	Restriction	Annotation	Strong Declaration		Weak Declaration			
						Unavoidable Risk		Risk that can be Mitigated	
				Command Injection	DOS	Command Injection	DOS	Command Injection	DOS
customerName	string	N	Y			Y			Y
customerAddress	string	N	N			Y	Y		
customerTelephone	string	Y	Y			Y	Y		Y
customerID	string	Y	Y	Y	Y				
itemCode	string	Y	N			Y	Y		
itemName	string	Y	N			Y	Y		
quantity	integer	Y	N	Y			Y		
orderID	string	Y	Y		Y	Y			
payment	string	Y	N	Y	Y				
creditCard	string	N	Y					Y	Y

Unavoidable Command Injection Score:	**16.2**
Mitigated Command Injection Score:	**2.7**
Unavoidable DOS Score:	**8.5**
Mitigated DOS Score:	**5.1**
Total Score:	**32.5**

ภาพที่ ค.3 หน้าจอการแสดงการประเมินดัชนีความเสี่ยงต่อการโจมตีสำหรับเว็บเซอร์วิซ

ประวัติผู้เขียนวิทยานิพนธ์

นายสาววันวิษา โพธิ์เจริญ เกิดเมื่อวันที่ 31 สิงหาคม พ.ศ. 2524 ที่จังหวัดนครนายก สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2552