กรอบการทำงานใหม่ในการป้องกันการโจมตีที่ใช้เอกซ์เอ็มแอล

The New Framework to Defend Against XML-Based Attacks

ชาตรี ทองวรรณ[1] และ ดร.วรพล ลีลาเกียรติสกุล[2]

[1]นิสิตปริญญาโท สาขาวิศวกรรมเครือข่าย คณะวิทยาการและเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีมหานคร กรุงเทพฯ

[2]ผู้ช่วยศาสตราจารย์ ประจำคณะวิทยาการและเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีมหานคร กรุงเทพฯ

ABSTRACT – This paper presents the new framework that has the ability to detect and prevent attacks that are based on XML to Web service. These would be Oversize Payload attack, Recursive Payload attack, Parameter Tampering attack, Buffer Overflow attack or Replay Attack. The framework consists of three stages. The first stage is to learn XML documents leading to create XML schema. The second stage is to set all needed parameters and the third stage is to validate the XML format. The framework has been developed using Apache Tomcat, Apache Axis2, MySQL and JAVA language. For the experiment, it shows that mechanisms of the framework work effectively in detection and preventing attacks that are based on XML attacks.

KEY WORDS -- Data validation; XML-based attack

บทคัดย่อ -- บทความนี้นำเสนอกรอบการทำงาน (Framework) ที่มีความสามารถในการตรวจสอบและป้องกันการโจมตีเว็บเซอร์วิส (Web service) ที่ใช้เอกซ์เอ็มแอล (XML) ไม่ว่าจะเป็นการโจมตีแบบ Oversize Payload, Recursive Payload, Parameter Tampering, Buffer Overflow หรือ Replay Attack สำหรับกรอบการทำงานนี้ ประกอบไปด้วยสามส่วนคือ ส่วนที่หนึ่งจะเป็นส่วนของการเรียนรู้รูปแบบของเอกซ์เอ็มแอลที่เป็นไปได้เพื่อนำมาสร้างเอกซ์เอ็มแอลสคีมา (XML schema) ส่วนที่สองจะเป็นส่วนของการกำหนดค่าพารามิเตอร์ (Parameter) ที่จำเป็นในการตรวจสอบและส่วนที่สามจะเป็นส่วนของการตรวจสอบรูปแบบเอกซ์เอ็มแอล ส่วนของการพัฒนาใช้ Apache Tomcat, Apache Axis2, MySQL และ JAVA ในการพัฒนา ผลการทดสอบแสดงให้เห็นว่ากลไกของกรอบการทำงาน ทำงานได้อย่างมีประสิทธิภาพ ในการตรวจสอบและป้องกันการโจมตีเว็บเซอร์วิสที่ใช้เอกซ์เอ็มแอล

คำสำคัญ --การตรวจสอบข้อมูล; การโจมตีบนพื้นฐานเอกซ์เอ็มแอล

1. บทนำ

ในปัจจุบันเทคโนโลยีสารสนเทศและการสื่อสารได้รับการพัฒนามาอย่างต่อเนื่อง ในปัจจุบันถือว่าเป็นยุคของการสื่อสารที่ไร้พรมแดน อินเทอร์เน็ตได้เข้ามาทำให้การสื่อสารข้อมูลเป็นไปอย่างสะดวกและรวดเร็ว จึงทำให้องค์กรต่างๆ ได้มีการนำเอาอินเทอร์เน็ตเข้ามาใช้งานภายในองค์กร ผลที่ได้คือทำให้องค์กรเหล่านี้สามารถที่จะติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันไม่ว่า ลูกค้า คู่ค้า หรือหน่วยงานของรัฐผ่านอินเทอร์เน็ตได้ตลอด 24 ชั่วโมง ทำให้มีความสะดวกและรวดเร็วในการดำเนินงาน แต่ในความเป็นจริงแล้วการแลกเปลี่ยนข้อมูลกันนั้น แต่ละองค์กรมักจะมีรูปแบบข้อมูลที่แตกต่างกันหรือมีโปรแกรมประยุกต์ที่อาจถูกพัฒนาบนแพลตฟอร์ม (Platform) ที่ต่างกัน แม้ในอดีตจะสามารถเชื่อมต่อระบบที่มีความแตกต่างกันได้ แต่ก็มีค่าใช้จ่ายที่สูงและมีความซับซ้อนมาก ด้วยเหตุนี้เองเว็บเซอร์วิส จึงถูกนำใช้ในการเชื่อมระบบที่มีความแตกต่างกันให้สามารถทำงานร่วมกัน โดยที่ระบบเดิมยังสามารถทำงานได้ปกติและระบบใหม่ที่เพิ่มเข้ามาก็สามารถทำงานร่วมกับระบบเดิมได้อย่างมีประสิทธิภาพ

เมื่อเว็บเซอร์วิสได้รับความนิยมนำมาใช้งานมากขึ้นเรื่อยๆ เว็บเซอร์วิสจึงได้ถูกนำมากำหนดให้เป็นมาตรฐานแต่มาตรฐานต่างๆ ของเว็บเซอร์วิสที่กำหนดขึ้นมาในปัจจุบันยังไม่สามารถรับประกัน ไม่ว่าเรื่องประสิทธิภาพการทำงาน ระบบรักษาความปลอดภัย ความคงทนของระบบ เนื่องจากยังไม่มีอะไรที่เป็นหลักประกันได้ว่าเมื่อนำเว็บเซอร์วิสมาใช้งานแล้วระบบจะสามารถให้บริการได้ตลอด 24 ชั่วโมง โดยที่ระบบจะไม่ถูกขัดขวางการให้บริการจากผู้ไม่หวังดี แม้ว่าในปัจจุบันจะมีการพัฒนามาตรฐานในการรักษาความปลอดภัยของเว็บเซอร์วิสขึ้นมาไม่ว่าจะเป็น WS-Security, WS-Trust และอื่นๆ [3] [4] ซึ่งมาตรฐานเหล่านี้ต่างมุ่งเน้นในการรักษาความปลอดภัยให้กับข้อความ (Message) เท่านั้น ซึ่งยังไม่เพียงพอสำหรับการรักษาความปลอดภัยให้กับระบบ เนื่องจากการติดต่อสื่อสารและแลกเปลี่ยนข้อมูลระหว่างระบบอื่นกับเว็บเซอร์วิสต่างอาศัย เอกซ์เอ็มแอลเป็นสื่อกลางในการแลกเปลี่ยนข้อมูลจึงทำให้เว็บเซอร์วิสมีความเสี่ยงต่อการโจมตีจากผู้ไม่หวังดีที่ใช้เอกซ์เอ็มแอล ในการโจมตีที่อาจมีผลทำให้เว็บเซอร์วิสถูกขัดขวางการให้บริการ ดังนั้นผู้วิจัยจึงได้นำเสนอกรอบการทำงานที่ใช้ในการป้องกันการโจมตีที่เกิดจากการใช้เอกซ์เอ็มแอล

โครงสร้างของบทความที่เหลือจะเป็นดังนี้ หัวข้อที่ 2 เป็นส่วนของการวิจารณ์วรรณกรรมที่ประกอบด้วย คำอธิบายเกี่ยวกับเทคโนโลยีเว็บเซอร์วิสและจุดอ่อนในเว็บเซอร์วิสที่ใช้เอกซ์เอ็มแอลโจมตี และงานวิจัยที่เกี่ยวข้องการป้องกันการโจมตีที่ใช้เอกซ์เอ็มแอล หัวข้อที่ 3 เป็นการอธิบายแนวคิดการออกแบบและหลักการทำงานของกรอบการทำงานสำหรับป้องกันการโจมตีที่ใช้เอกซ์เอ็มแอล ส่วนการทดสอบกรอบการทำงานและผลการทดสอบจะนำเสนอในหัวข้อที่ 4 และบทสรุปและข้อเสนอแนะจะอยู่ในหัวข้อที่ 5

2. พื้นฐานและงานวิจัยที่เกี่ยวข้อง

2.1. เทคโนโลยีเว็บเซอร์วิส

เว็บเซอร์วิส คือระบบซอฟต์แวร์ที่ออกแบบมาเพื่อสนับสนุนการทำงานร่วมกันระหว่างคอมพิวเตอร์ผ่านระบบเครือข่าย โดยใช้ภาษาเอกซ์เอ็มแอลในการติดต่อสื่อสารและมีการอธิบายส่วนของการติดต่อเว็บเซอร์วิสไว้ในรูปแบบที่คอมพิวเตอร์เข้าใจ (ระบุไว้ในดับเบิลยูเอสดีแอล (WSDL)) การติดต่อสื่อสารระหว่างระบบอื่นกับเว็บเซอร์วิส จะดำเนินการในลักษณะที่กำหนดโดยใช้ SOAP message ซึ่งปกติจะขนส่งไปบนเอชทีทีพี (HTTP) [1] [2] เนื่องจากเว็บเซอร์วิสมีพื้นฐานอยู่บนกลุ่มของโพรโทคอลมาตราฐานและเทคโนโลยี โดยแต่ละองค์ประกอบของเว็บเซอร์วิสจะถูกนำมาใช้ในการกำหนดรูปแบบในการสื่อสาร ไม่ว่าจะเป็น เอกซ์เอ็มแอล, โซพ (SOAP), ดับเบิลยูเอสดีแอล (WSDL), ยูดีดีไอ (UDDI) หรือ เอกซ์เอ็มแอลสคีมา ซึ่งสามารถอธิบายหน้าที่แต่ละส่วนได้ดังนี้

1) XML (Extensible Markup Language) เป็นภาษามาร์กอัป (markup language) ถูกพัฒนาโดยองค์กร W3C โดยมีจุดประสงค์ เพื่อเป็นภาษาที่ใช้ในการอธิบายข้อมูลสำหรับติดต่อสื่อสารกันในระบบที่มีความแตกต่างกัน เนื่องจากภาษาเอกซ์เอ็มแอลจัดเก็บข้อมูลที่เป็นมาตรฐานและเพิ่มในส่วนของการอธิบายข้อมูลด้วยทำให้ทุกๆ แพลตฟอร์ม สามารถเข้าใจความหมายของข้อมูลที่จัดเก็บไว้ภายในเอกซ์เอ็มแอล [5]

2.) SOAP (Simple Object Access Protocol) เป็นส่วนของโพรโทคอลที่ได้ถูกนำมาใช้งานกันอย่างกว้างขวางในการส่งข้อมูลแบบข้อความ ในปี 2000 องค์กร W3C ก็ได้ประกาศให้เป็นมาตรฐาน โดยให้เป็นโพรโทคอลสำหรับการสื่อสารข้อมูลในรูปแบบข้อความโดยโซพจะห่อหุ้มข้อมูลที่จะรับส่งบนพื้นฐานของเอกซ์เอ็มแอล ในการส่งข้อมูลออกไปยังเครือข่ายโซพ จะอาศัยโพรโทคอลอื่น ในการขนส่งแทนในปัจจุบันนิยมขนส่งผ่านเอชทีทีพี [6] [7]

3) WSDL (Web Services Description Language) เป็นภาษาที่มีพื้นฐานเป็นเอกซ์เอ็มแอล ที่ถูกนำมาใช้ในการอธิบายส่วนของการติดต่อเว็บเซอร์วิส ซึ่งภายในเอกสารจะประกอบไปด้วยส่วนของคำอธิบายต่าง ๆ ไม่ว่าจะเป็นตำแหน่งของเว็บเซอร์วิส โพรโทคอลที่ใช้ในการขนส่ง รูปแบบข้อมูล การดำเนินการ พฤติกรรมของการติดต่อสื่อสาร และอื่นๆ [6] [8]

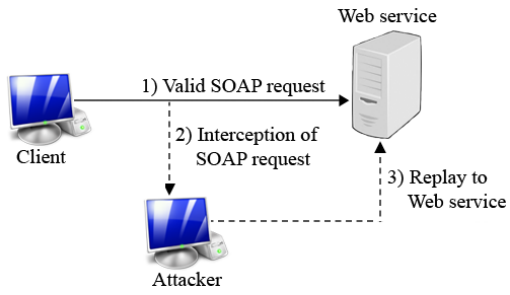
4) UDDI (Universal Description, Discovery and Integration) คือส่วนของการลงทะเบียนเว็บเซอร์วิสโดยจะจัดเก็บข้อมูลต่างๆ ของผู้ลงทะเบียนและเว็บเซอร์วิสไว้ไม่ว่าจะเป็น ชื่อบริษัท เบอร์โทร ชื่อผู้ติดต่อ และอื่นๆ พร้อมทำการเชื่อมโยงไปยังเอกสารดับเบิลยูเอสดีแอลด้วย โดยยูดีดีไอยังได้จัดเตรียมกลไกต่าง ๆ ไว้เช่น การลงทะเบียน การค้นหา แต่สำหรับยูดีดีไอ นี้อาจไม่จำเป็นเมื่อผู้ร้องขอทราบตำแหน่งของเว็บเซอร์วิส [6] [9]

5) XML schema บางที่อาจเรียกว่าเป็นภาษานิยามโครงสร้าง (XML schema definition) ถูกพัฒนาโดยองค์กร W3C เพื่อใช้นิยามโครงสร้างของเอกสารเอกซ์เอ็มแอลและข้อกำหนดต่าง ๆ ของข้อมูลที่จะถูกจัดเก็บไว้ภายในเอกสาร เอกซ์เอ็มแอลไม่ว่าจะเป็นโครงสร้างเอกซ์เอ็มแอล รูปแบบข้อมูล และยังสามารถใส่ข้อกำหนดให้กับข้อมูลที่จัดเก็บในแต่ละอิลิเมนต์ (element) หรือแอตทริบิวต์ (attribute) [5] [10]

2.2. การโจมตีบนเว็บเซอร์วิส

มาตราฐานต่างๆ ของเว็บเซอร์วิสไม่ว่า โซพ ดับเบิลยูเอสดีแอล ยูดีดีไอ ต่างมีพื้นฐานเป็นเอกซ์เอ็มแอล แต่โดยธรรมชาติของเอกซ์เอ็มแอล จะจัดเก็บข้อมูลในลักษณะข้อความ ถือว่าเป็นข้อเสียเปรียบอย่างหนึ่งที่ผู้ไม่หวังดีอาจจะนำเอาไปใช้ในการโจมตี ซึ่งการโจมตีที่ใช้เอกซ์เอ็มแอล มักเป็นการโจมตีเพื่อขัดขวางการทำงาน

1) Replay Attack เป็นการโจมตีที่เกิดขึ้นเมื่อผู้ไม่หวังดีทำการคัดลอก Message ที่เกิดขึ้นจากการสนทนากันระหว่างผู้ร้องขอกับเว็บเซอร์วิสแล้วทำการส่ง Message นั้นซ้ำ [11] [12] ดังแสดงดังใน รูปที่ 1



รูปที่ 1. แสดงการโจมตีแบบ Replay Attack

2) Buffer Overflow เป็นการโจมตีที่ผู้ไม่หวังดีทำการส่งข้อมูลมาแล้วเกินขอบเขตที่ประเภทข้อมูลนั้นรับได้ [12] [13] [14] [21] ดังแสดงตัวอย่างใน รูปที่ 2

```
<SOAP:Body>
  <tns:calNumber xmlns:tns=".....">
    <tns:number>3147483647</tns:number >
  </tns:calNumber >
</SOAP:Body>
```

รูปที่ 2. แสดงการโจมตีแบบ Buffer Overflow

3) Parameter Tampering ในเอกซ์เอ็มแอล จะมีส่วนที่เรียกว่า CDATA โดยส่วนนี้จะเป็นบล็อกที่อนุญาตให้ใส่ข้อความหรืออักขระพิเศษอื่นใดไว้ภายในบล็อกนี้ โดยที่ตัวแปลภาษาเอกซ์เอ็มแอล จะไม่นำไปตีความหมาย ดังนั้นผู้ที่ไม่หวังดีก็อาจจะนำเอาคุณสมบัตินี้ไปใช้ในการโจมตีเว็บเซอร์วิสได้ [6] [11] [13] ดังแสดงใน รูปที่ 3

```
<tns:name><![CDATA[<,&,# ]]></tns:name >
```

รูปที่ 3. แสดงการโจมตีแบบ Parameter Tampering

4) Recursive Payload ในโครงสร้างของเอกซ์เอ็มแอลจะมีโครงสร้างแบบลำดับชั้นและเอกซ์เอ็มแอลยังอนุญาตให้วางโครงสร้างแบบซ้อนทับกันได้ จากคุณสมบัตินี้ผู้ไม่หวังดีอาจจะนำไปใช้ในการโจมตี โดยอาจจะสร้างเอกสารเอกซ์เอ็มแอลที่มีอิลิเมนต์ซ้อนทับกัน ที่อาจจะซ้อนทับกันเป็น 10,000 หรือ 100,000 อิลิเมนต์ เพื่อมุ่งหวังโจมตีในส่วนของตัวจัดการเอกสารเอกซ์เอ็มแอลให้ทำงานผิดพลาด [12] [13] [14] ดังแสดงในรูปที่ 4

```
<credit_card_NO>
   <credit_card_NO>
      <credit_card_NO>
        <credit_card_NO>
          ...
</credit_card_NO>
```

รูปที่ 4. แสดงการโจมตีแบบ Recursive Payload

5) Oversize Payload เป็นการโจมตีที่ผู้ไม่หวังดีพยายามส่งเอกสารเอกซ์เอ็มแอลขนาดใหญ่ มาให้เว็บเซอร์วิสประมวลผล ซึ่งอาจจะมีขนาดหลายร้อยเมกะไบต์ เป็นผลทำให้ตัวจัดการเอกสารเอกซ์เอ็มแอล ทำงานผิดพลาด [12] [13] [14] ดังแสดงใน รูปที่ 5

```
<transaction>
  <total>1000.00</total>
  <item>data 1</item>
  <item>data 2</item>
  <item>data 3</item>
    ...
</transaction>
```

รูปที่ 5. แสดงการโจมตีแบบ Oversize Payload

2.3. งานวิจัยที่เกี่ยวข้อง

เนื่องจากการแลกเปลี่ยนข้อมูลระหว่างระบบอื่นกับเว็บเซอร์วิสจะใช้ภาษาเอกซ์เอ็มแอลเป็นสื่อกลางในการแลกเปลี่ยน [1] [2] โดยธรรมชาติของเอกซ์เอ็มแอลแล้วจะจัดเก็บข้อมูลในลักษณะข้อความ ดังนั้นข้อมูลที่แลกเปลี่ยนระหว่างเว็บเซอร์วิสจะถูกอธิบายด้วยเอกซ์เอ็มแอล แล้วส่งมาให้เว็บเซอร์วิสประมวลผล ถ้าข้อมูลนั้นขาดการตรวจสอบความถูกต้องที่ดี อาจทำให้ผู้ไม่หวังดีนำไปใช้ในการโจมตี [12] [16] โดยทำการส่งเอกสารเอกซ์เอ็มแอลที่เป็นอันตรายมา เพื่อหวังผลในการขัดขวางการทำงานเว็บเซอร์วิสไม่ว่าเป็นการโจมตีแบบ Oversize Payload, Recursive Payload, Parameter Tampering, Buffer Overflow หรือ Replay Attack [12] [13] [14] แม้ในปัจจุบันจะมีการพัฒนามาตรฐานในการรักษาความปลอดภัยของ เว็บเซอร์วิสขึ้นมาไม่ว่าจะเป็น WS-Security, WS-Trust และอื่นๆ [3] [4] ซึ่งมาตรฐานเหล่านี้ต่างมุ่งเน้นในการรักษาความปลอดภัยให้กับ Message เท่านั้น ในปัจจุบันก็ได้มีการศึกษาวิจัย ในการดำเนินการป้องกันการโจมตีที่ใช้เอกซ์เอ็มแอล จำนวนมากดังนี้

Nils Gruschka และ Norbert Luttenberger [15] ได้เสนอการป้องกันการโจมตีเว็บเซอร์วิสจาก DoS Attacks โดยการตรวจสอบความถูกต้องของ SOAP message ด้วยการใช้ความสามารถของเอกสารเอกซ์เอ็มแอลสคีมา ในการตรวจสอบ โดยระบบจะนำเอาเอกสารดับเบิลยูเอสดีแอลมาสกัดเอาบางส่วนของเอกสารมาสร้างเอกซ์เอ็มแอลสคีมาสำหรับตรวจสอบความถูกต้องของ SOAP message แต่ก็มีข้อจำกัด ระบบไม่สามารถรองรับรูปแบบของ SOAP message ในรูปแบบใหม่ ๆ ได้

Vipul Patel, Radhesh Mohandas และ Alwyn R. Pais [4] ได้เสนอรูปแบบการโจมตีบนเว็บเซอร์วิสและแผนการเพื่อลดทอนการโจมตีเว็บเซอร์วิสไม่ว่าจะเป็น XML Injection, XSS Injection และ XML Re-Writing โดยการตรวจสอบความถูกต้องของ SOAP message ด้วยเอกสารเอกซ์เอ็มแอลสคีมา โดยระบบจะนำเอาเอกสารดับเบิลยูเอสดีแอลมาสกัดเอาบางส่วนของเอกสารมาสร้างเอกซ์เอ็มแอลสคีมาที่สอดคล้องเก็บไว้ เมื่อมีการร้องขอเข้ามาระบบก็จะทำการคัดเลือกเอกสารเอกซ์เอ็มแอลสคีมาที่สอดคล้องมาตรวจสอบ พร้อมสร้างล็อก (Log) ของการร้องขอเก็บไว้ด้วย กรณีที่การร้องขอรูปแบบใหม่ๆ ระบบก็อาจจะปฏิเสธการร้องขอนั้น แต่ระบบจะนำเอาล็อกที่เก็บไว้ มาสร้างเอกสารเอกซ์เอ็มแอลสคีมาเก็บไว้สำหรับใช้ในการตรวจสอบในคราวต่อไป

Rafael Bosse Brinhosa, Carla Merkle Westphall, Carlos Becker Westphall, Daniel Ricardo dos Santos และ Fabio Grezele [16] ได้เสนอแบบอย่างการตรวจสอบรูปแบบของข้อมูลเข้าสำหรับเว็บเซอร์วิส เพื่อป้องกันการโจมตีไม่ว่าจะเป็นการโจมตีแบบ XSS injection และ SQL injection โดยการตรวจสอบข้อมูลที่ส่งมาด้วยเอกสารเอกซ์เอ็มแอลสคีมาร่วมกับเอกสารข้อกำหนดซึ่งเอกสารข้อกำหนดจะใช้นิยามขอบเขตข้อมูลที่ยอมรับได้และข้อกำหนดอื่นๆ ในการตรวจสอบ

Irfan siddavatam และ Jayant Gadge [14] ได้นำเสนอกลไกทดสอบแบบกว้างสำหรับตรวจหาการโจมตีบนเว็บเซอร์วิส ที่ใช้เอกซ์เอ็มแอล ผู้วิจัยได้นำเสนอกรอบการทำงานในการตรวจหาการโจมตีที่ประกอบด้วยสามส่วนและมีฐานข้อมูลเป็นศูนย์กลาง ส่วนที่หนึ่งจะเป็นการจับ Message ของทุกการร้องขอที่เข้ามาแล้วจัดให้อยู่ในรูปแบบเอกซ์เอ็มแอลแล้วจัดเก็บไว้ในฐานข้อมูล ส่วนที่สองจะเป็นการกำหนดค่าพารามิเตอร์ที่จำเป็นในการตรวจจับการโจมตี ซึ่งค่าพารามิเตอร์ต่างๆ จะถูกกำหนดโดยผู้ดูแลระบบและส่วนที่สามจะเป็นส่วนของการวิเคราะห์การโจมตี โดยนำเอาเอกซ์เอ็มแอลที่จัดเก็บไว้ในส่วนที่หนึ่งมาวิเคราะห์หาการโจมตีด้วยอัลกอริทึมที่เหมาะสมร่วมกับค่าพารามิเตอร์ที่กำหนดในส่วนที่สอง

และงานวิจัยอื่นๆ ที่เกี่ยวข้องต่างมุ่งเน้นไปเรื่องของ xml firewall อย่างเช่น (Nedgty,2005) [17] ทางผู้วิจัยได้นำเสนอแนวทางการป้องกันการโจมตีเว็บเซอร์วิสสำหรับป้องกัน DoS attacks, XDoS attacks และ Buffer Overflow โดยการกรอง SOAP message ที่เข้ามาและทำการตรวจสอบว่า IP ที่ส่งคำร้องขอในการเรียกใช้งานเว็บเซอร์วิสว่ามีสิทธิในการเรียกใช้งานเว็บเซอร์วิสหรือไม่ (S-Wall,2011) [18] ทางผู้วิจัยได้นำเสนอแนวทางการป้องกันการโจมตีเว็บเซอร์วิสสำหรับป้องกันโจมตีที่ใช้เอกซ์เอ็มแอล โดยการใช้งานระบบผู้ใช้จะต้องเข้าระบบและมีการใช้กลไลของการควบคุมการเข้าถึงตามบทบาทแบบไดนามิก (D-RBAC) ในการควบคุมการเข้าถึงเว็บเซอร์วิสของผู้ใช้งานแต่ละคน และยังมีกระบวนการตรวจสอบการโจมตีที่เอกซ์เอ็มแอลด้วยอัลกอริทึมที่เหมาะสมและยังมีการตรวจสอบสถานะของผู้ใช้ด้วย

จากงานวิจัยที่เกี่ยวข้องด้านบนนี้ โดยส่วนใหญ่จะเป็นการนำเอาเอกสารดับเบิลยูเอสดีแอลมาวิเคราะห์แล้วสกัดเอาบางส่วนของเอกสารเพื่อนำมาสร้างเอกสารเอกซ์เอ็มแอลสคีมาเพื่อนำมาใช้สำหรับตรวจสอบ Message แต่เนื่องจากเอกสารดับเบิลยูเอสดีแอลมีข้อจำกัดในการอธิบายรูปแบบของข้อมูลที่ใช้ในการรับส่ง ซึ่งดับเบิลยูเอสดีแอลมักจะอธิบายในภาพรวมที่ไม่ได้ระบุรายละเอียดมากนัก ดังนั้นเมื่อนำมาใช้เป็นต้นแบบในการสร้างเอกสารเอกซ์เอ็มแอลสคีมาจึงไม่ได้รายละเอียดที่ครบถ้วนตามที่ต้องการ จึงถือว่ายังใช้ความสามารถของเอกซ์เอ็มแอลสคีมายังไม่เต็มที่ ตัวอย่างการอธิบายรูปแบบข้อมูลในการรับส่งของดับเบิลยูเอสดีแอล ดังแสดงในรูปที่ 6

```
<wsdl:types>
 ..............
 <xs:element name="readXML">
  <xs:complexType>
   <xs:sequence>
    <xs:element  name="readXML"
type="xs:anyType"/>
   </xs:sequence>
 </xs:complexType>
  </xs:element>
```

รูปที่ 6. แสดงเอกสารดับเบิลยูเอสดีแอลส่วนของการอธิบายรูปแบบข้อมูลที่ใช้รับส่งที่เป็นเอกซ์เอ็มแอล

ดังนั้นผู้วิจัยจึงได้นำเสนอกรอบการทำงานใหม่ที่มีความสามารถในการตรวจสอบและป้องกันการโจมตีเว็บเซอร์วิสที่ใช้เอกซ์เอ็มแอล โดยปรับปรุงข้อด้อยต่างๆ ที่เกิดขึ้น และเพิ่มความสามารถในการเรียนรู้รูปแบบเอกซ์เอ็มแอลแบบอัตโนมัติ ซึ่งจะทำให้กรอบการทำงานสามารถเรียนรู้รูปแบบของเอกสารเอกซ์เอ็มแอลรูปแบบใหม่ๆ ได้

3. กรอบการทำงานใหม่สำหรับป้องกันการโจมตีผ่านเอกซ์เอ็มแอล

การออกแบบกรอบการทำงานสำหรับการตรวจสอบและป้องกันการโจมตีเว็บเซอร์วิสที่ใช้เอกซ์เอ็มแอลประกอบไปด้วย 3 ส่วนหลักและมีฐานข้อมูลเป็นศูนย์กลาง ดังรูปที่ 7 กล่าวคือ

1) ส่วนที่หนึ่งจะเป็นส่วนของการเรียนรู้รูปแบบของข้อมูลที่เป็นไปได้ที่จะส่งมาให้เว็บเซอร์วิสประมวลผล ซึ่งตรงส่วนนี้จะถือว่าเป็นการดำเนินการโดยปราศจากการโจมตีใดๆ โดยผู้ทดสอบจะทำการส่งข้อมูลที่เป็นเอกซ์เอ็มแอลในรูปแบบต่าง ๆ ที่เป็นไปได้มาให้เว็บเซอร์วิสประมวลผล ระบบจะทำการจับเอา Message แล้วสกัดเอาเอกสารเอกซ์เอ็มแอล นำมาสร้างเอกสารเอกซ์เอ็มแอลสคีมาที่สอดคล้องเก็บไว้ในฐานข้อมูล โดยเอกสารเอกซ์เอ็มแอลสคีมานี้ จะถูกนำมาใช้ในการตรวจสอบข้อมูลในคราวต่อไปและนอกจากนี้ระบบยังดำเนินการสกัดเอาค่าพารามิเตอร์บางอย่าง เช่น ขนาดของ Message เล็กสุดหรือขนาด Message ใหญ่สุดที่ยอมรับได้ [14] [19].

2) ส่วนที่สองจะเป็นส่วนของการกำหนดค่าพารามิเตอร์ที่จำเป็นเพื่อนำไปใช้ในการตรวจสอบข้อมูลที่สามารถยอมรับได้หรือไม่อนุญาตให้มี [14] ดังตารางที่ 1

ตารางที่ 1. ตารางแสดงค่าพารามิเตอร์ต่างๆ ที่จะถูกกำหนดค่าโดยผู้ทดสอบระบบ

พารามิเตอร์	ใช้ในการตรวจสอบ
ขนาด Message (min , max)	Oversize Payload Recursive Payload
ขอบเขตอิลิเมนต์ (min, max)	Buffer Overflow
อักขระต้องห้าม เช่น &, >, <	Parameter Tampering
ช่วงชีวิต Message (วินาที)	Replay Attack
อัตรารับ Message ต่อ วินาที	SOAP Flooding

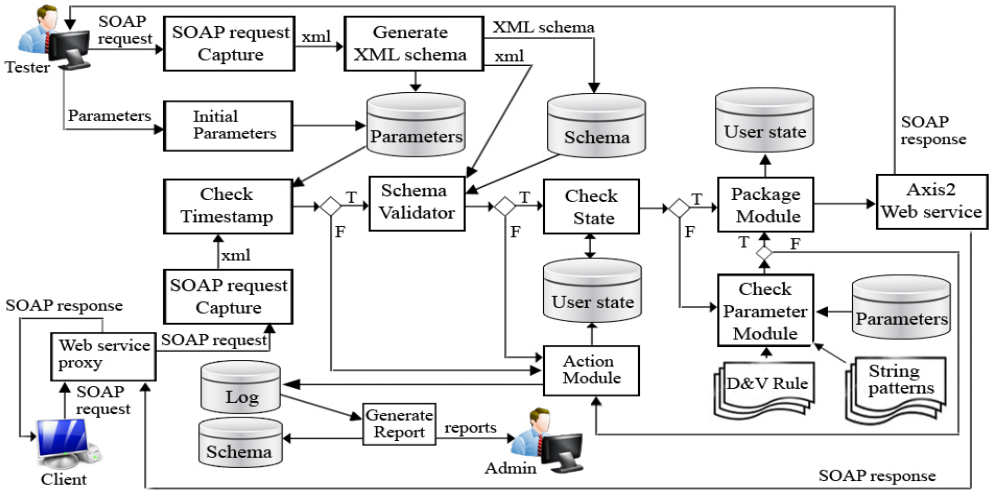
ส่วนของขอบเขตแต่ละอิลิเมนต์นี้ บางทีทางผู้ให้บริการอาจต้องการขอบเขตข้อมูลที่ตัวเองสามารถยอมรับได้ อย่างเช่น ข้อมูลอายุจะต้องอยู่ในช่วง 1 ถึง 150 เป็นต้น ก็สามารถกำหนดได้ตรงส่วนนี้

3) ส่วนที่สามจะเป็นการตรวจสอบเอกซ์เอ็มแอลแต่ละการร้องขอ โดยจะทำการคัดเลือกเอกสารเอกซ์เอ็มแอลสคีมาที่เหมาะสม มาตรวจสอบรูปแบบของเอกสารเอกซ์เอ็มแอลพร้อมนำค่าพารามิตอร์ที่กำหนดไว้มาร่วมในการตรวจสอบที่ประกอบไปด้วย

Check Timestamp ส่วนของการตรวจสอบเวลาชีวิตของ Message เมื่อผู้ร้องขอต้องการขอใช้บริการก็จะสร้าง Message ที่เรียกว่า SOAP request พร้อมทำการเพิ่ม Timestamp ในส่วนของ Header เมื่อ Message เดินทางมาถึง S-DV (State-based

data validation) ก็จะทำการจับเอา Message พร้อมนับจำนวนของ Message ที่ส่งมาจากต้นทางเดียวกัน ว่าในหนึ่งวินาทีส่ง

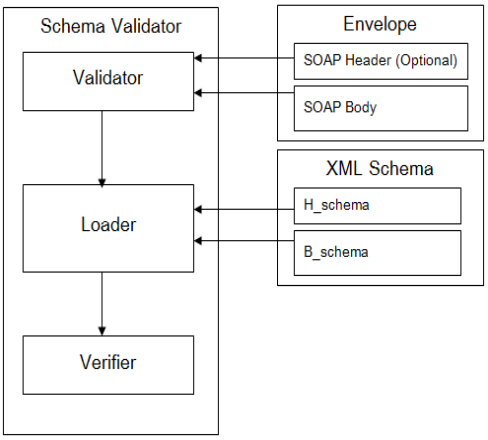
H_schema เป็นเอกสารเอกซ์เอ็มแอลสคีมาของ SOAP Header และส่วน B_schema เป็นเอกซ์เอ็มแอลสคีมาของ SOAP Body



รูปที่ 7. กรอบการทำงานที่นำเสนอสำหรับตรวจสอบข้อมูลบนพื้นฐานสถานะ (State-based data validation: S-DV)

Message มาเกินจำนวนที่กำหนดไว้หรือไม่ ถ้าเกินจะทำการยกเลิกการร้องของพร้อมสร้าง ล็อกและแก้ไขสถานะของผู้ร้องขอไปอยู่ในสถานะ lock โดยระบบจะสันนิษฐานว่าเป็นการโจมตีแบบ SOAP flooding [20] ถ้าไม่ใช่ก็จะสกัดเอา Timestamp ออกมาตรวจสอบกับเวลาปัจจุบัน ถ้าเกินช่วงเวลาที่กำหนด S-DV จะทำการยกเลิกการร้องขอพร้อมสร้างล็อกและลดสถานะของผู้ร้องขอลงหนึ่งระดับ ถ้าไม่เกินจะส่งไปทำการตรวจสอบในส่วนอื่นถัดไป [23]

Schema Validator [4] [16] [22] เป็นส่วนของการตรวจสอบความถูก ต้องของ Message ที่ส่งมาว่าเป็นไปตามรูปแบบที่กำหนดหรือไม่ โดยจะทำการคัดเลือกเอกสารเอกซ์เอ็มแอลสคีมาที่เหมาะสมที่ได้จัดเก็บไว้ในขั้นตอนเรียนรู้รูปแบบของเอกสารเอกซ์เอ็มแอล มาทำการตรวจสอบว่าเป็นไปตามรูปแบบที่กำหนดหรือไม่ กรณีที่มีเอกซ์เอ็มแอลรูปแบบใหม่ ๆ ส่งมาจะไม่มีเอกซ์เอ็มแอลสคีมาที่สอดคล้อง S-DV จะยกเลิกการร้องขอและสร้างล็อกเก็บไว้ โดยล็อกจะถูกนำมาสร้างเป็นรายงานให้ผู้ดูแลระบบ ดังนั้นผู้ดูแลระบบสามารถตรวจสอบแล้วยืนยัน S-DV ก็จะนำเอาล็อกมาสร้างเอกซ์เอ็มแอลสคีมาที่สอดคล้องเก็บไว้สำหรับตรวจสอบในคราวถัดไป ดังแสดงในรูปที่ 8 แสดงให้เห็นว่าเมื่อ Message เดินทางมาถึงระบบจะทำการจับแล้วสกัดเอาเอกสารเอกซ์เอ็มแอลออกมา แต่สำหรับส่วนของ SOAP Header อาจจะไม่มีก็ได้ จากนั้นจะทำการคัดเลือกเอกซ์เอ็มแอลสคีมาที่เหมาะสม มาตรวจสอบ



รูปที่ 8. แสดงการทำงานของ Schema Validator

Check State ส่วนของการตรวจสอบสถานะของการร้องขอใช้บริการเมื่อ Message ผ่านการตรวจสอบรูปแบบแล้วก็จะมาทำการตรวจสอบสถานะของการร้องขอว่ามีความน่าเชื่อเพียงใดเพราะว่า ถ้าการร้องขอนั้นมีความน่าเชื่อถือก็ไม่ควรที่จะไปตรวจสอบอย่างอื่นเพิ่ม เนื่องจาก Message ผ่านกระบวนการตรวจสอบรูปแบบ Message ด้วยเอกสารเอกซ์เอ็มแอลสคีมาก็ถือได้ว่าข้อมูลนั้นมีความปลอดภัยแล้วแน่นอนว่าการร้องขอที่มีความน่าเชื่อถือก็อาจจะเป็นการร้องขอที่ไม่น่าเชื่อถือในอนาคตก็ได้ ระบบการตรวจสอบจึงได้มีการกำหนดสถานะของการร้องของขึ้นมา 4 สถานะคือการร้องขอมีความน่าเชื่อถือ (Good), การร้องขอมีความน่าสงสัย

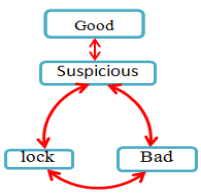
(Suspicious), การร้องขอที่ไม่ดี (Bad) และการร้องขอที่ไม่มีความปลอดภัยเลย (lock) ถ้าการร้องขออยู่ในสถานะ lock นี้จะถูกแก้ไขให้ไปอยู่ในสถานะอื่นได้จากผู้ดูแลระบบเท่านั้น สำหรับการร้องขออยู่ในสถานะ Good จะมีช่วงช่วงเวลาที่จำกัด เช่น 10 นาที 20 นาที หรือ 1 ชั่วโมงเป็นต้น เมื่อครบเวลาที่กำหนดก็จะย้ายมาอยู่ในสถานะ Suspicious แทนเพื่อทำการตรวจสอบว่ายังคงเป็นการร้องขอที่มีความน่าเชื่อถืออยู่หรือไม่ ส่วนการร้องขอใหม่ที่ยังไม่เคยเข้ามาในระบบเลยจะถูกกำหนดให้อยู่ในสถานะ Suspicious โดยอัตโนมัติ [18]

Action Module ส่วนการกำหนดการกระทำให้เหมาะสมกับข้อมูลนั้นๆ ว่าจะให้ทำอะไรขึ้นอยู่กับว่าข้อมูลที่ส่งมาให้นั้นเป็นข้อมูลอะไร

Check parameter module ส่วนวิเคราะห์พารามิเตอร์ส่วนนี้จะเป็นการวิเคราะห์ข้อมูลโดยละเอียด ทำให้การทำงานตรงส่วนนี้จำเป็นต้องใช้เวลาในการวิเคราะห์มากกว่าการทำงานของส่วนอื่นๆ โดยการวิเคราะห์จะเลือกอัลกอริทึมที่เหมาะสมที่ได้มีการกำหนดไว้ในส่วนของ D&V Rules (detection and verification rules) มาวิเคราะห์ร่วมกับค่าพารามิเตอร์ที่ได้จัดเก็บไว้ในขั้นตอนเรียนรู้หรือกำหนดค่าโดยผู้ทดทดสอบระบบมาร่วมวิเคราะห์ [18]

Generate reports ส่วนนี้จะเป็นการนำล็อกมาสร้างรายงานให้ผู้ดูแลระบบได้ทราบ สำหรับในส่วนล็อกที่เกิดขึ้นจาก Schema validator อาจมาจากสองประการคือ กรณีที่หนึ่งเป็นล็อกที่เกิดขึ้นจากเอกซ์เอ็มแอลที่ส่งมาในรูปแบบที่เป็นอันตราย กรณีที่สองเป็นล็อกที่เกิดขึ้นจากเอกซ์เอ็มแอลที่ส่งมาไม่สอดคล้องกับเอกซ์เอ็มแอลสคีมาที่มีอยู่ การร้องขอจึงถูกปฏิเสธตรงส่วนนี้ทางผู้ดูแลระบบสามารถตรวจสอบแล้วทำการยืนยัน S-DV ก็จะนำล็อกมาสร้างเอกซ์เอ็มแอลสคีมาเก็บไว้เพื่อใช้ในการตรวจสอบในครั้งต่อไป

สำหรับการตรวจสอบสถานะของการร้องขอนั้นจะตรวจสอบจาก IP address ของผู้ร้องขอที่ส่งการร้องขอใช้งานเว็บเซอร์วิส ดังนั้นสถานะของการร้องขอของแต่ละการร้องขออาจจะอยู่ในสถานะใดสถานะหนึ่งขึ้นอยู่กับการตรวจสอบข้อมูลที่ส่งมาว่าเป็นอย่างไร ดังแสดงใน รูปที่ 9



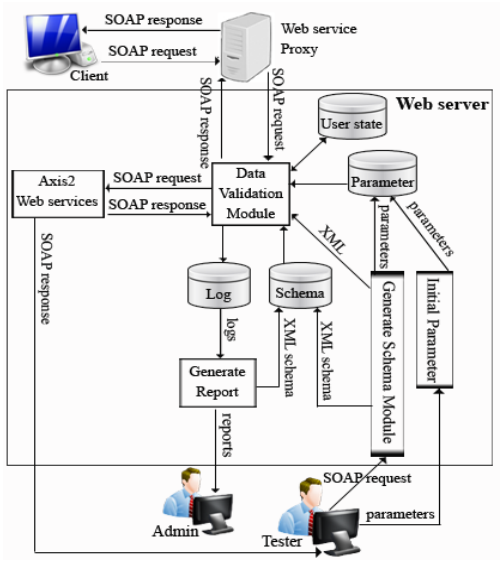
รูปที่ 9. แสดงการเปลี่ยนสถานะของการร้องขอ

- การร้องขอใช้งานเว็บเซอร์วิสที่ถูกส่งมาจากผู้ร้องขอที่เข้ามาในระบบเป็นครั้งแรกจะถูกกำหนดให้อยู่ในสถานะ Suspicious
- สถานะ Suspicious จะเปลี่ยนสถานะไปเป็น Good ได้กรณีเดียวเท่านั้นคือ การร้องขอใช้งานเว็บเซอร์วิสที่ผ่านการตรวจสอบจาก Check parameter module เท่านั้น
- สถานะ Suspicious จะเปลี่ยนสถานะไปเป็น Bad เมื่อการร้องขอนั้นไม่ผ่านการตรวจสอบจากส่วนใดส่วนหนึ่งของระบบ การร้องขอนั้นก็จะถูกลดสถานะลงมาเป็น Bad ในทำนองเดียวกันการร้องขอที่อยู่ในสถานะ Suspicious จะเปลี่ยนสถานะไปเป็น lock เมื่อการร้องขอนั้นถูกสงสัยว่าจะเป็นการส่งการร้องขอแบบ Flooding
- สถานะ Bad จะเปลี่ยนสถานะไปเป็น Suspicious เมื่อการร้องขอนั้นผ่านการตรวจสอบจาก Check parameter module ก็แสดงว่าการร้องขอนั้นมีความน่าเชื่อถือเพิ่มขึ้นแต่ยังมีความน่าสงสัยอยู่และจะเปลี่ยนสถานะไปเป็น lock เมื่อการร้องขอนั้นไม่ผ่านการตรวจสอบจากส่วนใดส่วนหนึ่งของระบบจะถูกสงสัยว่าเป็นการโจมตีแน่นอน
- สถานะ lock จะเปลี่ยนไปเป็นสถานะใด จะขึ้นจากการพิจารณาของผู้ดูแลระบบว่าเห็นสมควรให้ไปอยู่ในสถานะใด
- สถานะ Good จะเปลี่ยนสถานะมาเป็น Suspicious เมื่อหมดช่วงระยะเวลาที่กำหนด

4. การทดลอง

ในการทดลองการตรวจสอบและคัดกรองเอกซ์เอ็มแอลได้มีการพัฒนากรอบการทำงานและจัดสภาพแวดล้อมในการทดลองดังรูปที่ 10 ในแต่ละองค์ประกอบย่อยๆ ของกรอบการทำงานได้มีการแบ่งแยกในการการทดสอบ โดยใช้กรณีทดสอบที่กำหนดขึ้นเองในช่วงขั้นตอนของการพัฒนาเพื่อให้แน่ใจว่าแต่ละองค์ประกอบสามารถทำหน้าที่ได้อย่างถูกต้อง หลังจากนั้นได้นำเอาแต่ละองค์ประกอบทั้งหมดมารวมกันให้เป็นกรอบการทำงานสำหรับการทดสอบบนเครือข่ายดังรูปที่ 10 ในการทดสอบกรอบการทำงานบนเครือข่าย ประกอบไปด้วยโปรแกรมประยุกต์ของไคลเอนต์ (Client), เว็บเซอร์วิสพร็อกซี

(Web service proxy), เว็บเซิร์ฟเวอร์ (web server) และเว็บเซอร์วิสทั้งหมดเชื่อมต่อกันโดยตรง



รูปที่ 10. แสดงสภาพแวดล้อมของการทดลอง

โปรแกรมประยุกต์ของไคลเอนต์ที่ใช้เรียกใช้งานเว็บเซอร์วิสผ่านเว็บเซอร์วิสพร็อกซีพัฒนาด้วยภาษา JAVA และเอกซ์เอ็มแอล

เว็บเซอร์วิสพร็อกซี พัฒนามาจาก Squid Proxy server ที่ติดตั้งอยู่บนระบบปฏิบัติการ Linux โดยจะเป็นตัวกลางที่คอยรับทุกการร้องขอใช้งานเว็บเซอร์วิสเพื่อลดภาระการทำงานของเว็บเซิร์ฟเวอร์

เว็บเซอร์วิส พัฒนาด้วยภาษา JAVA และใช้ Apache axis2 ในการจัดการกับ SOAP message

การตรวจสอบความถูกต้องของ SOAP message (Data Validation module) โดยจะทำการตรวจสอบทุกการร้องขอที่เข้ามาว่ามีความปลอดภัยหรือไม่ ก่อนที่จะถูกส่งไปให้เว็บเซอร์วิสประมวลผลซึ่งพัฒนาด้วยภาษา JAVA และใช้ Apache axis2 ในการจัดการกับ SOAP message การตรวจสอบความถูกต้องของ SOAP message จะใช้ความสามารถของเอกซ์เอ็มแอลสคีมาร่วมกับค่าพารามิเตอร์ที่จำเป็นที่กำหนดโดยผู้ทดสอบระบบและใช้อัลกอริทึมที่เหมาะสมในการตรวจสอบ ในกรอบการทำงานยังมีการตรวจสอบสถานะของผู้ขอใช้เว็บเซอร์วิสด้วยว่ามีความน่าเชื่อถือหรือไม่ ถ้าผู้ขอใช้บริการมีความน่าเชื่อถือน้อยก็จะทำการตรวจสอบ SOAP message โดยละเอียดตรงส่วนนี้จะใช้เวลาในการตรวจสอบมาก ถ้าผลของการตรวจสอบความถูกต้องของ SOAP message แล้วไม่ถูกต้องตามข้อกำหนด การร้องขอนั้นจะถูกยกเลิกไปพร้อมสร้างล็อกเก็บไว้ด้วย

ฐานข้อมูล จะเป็นศูนย์กลางที่คอยจัดเก็บข้อมูลต่างๆ ที่จำเป็นในการตรวจสอบ พัฒนาขึ้นด้วย MySQL server

การสร้างเอกซ์เอ็มแอลสคีมา (Generate Schema Module) จะเป็นส่วนของการเรียนรู้รูปแบบของ SOAP message ในรูปแบบต่างๆ ที่เป็นไปได้ เพื่อนำมาสร้างเอกสารเอกซ์เอ็มแอลสคีมา ที่ใช้ในการตรวจสอบความถูกต้องของ SOAP message ในคราวต่อไป พัฒนาด้วยภาษา JAVA

การกำหนดค่าพารามิเตอร์ (Initial Parameter) ที่จำเป็นในการตรวจสอบความถูกต้องของ SOAP message ซึ่งค่าพารามิเตอร์เหล่านี้จะถูกกำหนดค่าโดยผู้ทดสอบระบบ เหตุผลที่ต้องให้ผู้ทดสอบระบบเป็นคนกำหนดค่า เนื่องจากผู้วิจัยมองว่าผู้ทดสอบระบบจะมีความรู้และเข้าใจระบบมากกว่าผู้ดูแลระบบที่อาจจะมีความรู้หรือเข้าใจในระบบไม่ดีพอ ซึ่งอาจมีผลทำให้การกำหนดค่าที่ไม่ครอบคลุม ที่อาจส่งผลให้การตรวจสอบความถูกต้องของ SOAP message เกิดช่องโหว่ได้ พัฒนาด้วยภาษา JAVA

การสร้างรายงาน (Generate Report) ส่วนนี้จะนำเอาล็อก ที่ถูกจัดเก็บไว้ในขบวนการตรวจสอบความถูกต้องของ SOAP message มาสร้างเป็นรายงานให้ผู้ดูแลระบบโดยจะมีการสร้างรายงานแบบ Real-time และตรงส่วนนี้ยังมีอัลกอริทึมที่จะนำเอาล็อกที่เกิดจากการร้องขอที่ถูกปฏิเสธเนื่องจากเอกสารเอกซ์เอ็มแอลสคีมา ที่มีไม่สอดคล้องกับรูปแบบของ SOAP message เพื่อเป็นการเรียนรู้รูปแบบของ SOAP message ในรูปแบบใหม่ๆ พัฒนาด้วยภาษา JAVA และใช้ Apache axis2 ในการจัดการกับ SOAP message

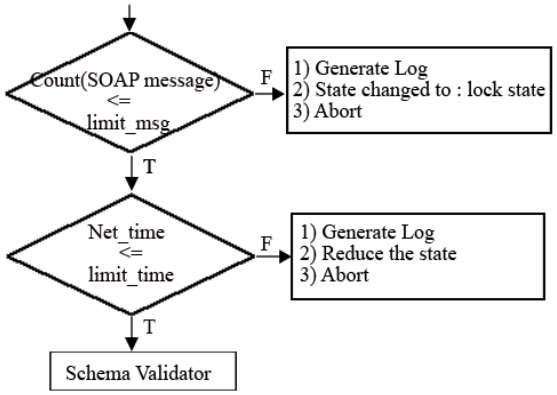
4.1. การทดลองการโจมตีแบบ Replay Attack

ในการทดลองสำหรับตรวจสอบการโจมตีแบบ Replay Attack การดำเนินการทดลองจะมีส่วนที่สำคัญคือ

1) เมื่อมีการร้องขอใช้บริการจะต้องเพิ่ม Timestamp ในส่วนของ Header ของ SOAP request

2) เมื่อ Message เดินทางมาถึง S-DV ก็จะจับ Message พร้อมนับจำนวน Message ที่ส่งมาจากต้นทางเดียวกันว่าในหนึ่งวินาทีส่งเกินจำนวนที่กำหนดไว้หรือไม่ เพื่อป้องกันการโจมตีแบบ SOAP flooding

3) สกัดเอา Timestamp มาคำนวณเทียบกับเวลาปัจจุบันว่าเกินช่วงชีวิตของ Message หรือไม่จาก รูปที่ 11 ในส่วนของ Net_time คือผลต่างของวันเวลาที่สร้าง Message กับวันเวลาปัจจุบัน ส่วน limit_Time คือช่วงเวลาชีวิตของ Message และ limit_msg คือจำนวน Message สูงสุดที่ยอมรับได้ต่อวินาที



รูปที่ 11. การตรวจจับการโจมตีแบบ Replay Attack

4.2. การทดลองการโจมตีแบบ Buffer Overflow

ในการทดสอบเพื่อตรวจจับการโจมตีแบบ Buffer Overflow นั้นจะใช้ความสามารถของเอกซ์เอ็มแอลสคีมา ในการตรวจจับ ดังตัวอย่างรูปที่ 12 ด้วยความสามารถของเอกซ์เอ็มแอลสคีมา เราสามารถกำหนดรูปแบบของโครงสร้างเอกสารเอกซ์เอ็มแอล ประเภทของข้อมูล รูปแบบข้อมูล ขอบเขตข้อมูล ให้กับแต่ละอิลิเมนต์หรือแอตทริบิวต์ ด้วยความสามารถนี้เราจึงสามารถนำมาตรวจสอบขอบเขตของข้อมูลเพื่อไม่ให้เกินขอบเขตที่กำหนดไว้ได้ ตัวอย่างใน รูปที่ 12 ใช้ในการตรวจสอบอิลิเมนต์ number ว่าเก็บข้อมูลแบบตัวเลขจำนวนเต็มและเก็บได้ต่ำสุดคือ -2147483648 สูงสุดคือ 2147483647 ถ้านอกเหนือจากนี้ถือว่าเป็นค่าไม่ถูกต้อง

```
<xs:element name="number">
 <xs:simpleType>
  <xs:restriction base="xs:int">
   <xs:minInclusive value="-2147483648" />
   <xs:maxInclusive value="2147483647"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

รูปที่ 12. แสดงเอกซ์เอ็มแอลสคีมาสำหรับกำหนดขอบเขตข้อมูล

4.3. การทดลองการโจมตีแบบ Parameter Tampering

สำหรับการทดลองเพื่อตรวจจับการโจมตีแบบ Parameter Tampering จะใช้คุณสมบัติของ regexp ที่มีลักษณะเป็นลำดับของอักขระที่ใช้จัดรูปแบบสำหรับค้นหาหรือจับคู่สายอักขระ เราสามารถนำเอา regexp มาจัดรูปแบบในการจับคู่สายอักขระที่เราสามารถอนุญาตให้มีได้หรือไม่อนุญาตให้มี ตัวอย่างเช่น รูปแบบอีเมลในส่วนของ regexp กำหนดได้คือ "^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z]+(\\.[A-Za-z]+)*(\\.[A-Za-z]{2})?$" หรือไม่อนุญาตให้มีอักขระที่กำหนดในสายอักขระ สามารถกำหนดรูปแบบของ regexp คือ "^[^<>&]*$" หมายถึงในสายอักขระไม่อนุญาตให้มีอักขระ "<" หรือ ">" หรือ "&"

4.4. การทดลองการโจมตีแบบ Recursive Payload

ในการทดสอบเพื่อตรวจจับการโจมตีแบบ Recursive Payload นี้ ก็ใช้ความสามารถของเอกซ์เอ็มแอลสคีมา ดังตัวอย่างรูปที่ 13 ในการตรวจสอบโครงสร้างเอกสารเอกซ์เอ็มแอลว่าถูกต้องตามที่กำหนดไว้หรือไม่ จากตัวอย่างด้านล่างนี้ แสดงให้เห็นการใช้เอกสารเอกซ์เอ็มแอลสคีมา มาควบคุมโครงสร้างของอิลิเมนต์ transaction ที่มีอิลิเมนต์ลูกคือ total และ credit_NO เท่านั้น โดยแต่ละอิลิเมนต์สามารถปรากฏเป็นอิลิเมนต์ลูกของ transaction ได้เพียงครั้งเดียวและไม่สามารถวางซ้อนทับกันได้

```
<xs:element name="transaction">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="total" type="xs:float" />
  <xs:element name="credit_NO" type="xs:string" />
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

รูปที่ 13. เอกซ์เอ็มแอลสคีมาควบคุมโครงสร้างเอกซ์เอ็มแอล

4.5. การทดลองการโจมตีแบบ Oversize Payload

ในการทดสอบเพื่อตรวจจับการโจมตีแบบ Oversize Payload จะใช้ความสามารถของเอกซ์เอ็มแอลสคีมา ร่วมกับการตรวจสอบขนาดของเอกซ์เอ็มแอลกับค่าพารามิเตอร์ที่กำหนดโดยผู้ทดสอบระบบในขั้นตอนการเรียนรู้รูปแบบเอกซ์เอ็มแอลจากตัวอย่างแสดงในรูปที่ 14 แสดงเอกสารเอกซ์เอ็มแอลสคีมาที่นำมาใช้ควบคุมโครงสร้างของเอกสารเอกซ์เอ็มแอลให้อยู่ในรูปแบบที่ต้องการ จากรูปที่ 14 ได้แสดงให้เห็นว่าเอกสาร

เอกซ์เอ็มแอลสามารถประกอบไปด้วย อิลิเมนต์ item สามารถปรากฏได้ 2 ครั้ง เพราะว่าได้มีการกำหนดแอตทริบิวต์ maxOccurs="2" คืออนุญาตให้อิลิเมนต์ item ปรากฏได้สูงสุดคือ 2 ครั้ง

```
……..
<xs:sequence>
<xs:element name="item"  type="xs:string"
maxOccurs="2"/>
</xs:sequence>
……..
```

รูปที่ 14. แสดงเอกซ์เอ็มแอลสคีมาสำหรับตรวจสอบ

4.6. ผลการทดลอง

เพื่อทดสอบกลไกในการป้องกันและตรวจจับการโจมตีต่างๆ ที่ใช้เอกซ์เอ็มแอล ได้มีการพัฒนาโมดูลต่างๆ ของระบบตรวจจับด้วย Apache Tomcat, Apache Axis2, JAVA และฐานข้อมูลใช้ MySQL ผลของการทดลองแสดงในตารางที่ 2 จากผลการทดลองในตารางนี้แสดงให้เห็นผลการทดลองที่ได้ทำการทดลองในการตรวจจับการโจมตีแบบต่างๆ โดยสรุปได้ดังนี้

ตารางที่ 2. ตารางแสดงผลลัพธ์จากการทดลอง

ทดลอง	ค่าที่ใช้ทดลอง	ค่าสูงสุดที่ระบบยอมรับได้	ผลการทดลอง
SOAP flooding	15 Message	10 Message/s	15 Message
Replay Attack	30/07/2557 16:9:19	5 Second	30/07/2557 16:9:20
Buffer Overflow	3.4028235E39	3.4028235E38	3.4028235E39
Parameter Tampering	<, &, >, #	-	<, &, >, #
Recursive Payload	3134 Byte	4000 Byte	3134 Byte
Oversize Payload	5809 Byte	5000 Byte	5809 Byte

SOAP flooding เป็นการทดลองการนับจำนวนของการร้องขอที่ส่งคำร้องขอเข้ามาโดยทดลองส่งการร้องขอไป 15 Message แต่ระบบรองรับได้สูงสุด 10 Message ต่อวินาที สำหรับการร้องขอจากผู้ใช้งานแต่ละคนผลจากการทดลองระบบสามารถตรวจจับได้ทั้งหมด 15 Message

Replay Attack เป็นการทดลองในการตรวจสอบช่วงเวลาชีวิตของ Message ในการทดลองได้ส่ง Message พร้อมเพิ่ม Timestamp ที่เป็นเวลาที่ Message ถูกสร้างขึ้นในเวลา 30/07/2557 16:9:19 โดยจะมีอายุได้ไม่เกิน 5 Second ตั้งแต่เวลาที่สร้าง Message จนถึงเวลาที่ได้รับ Message ผลของการทดลองสามารถจับ Message ได้ในเวลา 30/07/2557 16:9:20

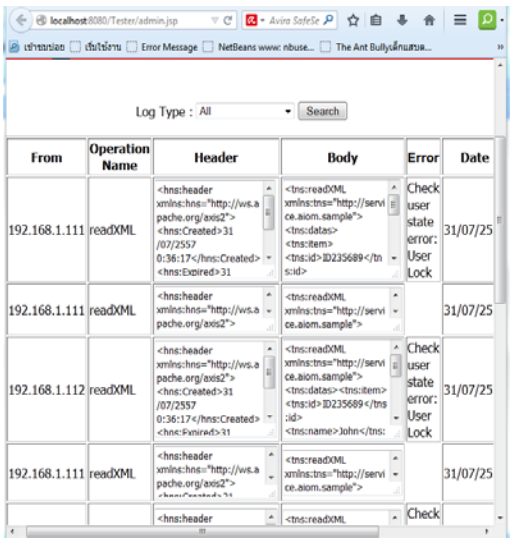
Buffer Overflow เป็นการทดลองในการตรวจสอบขอบเขตของข้อมูล โดยส่งข้อมูลที่เป็นเงินเดือนที่มีชนิดข้อมูลแบบ float มีค่า 3.4028235E39 แต่ขอบเขตสูงสุดของชนิดข้อมูลแบบ float คือ 3.4028235E38 และระบบสามารถตรวจจับได้ค่าที่ส่งมาได้คือ 3.4028235E39

Parameter Tampering เป็นการทดลองในการตรวจสอบหาอักขระ ที่ไม่อนุญาตให้มีในกรณีที่มีการใช้งานบล็อกของ CDATA ซึ่งอักขระที่ใช้ในการทดลองคือ <, &, >, # ผลจากการทดลองระบบสามารถตรวจจับได้คือ <, &, >, #

Recursive Payload เป็นการทดลองในการตรวจสอบเอกสารเอกซ์เอ็มแอลที่มีการวางอิลิเมนต์ซ้อนทับกันขนาด 3134 ไบต์ โดยที่ระบบยอมรับขนาดของเอกซ์เอ็มแอลได้สูงสุด 4000 ไบต์ ผลการทดลองระบบสามารถตรวจจับได้คือ 3134 ไบต์

Oversize Payload เป็นการทดลองในการตรวจสอบขนาดของเอกสารเอกซ์เอ็มแอลในการทดลองได้ส่งเอกสารเอกซ์เอ็มแอล ขนาด 5809 ไบต์ แต่ระบบสามารถยอมรับขนาดของเอกซ์เอ็มแอลได้สูงสุด 5000 ไบต์ ผลการทดลองระบบสามารถตรวจจับได้คือ 5809ไบต์

นอกจากนี้ระบบยังได้นำเอาล็อกต่างๆ ที่เกิดขึ้นในระบบมาสร้างเป็นรายงานให้กับผู้ดูแลระบบได้ทำการตรวจสอบว่าเกิดอะไรขึ้นบ้างในระบบ ดังแสดงใน รูปที่ 15



รูปที่ 15. แสดงรายงานสำหรับผู้ดูแลระบบ

จากรูปที่ 15 แสดงรายงานสำหรับผู้ดูแลระบบที่เกิดจากการนำเอาล็อกต่างๆ ที่เกิดขึ้นในระบบมาสร้างเป็นรายงานเพื่อให้ผู้ดูแลระบบสามารถที่จะทำการตรวจสอบได้ว่าเกิดอะไรขึ้นบ้างในระบบ โดยรายงานสำหรับผู้ดูแลระบบจะประกอบไปด้วย

- From ตรงส่วนนี้จะแสดง IP address ของผู้ร้องขอว่าการร้องขอใช้บริการนั้นส่งมาที่ไหน
- Operation Name จะเป็นส่วนของการแสดงชื่อของบริการว่าการร้องขอนี้เป็นการร้องขอใช้บริการอะไร
- Header จะเป็นส่วนของการแสดงเอกสารในส่วนของ Header ของเอกสารโซพ
- Body จะเป็นส่วนของการแสดงเอกสารในส่วนของ Body ของเอกสารโซพ
- Error จะเป็นส่วนข้อความข้อผิดพลาดที่เกิดขึ้นจากการตรวจสอบของระบบ
- Date จะเป็นส่วนของการแสดงวันที่ของการสร้างล็อก
- Time จะเป็นส่วนของการแสดงเวลาของการสร้างล็อก

5. สรุปผลและข้อเสนอแนะ

งานวิจัยนี้จะเป็นการนำเสนอกรอบการทำงานสำหรับป้องกันการโจมตีเว็บเซอร์วิสที่ใช้เอกซ์เอ็มแอล ซึ่งกรอบการทำงานที่นำเสนอนี้ จะมุ่งเน้นไปที่การตรวจสอบความถูกต้องของเอกซ์เอ็มแอลของทุก ๆ การร้องขอก่อนที่จะถูกส่งไปให้กับเว็บเซอร์วิสประมวลผล ดังนั้นข้อมูลที่ถูกต้องเท่านั้นที่จะได้รับอนุญาต เมื่อตรวจสอบข้อมูลแล้วไม่ถูกต้อง การร้องขอนั้นก็จะถูกยกเลิกไป สำหรับการตรวจสอบความถูกต้องของเอกซ์เอ็มแอลในกรอบการทำงานที่ได้นำเสนอนี้ จะใช้ความสามารถของเอกซ์เอ็มแอลสคีมา ในการตรวจสอบโครงสร้างเอกซ์เอ็มแอล ชนิดของของข้อมูล รูปแบบของข้อมูลและขอบเขตของข้อมูลที่จัดเก็บในแต่ละอิลิเมนต์หรือในแต่ละแอตทริบิวต์ของเอกสารเอกซ์เอ็มแอลร่วมกับค่าพารามิเตอร์ที่กำหนดโดยผู้ทดสอบระบบและ regexp สำหรับตรวจสอบข้อความหรืออักขระบางอย่างที่เราไม่อนุญาตให้มีในข้อมูลที่เป็นข้อความ

สำหรับปริมาณของการร้องขอที่อาจจะถูกยกเลิกหรือได้รับอนุญาตนั้น ขึ้นอยู่กับขั้นตอนการเรียนรู้รูปแบบของเอกซ์เอ็มแอลและการกำหนดค่าพารามิเตอร์ถ้าในขั้นตอนการเรียนรู้รูปแบบของเอกซ์เอ็มแอลและการกำหนดค่าพารามิเตอร์ครอบคลุมปริมาณการร้องขอที่จะถูกยกเลิกจะน้อยแต่ถ้าในขั้นตอนการเรียนรู้รูปแบบของเอกซ์เอ็มแอลและการกำหนดค่าพารามิเตอร์ไม่ครอบคลุมปริมาณการร้องขอที่จะถูกยกเลิกก็จะมากขึ้น

ผลจากการทดลองเบื้องต้นแสดงให้เห็นว่ากรอบการทำงานที่นำเสนอนี้มีความสามารถในการตรวจจับการโจมตีที่ใช้เอกซ์เอ็มแอลได้อย่างอย่างมีประสิทธิภาพและสามารถเพิ่มความต้านทานต่อความผิดพลาดของเว็บเซอร์วิสเพิ่มขึ้น

เอกสารอ้างอิง

[1] W3C Working Group, Web Services Architecture. [Online]. 2004. Available from: http://www.w3.org/TR/ws-arch/ [2014, July, 25]

[2] Doug Tidwell, James Snell and Pavel Kulchenko, Programming Web Services with SOAP First Edition. O'Reilly. Sebastopol. December 2001.

[3] Marzouk S. Mokbel and Le Jiajin, "Integrated Security Architecture for Web Services and this Challenging", Asian Journal of Information Technology, Volue 7, Issue 5, 2008. pp. 226-231.

[4] Vipul Patel, Radhesh Mohandas and Alwyn R. Pais, " Attacks on Web Services and mitigation schemes", IEEE Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference, July 2010. pp.1-6.

[5] David Hunter, Jeff Rafter, Joe Fawcett, Eric van der Vlist, Danny Ayers, Jon Duckett, Andrew Watt, and Linda McKinnon, Beginning XML 4th Edition. Wiley Publishing. Indiana. May, 2007.

[6] Poornachandra Sarang, Ph.D, Pro Apache XML First Edition. Apress. New York. May 2006.

[7] W3C, Simple Object Access Protocol 1.1. [Online]. 2000. Available from: http://www.w3.org /TR/2000/NOTE-SOAP-20000508/#_Toc478383486 [2014, July, 25]

[8] W3C, Web Services Description Language 1.1. [Online]. 2001. Available from: http://www.w3.org /TR/wsdl [2014, July, 25]

[9] OASIS, Universal Description, Discovery and Integration (UDDI) v2.0. [Online]. 2003. Available from: https://www.oasisopen.org/standards#uddiv2 [2014, July, 25]

[10] Priscilla Walmsley, Definitive XML Schema 2th Edition. PRENTICE HALL. New Jersey. September 2012.

[11] A. Karthigeyan, C. Andavar, A. Jaya Ramya, "Adaptable Practices for Curbing XDoS Attacks", International Journal of Scientific & Engineering Research, Volume 3, Issue 6, June 2012. pp.1-6.

[12] Esmiralda Moradian, and Anne Håkansson, "Possible attacks on XML Web Services", IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.1B, January 2006. pp.154-170.

[13] Abhinav Nath Gupta and Dr. P. Santhi Thilagam, "Attacks on Web Services Need to Secure XML on Web", Computer Science & Engineering: An International Journal, Vol. 3, No. 5, October 2013. pp.1-11.

[14] Irfan siddavatam and Jayant Gadge, "Comprehensive Test Mechanism to Detect Attack on Web Services", IEEE International Conference on Networking, December 2008. pp.1-6.

[15] Nils Gruschka, Norbert Luttenberger, "Protecting Web Services from DoS Attacks by SOAP Message Validation", IFIP International Federation for Information Processing Volume 201, 2006, May 2006. pp 171-182.

[16] Rafael Bosse Brinhosa, Carla Merkle Westphall, Carlos Becker Westphall, Daniel Ricardo dos Santos, Fabio Grezele, "A Validation Model of Data Input for Web Services", Twelfth International Conference on Networks, January 2013. pp.87-94.

[17] R. Bebawy, H. Sabry, S. El-Kassas, Y. Hanna, and Y. Youssef, "Nedgty: Web services firewall", Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference, July 2005. pp. 597–601.

[18] Haiping Xu, Abhinay Reddyreddy, and Daniel F. Fitch, "Defending Against XML-Based Attacks Using State-Based XML Firewall", JOURNAL OF COMPUTERS, VOL. 6, NO.11, November 2011. pp. 2395-2407.

[19] Nuno Antunes, Nuno Laranjeiro, Marco Vieira, Henrique Madeira,"Effective Detection of SQL/XPath Injection Vulnerabilities in Web Services", Services Computing, 2009. SCC '09, IEEE International Conference, September 2009. pp.260-267.

[20] Gökhan Muharremoğlu, Web Application Level Approach against the HTTP Flood Attacks IOSEC HTTP Anti Flood/DoS Security Gateway Module. [Online]. 2012. Available from: http://goo.gl/aQM4Di [2014, July ,25]

[21] Eric Chien and Péter Ször, Blended Attacks Exploits, Vulnerabilities and Buffer-Overflow Techniques in Computer Viruses*. [Online]. 2002. Available from: http://www.symantec.com/avcenter/reference/blended.attacks.pdf [2014, July ,25]

[22] Meiko Jensen, Christopher Meyer, Juraj Somorovsky, and Jorg Schwenk, "On the effectiveness of XML Schema validation for countering XML Signature Wrapping attacks", Securing Services on the Cloud (IWSSC), 2011 1st International Workshop on, September 2011. pp.7-13,6-8.

[23] Shujun Pei, Deyun Chen, Yuyuan Chu, Qingfeng Xu and Shi Xi,"Research of Web Service Security Model Based on SOAP Information", Asian Network for Scientific Information, December 2011. pp. 241-247.