บทความวิจัย (Research Article)

ระบบตรวจสอบและแจ้งเตือนสถานะและการเปลี่ยนแปลงเว็บไซต์แบบเรียลไทม์

Real-time monitoring and notification system for website status and defacement

กำธร สารวรรณ[1*] ภูริ จันทิมา[1] วรพจน์ สมมูล[1] รณชัย สังหมื่นเม้า[1] และ พรศิริ คำหล้า[2]

[1] สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยกาฬสินธุ์
[2] สาขาวิชาวิศวกรรมโลจิสติกส์และเทคโนโลยีขนส่ง คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยกาฬสินธุ์
* ผู้รับผิดชอบบทความ
kamthorn.sa@ksu.ac.th

Received: 31 Jan 2024
Revised: 30 Apr 2024
Accepted: 30 Apr 2024

บทคัดย่อ

ความสามารถในการให้บริการอย่างต่อเนื่องของเว็บไซต์เป็นหัวใจสำคัญขององค์กรในยุคปัจจุบัน หากเว็บไซต์ไม่สามารถให้บริการได้หรือถูกโจมตีด้วยการเปลี่ยนแปลงข้อมูลหน้าเว็บไซต์ย่อมส่งผลเสียหายทั้งทางตรงและทางอ้อม การเฝ้าระวังและแจ้งเตือนเมื่อเกิดเหตุดังกล่าวเป็นสิ่งสำคัญไม่น้อยกว่าป้องกันไม่ให้เกิดเหตุ สำหรับองค์กรที่มีศูนย์ข้อมูลเป็นของตนเองการออกแบบให้ระบบเฝ้าระวังติดตั้งภายในศูนย์ข้อมูลเอง จึงมีความไม่เหมาะสมมากนัก เพราะหากระบบการสื่อสารของศูนย์ข้อมูลมีปัญหา จะส่งผลให้ระบบเฝ้าระวังไม่สามารถทำงานได้ด้วยเช่นกัน งานวิจัยนี้ได้นำเสนอการประยุกต์ใช้เทคโนโลยีคลาวด์ ซึ่งเป็นการเฝ้าระวังจากระยะไกล อีกทั้งระบบที่นำเสนอสามารถใช้งานได้ฟรี เพื่อออกแบบและพัฒนาระบบตรวจสอบและแจ้งเตือนเมื่อเว็บไซต์ไม่สามารถให้บริการด้วยการตรวจสอบจากรหัส HTTP Status หรือตรวจสอบการถูกโจมตีการเปลี่ยนแปลงข้อมูลเว็บไซต์ด้วยเทคนิคการ Hashing ซึ่งระบบที่นำเสนอสามารถบรรลุวัตถุประสงค์ กล่าวคือระบบสามารถส่งข้อความแจ้งเตือนไปยังแอปพลิเคชั่นไลน์ อีเมล์ และทำการบันทึกข้อมูลได้ 100% ทั้งในกรณีเว็บไซต์ไม่สามารถให้บริการได้และกรณีถูกโจมตีด้วยการเปลี่ยนแปลงข้อมูล

คำสำคัญ: ระบบตรวจสอบและแจ้งเตือน สถานะเว็บไซต์ การโจมตีด้วยการเปลี่ยนแปลงข้อมูลเว็บไซต์

Abstract

In contemporary digital enterprises, the uninterrupted functionality of a website is pivotal. Disruptions, whether due to operational failures or targeted cyber-attacks such as web defacement, can inflict substantial direct and indirect ramifications. This study underscores the paramount importance of monitoring and instantaneous notification mechanisms, which are as crucial as preventive measures against such adversities. Traditional monitoring systems in data centers face limitations. Because the monitoring system won't function as well if there is an issue with the data center's communication system. This research advocates for the utilization of cloud-based technologies to implement remote monitoring solutions. The system proposed herein is notable for its cost-effectiveness, being free to use. It focuses on the development and deployment of a monitoring and alert mechanism. This system diligently checks the HTTP status codes to ascertain service availability and employs hashing techniques to detect instances of

web defacement. The efficacy of the system is demonstrated by its ability to seamlessly relay notification messages through diverse channels, including the LINE application and email, ensuring robust data logging capabilities. The system has proven its merit by maintaining a 100% success rate in sending alerts in scenarios where the website either fails to render services or falls victim to web defacement.

Keywords: Kubernetes, pod control, HPA, horizontal pod autoscaler

1. บทนำ

ยุคของการแข่งขันด้วยเทคโนโลยีอินเทอร์เน็ตและดิจิทัล ความพร้อมในการให้บริการเว็บไซต์ขององค์กรมีความสำคัญอย่างยิ่ง [1-2] โดยเฉพาะเว็บไซต์ที่ให้บริการด้านอีคอมเมิร์ช ซึ่งส่งผลสัมพันธ์โดยตรงกับการสร้างรายได้ เพราะหากไม่สามารถให้บริการได้ย่อมสูญเสียโอกาสในการขายสินค้าอย่างแน่นอน หรือแม้แต่องค์กรด้านการศึกษา หรือภาครัฐ ในปัจจุบันต่างมีสินค้าและบริการที่ให้บริการแก่นักศึกษา ประชาชน หรือผู้ใช้บริการในรูปแบบออนไลน์มากยิ่งขึ้น รวมถึงบางองค์กรมีการใช้เป็นแพลตฟอร์มหลักสำหรับฟังก์ชันการทำงานภายในองค์กรอีกด้วย เช่นเดียวกันหากระบบไม่สามารถให้บริการได้ การดำเนินกิจกรรมภายในองค์กรย่อมเกิดความล่าช้า นอกเหนือไปกว่านั้นแล้วยังส่งผลทางอ้อมต่อความน่าเชื่อถือของผู้รับบริการต่อองค์กรด้วย ดังนั้นการหยุดให้บริการของเว็บไซต์หรือการถูกโจมตีเพื่อเปลี่ยนแปลงข้อมูลหน้าเว็บเพจ (web defacement) ย่อมส่งผลต่อภาพลักษณ์และชื่อเสียง ในทางตรงกันข้ามหากเว็บไซต์สามารถให้บริการได้ตลอดเวลาย่อมส่งผลต่อภาพลักษณ์ที่ดีขององค์กร ด้วยเหตุผลนี้การป้องกันและแก้ไขไม่ให้เว็บไซต์หยุดให้บริการจึงเป็นสิ่งสำคัญมาก

โดยทั่วไปของการให้บริการของผู้ให้บริการโครงข่ายอินเทอร์เน็ต (internet service provider: ISP) และเว็บเซอร์เวอร์ (web server) จะมีการทำข้อตกลงระดับการให้บริการระหว่าง “ผู้ให้บริการ” และ “ลูกค้า” หรือ Service Level Agreement (SLA) [3] ซึ่งโดยปกติจะกำหนดค่าเป้าหมายให้สามารถบริการได้ (uptime) เท่ากับ 100% แต่ในความเป็นจริงหากให้บริการได้ 99.99% [4] ถือได้ว่ามีความสามารถในการให้บริการสูง อย่างไรก็ตาม การที่จะทำให้เว็บไซต์สามารถบริการได้ถึง 99.99% เป็นเรื่องที่ท้าทาย นอกจากการป้องกันไม่ให้เกิดเหตุที่ส่งผลต่อการเกิดเวลาหยุดทำงาน (downtime) แล้วนั้น การเฝ้าระวังด้วยระบบตรวจสอบที่มีประสิทธิภาพและสามารถแจ้งเตือนไปยังผู้ดูแลระบบให้สามารถแก้ไขปัญหาได้อย่างรวดเร็วก็เป็นอีกแนวทางหนึ่ง ถึงแม้ว่าในปัจจุบันมีระบบในลักษณะการตรวจสอบและแจ้งเตือนจำนวนมาก แต่หลายระบบยังมีข้อจำกัดในการนำไปใช้งาน ไม่ว่าจะเป็นราคาที่สูง การปรับแต่งให้เหมาะสมกับองค์กร ความต้องการของทรัพยากรฮาร์ดแวร์ในการติดตั้งซอฟต์แวร์ ซึ่งนั้นอาจจะไม่ใช่ประเด็นปัญหาหากองค์กรนั้นเลือกใช้ศูนย์ข้อมูล (data center) จากบริษัทเอกชนที่มีการทำข้อตกลงในการให้บริการอย่างเป็นระบบ แต่สำหรับองค์กรที่มีศูนย์ข้อมูลตนเองและมีการดำเนินงานด้วยตนเองจึงเป็นเรื่องที่ท้าทายสำหรับการสร้างระบบตรวจสอบและแจ้งเตือน หรือในกรณีใช้ซอฟต์แวร์แบบโอเพนซอร์สจำเป็นต้องมีผู้เชี่ยวชาญในการติดตั้งและจัดการซอฟต์แวร์ และโดยปกติการติดตั้งซอฟต์แวร์สำหรับระบบตรวจสอบนี้มักจะติดตั้งบนเครื่องแม่ข่ายที่อยู่ภายในศูนย์ข้อมูลเดียวกันกับเว็บเซิร์ฟเวอร์ที่ต้องการเฝ้าระวัง ทำให้เมื่อเกิดเหตุการณ์ที่ทำให้การสื่อสารขัดข้องทั้งภาพรวมของศูนย์ข้อมูลไม่ว่าจะด้วยเหตุใดก็ตาม จะทำให้ระบบแจ้งเตือนไม่สามารถทำงานได้ เพราะระบบแจ้งเตือนถูกตัดขาดการสื่อสารเช่นกัน ด้วยเหตุนี้จึงเป็นข้อจำกัดของการออกแบบระบบการตรวจสอบที่ไม่เหมาะสมมากนัก

การศึกษาครั้งนี้ได้ทำการศึกษาข้อจำกัดและปัญหาของการเฝ้าระวังการให้บริการเว็บไซต์ โดยมุ่งเน้นไปที่กรณีที่องค์กรมีศูนย์ข้อมูลตนเอง เช่น สถาบันอุดมศึกษา เป็นต้น เพื่อนำเสนอระบบตรวจสอบและแจ้งเตือนสถานะเว็บเซิร์ฟเวอร์แบบเรียลไทม์ ด้วยการประยุกต์ใช้บริการประมวลผลแบบคลาวด์ของกูเกิล (google cloud platform) ประกอบด้วย Apps Script, Sheets และ Looker Studio นอกจากนั้นระบบที่นำเสนอสามารถแจ้งเตือนข้อมูลแบบเรียลไทม์ (real time) ไปยังผู้ดูแลระบบที่เกี่ยวข้องได้ ซึ่งองค์ประกอบภาพรวมระบบที่ออกแบบและนำเสนอสามารถใช้งานได้ฟรี และยังมีความเหมาะสมสำหรับการตรวจสอบศูนย์ข้อมูลองค์กรจากระยะไกลด้วยหลักการระบบอัตโนมัติบนคลาวด์ (cloud-based automation)

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงเกี่ยวกับเทคโนโลยี ทฤษฎี และงานวิจัยที่เกี่ยวข้องกับการศึกษานี้ ดังหัวข้อต่อไปนี้

2.1 การโจมตีเพื่อเปลี่ยนข้อมูลหน้าเว็บไซต์

การโจมตีด้วยเทคนิคบางอย่างหรือหลายเทคนิครวมกัน เช่น SQL injection, Cross-site scripting (XSS) หรือ Remote file inclusion เป็นต้น [5] เพื่อเปลี่ยนข้อมูลหน้าเว็บไซต์ หรือเรียกว่า Web Defacement [6-7] เป็นการโจมตีทางไซเบอร์ที่ผู้โจมตีแฮ็กเข้าไปในเว็บไซต์และเปลี่ยนแปลงหน้าเว็บเพื่อแสดงเนื้อหาที่ผู้โจมตีต้องการ หรือการแก้ไขเนื้อหาใหม่ ซึ่งอาจจะเป็นข้อความ รูปภาพ หรือวิดีโอ การโจมตีประเภทนี้มักจะมีจุดประสงค์เพื่อสร้างความสับสน ประท้วง หรือเพื่อเป็นการแสดงความสามารถของผู้โจมตี โดยจากรายงานของ Trend Micro [8] เกิดเหตุการณ์นี้มากถึง 13 ล้านครั้งในช่วงปี 1998 ถึง 2016 และยังติด 10 อันดับแรกของภัยคุกคามเว็บไซต์อีกด้วย [9] ซึ่งปัจจุบันยังพบเหตุการณ์นี้เกิดขึ้นบ่อย ๆ โดยเฉพาะหน่วยงานราชการของประเทศไทย [10-11] ตัวอย่างเว็บไซต์ที่ถูกโจมตีด้วยวิธีการนี้แสดงดังรูปที่ 1

รูปที่ 1 เว็บไซต์ที่ถูกโจมตีด้วยวิธี Web Defacement [8]

2.2 บริการประมวลผลแบบคลาวด์

บริการประมวลผลแบบคลาวด์ (Cloud Computing) [12] คือการให้บริการทางด้านเทคโนโลยีสารสนเทศที่อาศัยทรัพยากรคอมพิวเตอร์และซอฟต์แวร์ที่อยู่บนเครือข่ายคลาวด์หรืออินเทอร์เน็ต แทนที่จะใช้ระบบคอมพิวเตอร์ส่วนตัวหรือเซิร์ฟเวอร์ในสถานที่ (on-premise servers) การประมวลผลแบบคลาวด์ช่วยให้ผู้ใช้และองค์กรสามารถเข้าถึงข้อมูลซอฟต์แวร์ และทรัพยากรการประมวลผลอื่น ๆ ได้ทุกที่ทุกเวลาผ่านทางอินเทอร์เน็ต

Google Apps Script [13] เป็นแพลตฟอร์มสคริปต์ที่พัฒนาโดย Google สำหรับการสร้างแอปพลิเคชั่นของ Google และบริการของบุคคลที่สาม โดยเป็นภาษาสคริปต์ที่ฝังอยู่บนคลาวด์ซึ่งสร้างขึ้นบน JavaScript และสามารถใช้เพื่ออัตโนมัติหรือขยายความสามารถของแอปพลิเคชั่นใน G Suite เช่น Sheets, Docs, Forms, Drive และ Calendar เป็นต้น

2.3 การแจ้งเตือนด้วยแอปพลิเคชั่นไลน์

LINE Notify [14] เป็นบริการที่เปิดให้ใช้งานโดยแอปพลิเคชั่นไลน์ ซึ่งเป็นแพลตฟอร์มสื่อสารยอดนิยมในหลายประเทศทั่วโลก ซึ่ง LINE Notify บริการที่ช่วยให้ผู้ใช้สามารถส่งข้อความไปยัง LINE จากแอปพลิเคชั่นหรือบริการอื่นผ่านส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interfaces: API) โดยนักพัฒนาสามารถประยุกต์ใช้สำหรับระบบแจ้งเตือนอัตโนมัติตามเงื่อนไขของซอฟต์แวร์ที่ออกแบบ [15]

2.4 งานวิจัยที่เกี่ยวข้อง

วัลภา พุทธางกูร และคณะ [16] นำเสนอระบบสารสนเทศเพื่อการบริหารจัดการการเฝ้าระวังเครื่องแม่ข่ายโดยระบบสารสนเทศทางภูมิศาสตร์ ในรูปแบบเว็ปแอปพลิเคชั่น โดยใช้ภาษา PHP และฐานข้อมูล MySQL ซึ่งมีการประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์ช่วยในการวิเคราะห์และแสดงผลในการนำเสนอข้อมูล ระบบที่นำเสนอสามารถแจ้งเตือนไปยัง Smart Phone ด้วยบริการของ Pushover

เชฐ ศรีแย้ม [17] นำเสนอการพัฒนาระบบติดตามสถานะเครือข่ายคอมพิวเตอร์ พัฒนาด้วย PHP และอ่านค่าแบบเรียลไทม์ด้วย SNMP โปรโตคอล ซึ่งสามารถทำการตรวจสอบได้ทั้งเครื่องคอมพิวเตอร์แม่ข่ายและอุปกรณ์เครือข่าย โดยทำการประเมินระบบที่นำเสนอด้วยผู้เชี่ยวชาญ

Damrongsak Arunyagool และ คณะ [15] นำเสนอระบบสำหรับการควบคุมและตรวจสอบเครื่องใช้ไฟฟ้าภายในบ้านจากระยะไกลโดยใช้บัญชีกูเกิล ด้วยการแจ้งเตือนทาง Line โดยใช้ความช่วยเหลือของตัวรับส่งสัญญาณไร้สาย ระบบจะวัดกระแสและแรงดันไฟฟ้าโดยไม่ต้องสัมผัสโดยตรงกับวงจรไฟฟ้าแรงสูง ส่งข้อมูลไปยัง Google sheet เพื่อจัดเก็บ และ

แสดงค่าแบบเรียลไทม์บนแดชบอร์ด อีกทั้งยังส่งการแจ้งเตือนผ่านแอพพลิเคชั่น LINE กรณีกระแสไฟหรือกระแสไฟผิดปกติ ผลการทดลองแสดงความแม่นยำ 95.76% ในการวัดกำลัง

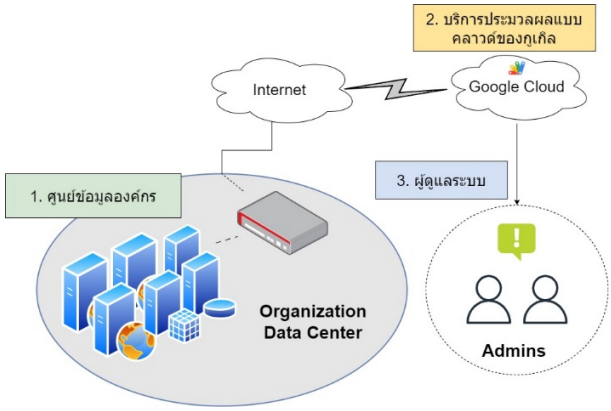
Giuseppe Aceto [18] นำเสนอการสำรวจระบบเฝ้าระวังเครือข่ายคอมพิวเตอร์ผ่านระบบคลาวด์จำนวน 28 บริการในด้านของประสิทธิภาพการทำงาน ปัญหา และข้อจำกัดต่าง ๆ โดยแบ่งเป็น 2 มิติ คือ มิติของคุณสมบัติเชิงพาณิชย์ และบริการตรวจสอบความน่าเชื่อถือ ซึ่งการนำระบบเฝ้าระวังเครือข่ายคอมพิวเตอร์ผ่านระบบคลาวด์มาใช้งานเป็นประโยชน์ในแง่ของความประหยัดด้านการซื้อฮาร์ดแวร์ แต่ยังมีข้อจำกัดอื่น ๆ ในด้านประสิทธิภาพและความครอบคลุมของการตรวจสอบในบางบริการ

Mariam Albalawi และคณะ [6] นำเสนอบทความเปรียบเทียบวิธีการตรวจสอบและเฝ้าระวังจากโจมตีด้วยวิธีการเปลี่ยนข้อมูลเว็บไซต์ โดยเปรียบเทียบผลลัพธ์ความแม่นยำและเทคนิคที่ใช้ในงานก่อนหน้าที่เกี่ยวข้องกับการเปลี่ยนแปลงข้อมูลเว็บไซต์ โดยมุ่งเน้นที่การบรรลุความแม่นยำในการตรวจจับสูงและลดการแจ้งเตือนที่ผิดพลาดให้เหลือน้อยที่สุด ซึ่งพบว่ามีการใช้เทคนิคและเครื่องมือการตรวจจับหลายอย่างเพื่อตรวจจับและติดตามการโจมตีที่ทำให้เว็บไซต์เสียหาย รวมถึง เทคนิคการเปรียบเทียบและจำแนกประเภท และอัลกอริธึมการเรียนรู้ของเครื่อง ซึ่งวิธีการผสมผสานระหว่างเทคนิคการเรียนรู้ของเครื่องและการใช้ลายเซ็นต์ พบว่ามีประสิทธิภาพสูงสุดในแง่ของอัตราความแม่นยำและอัตราผลบวกลวงต่ำ

3. วิธีดำเนินการวิจัย

3.1 ภาพรวมของระบบ

ภาพรวมของระบบประกอบไปด้วย 3 ส่วน ได้แก่ 1) ศูนย์ข้อมูลองค์กร ซึ่งเป็นพื้นที่ติดตั้งเครื่องเว็บเซิร์ฟเวอร์ขององค์กร สำหรับใช้เป็นกรณีศึกษาในการทดสอบ 2) บริการประมวลผลแบบคลาวด์ของกูเกิล ซึ่งจะประกอบด้วยบริการต่าง ๆ โดยในการศึกษานี้ใช้บริการ Apps Script Sheets และ Looker Studio สำหรับเป็นโปรแกรมในการตรวจสอบสถานะเครื่องเว็บเซิร์ฟเวอร์ขององค์กร เก็บข้อมูล และแสดงผลข้อมูลตามลำดับ และ 3) ผู้ดูแลระบบ หมายถึง เจ้าหน้าที่ที่มีหน้าที่ในการเฝ้าระวังการทำงานของเครื่องเว็บเซิร์ฟเวอร์ในองค์กร โดยใช้แอปพลิเคชั่นไลน์ในการรับการแจ้งเตือนแบบเรียลไทม์ และสามารถดูรายงานผ่าน Looker Studio ซึ่งภาพรวมการทำงานของระบบแสดง ดังรูปที่ 2



รูปที่ 2 ภาพรวมของระบบ

3.2 ขั้นตอนวิธีการตรวจสอบ

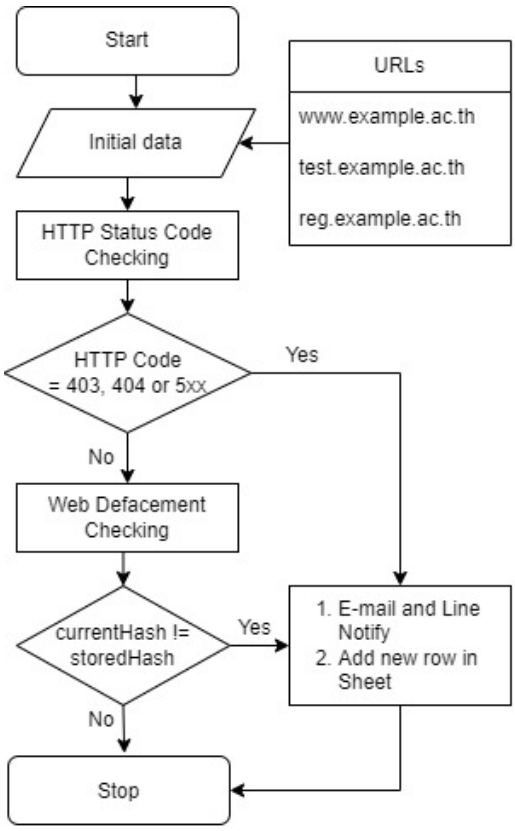
การตรวจสอบสถานะของเว็บเซิร์ฟเวอร์ในการศึกษานี้ออกแบบโดยมีแนวคิดหลัก คือ ใช้ระบบตรวจสอบจากระบบคลาวด์และเป็นบริการฟรี เนื่องจากการตรวจสอบจากคลาวด์มีจุดเด่นที่แยกพื้นที่ตั้งระหว่างเครื่องในองค์กรที่ตรวจสอบ กล่าวคือ ระบบตรวจสอบจากคลาวด์ไม่ได้ตั้งอยู่ในศูนย์ข้อมูลขององค์กร ทำให้ระบบตรวจสอบยังสามารถทำงานได้หากศูนย์ข้อมูลมีปัญหาในการสื่อสาร รวมทั้งระบบที่นำเสนอยังสามารถใช้งานได้ฟรี จึงมีความเหมาะสมสำหรับการประยุกต์ใช้สำหรับองค์กรที่ต้องการประหยัดงบประมาณ ดังนั้นผู้วิจัยจึงนำเสนอการประยุกต์ใช้ Apps Script Sheets และ Looker Studio รวมถึงแอปพลิเคชั่นไลน์

จากรูปที่ 3 แสดงขั้นตอนการตรวจสอบสถานะเว็บเซิร์ฟเวอร์ ซึ่งสามารถอธิบายการทำงานแต่ละขั้นตอนได้ดังนี้

Initial data คือ ขั้นตอนการเตรียมข้อมูล โดยการกำหนดโดเมนเนมหรือ URL ที่อยู่ของเว็บไซต์ที่ต้องการจะทำการตรวจสอบสถานะ

HTTP Status Code Checking คือ ขั้นตอนการตรวจสอบสถานะเว็บเซิร์ฟเวอร์โดยใช้การทำงานของฟังก์ชัน getResponseCode ของ Apps Script ด้วยการอ่านค่า URL

ของเว็บไซต์แต่ละ URL ซึ่งฟังก์ชันนี้จะตอบกลับมาเป็นรหัสตัวเลข โดยหากรหัสตอบกลับมาเป็นรหัส 403 404 หรือกลุ่มรหัส 500 ซึ่งหมายถึงเว็บไม่สามารถให้บริการได้ [19] ให้เรียกใช้ฟังก์ชันส่งแจ้งเตือนไปยังอีเมล์และแอปพลิเคชั่นไลน์กลุ่มของผู้ดูแลระบบที่ทำการตั้งค่าไว้ รวมถึงเรียกการทำงานฟังก์ชันการเพิ่มแถวใน Sheet แต่หากไม่ใช่รหัส 403 404 หรือกลุ่มรหัส 500 ให้จบการทำงาน เหตุผลที่เลือกตรวจสอบรหัส 403 และ 404 ด้วย ถึงแม้จะเป็นรหัสความผิดพลาดจากฝั่งลูกข่าย แต่อย่างไรก็ตามบางเหตุการณ์อาจจะเกิดขึ้นจากฝั่งเซิร์ฟเวอร์ เช่น กรณีไฟล์สูญหาย (404) หรือ ถูกปิดกั้นการเข้าถึงไฟล์ (403) โดยในกรณีศึกษานี้ลูกข่าย คือ กูเกิลคลาวด์ ซึ่งได้รับสิทธิ์เข้าถึงเว็บไซต์ที่ทำการทดสอบ



รูปที่ 3 ขั้นตอนภาพรวมการตรวจสอบเว็บเซิร์ฟเวอร์

Web Defacement Checking คือ ขั้นตอนการตรวจสอบการถูกโจมตีด้วยการเปลี่ยนหน้าเว็บไซต์ โดยขั้นตอนนี้ได้ประยุกต์ใช้ฟังก์ชันแฮช (hashing function) เพื่อทำการเปรียบเทียบหน้าเว็บก่อนการเปลี่ยนแปลง และหลังถูกเปลี่ยนแปลง เนื่องจากการใช้วิธีแฮชเป็นวิธีที่ง่ายและทำงานได้เร็ว [6] ซึ่งทำการอ่าน URL เว็บไซต์ที่อยู่ในรายการทีละค่า จากนั้นแปลง HTML เป็น Text แล้วทำการเข้าแฮชซิ่งด้วย MD5 จากนั้นทำการเก็บค่าแฮชไว้ด้วยบริการ PropertiesService เพื่อใช้สำหรับเปรียบเทียบค่าแฮช ซึ่งหากค่าแฮชที่เก็บไว้ไม่ตรงกับค่าแฮชใหม่แสดงว่าเว็บไซต์มีการเปลี่ยนแปลงข้อมูลหน้าเว็บเพจนั้น หลักการทำงานของขั้นตอนนี้ แสดงดังรูปที่ 4

```
var scriptProperties = PropertiesService.getScriptProperties();
var response =  UrlFetchApp.fetch(webksu[i].trim());
var content = response.getContentText();
var currentHash = Utilities.computeDigest(Utilities.DigestAlgorithm.MD5, content);
var currentHashString = currentHash.reduce(function(str, chr){
  chr = (chr < 0 ? chr + 256 : chr).toString(16);
  return str + (chr.length==1?'0':'') + chr;
},'');
var storedHashKey = 'storedHash_' + webksu[i].trim();
var storedHash = scriptProperties.getProperty(storedHashKey);
if (currentHashString !== storedHash) {
  scriptProperties.setProperty(storedHashKey, currentHashString);
  MailApp.sendEmail('kamthorn.sa@ksu.ac.th',
  'Web Page Changed',
  'The web page at ' + webksu[i].trim() + ' มีการเปลี่ยนแปลงเนื้อหาในเว็บไซต์');
  message += ""+ url + " : มีการเปลี่ยนแปลงเนื้อหาในเว็บไซต์" ;
  sendMessage(message);
  addNewRowToSpecificSheet(webksu[i], "Page Changed")
}
```

รูปที่ 4 วิธีตรวจสอบการถูกโจมตีด้วยการเปลี่ยนหน้าเว็บไซต์

การศึกษานี้ได้ทำการตั้งค่าทริกเกอร์ (trigger) [20] เพื่อให้โปรแกรมทำงานตามฟังก์ชันอัตโนมัติ โดยกำหนดให้ทำงานทุก 15 นาที

3.3 วิธีวัดประสิทธิภาพการทำงาน

การศึกษาครั้งนี้ได้ออกแบบการวัดประสิทธิภาพการทำงานของระบบที่นำเสนอ เพื่อทดสอบการตรวจสอบเว็บไซต์เป้าหมายในสถานการณ์ที่จำลองขึ้น ซึ่งในส่วนของการทดสอบการตรวจสอบสถานะจากรหัส HTTP Status ได้ทำการสร้างเว็บเพจที่ส่งค่ารหัส HTTP Status ประกอบด้วยรหัส 403 500 501 502 503 504 และรหัส 404 จะทำการกำหนด URL ที่ไม่มีอยู่จริง และส่วนของการตรวจสอบการโจมตีการเปลี่ยนเนื้อหาเว็บไซต์ได้ทำการทดสอบด้วยการแก้ไขข้อมูลก่อนการทดสอบ โดยทั้งหมดได้ทำการทดสอบทั้งหมด 5 ครั้ง ด้วยการกดรันสคริปโดยตรง ซึ่งผลที่คาดหวังคือระบบสามารถแจ้งเตือนข้อความไปยังแอปพลิเคชั่นไลน์ได้ ส่งอีเมล์ได้ และเพิ่มรายการไปยัง Sheet ได้ นอกจากนั้นแล้วยังทดสอบระบบการทำงานด้วย Trigger โดยทดสอบเป็นระยะเวลา 7 วัน

4. ผลการวิจัยและข้อเสนอแนะ

4.1 ผลของประสิทธิภาพระบบตรวจสอบและแจ้งเตือน

ผลการทดสอบประสิทธิภาพของระบบที่นำเสนอ ซึ่งทำการทดสอบในกรณีเว็บไซต์เกิดข้อผิดพลาดในกรณีรหัส 403 404 500 501 502 503 และ 504 ระบบสามารถส่งข้อความไปยังแอปพลิเคชั่นไลน์ได้ 100% ส่งอีเมล์ได้ 100% และทำการเพิ่มรายการใน Sheet ได้ 100% และการทดสอบการโจมตีด้วยการเปลี่ยนข้อมูลในเว็บเพจระบบสามารถส่งข้อความไปยังแอปพลิเคชั่นไลน์ได้ 100% ส่งอีเมล์ได้ 100% และทำการเพิ่มรายการใน Sheet ได้ 100% เช่นเดียวกัน

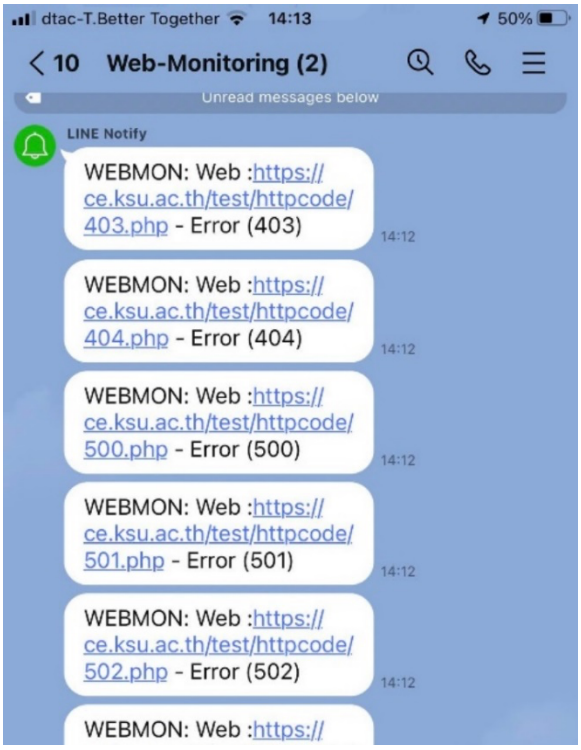
ตารางที่ 1 ผลของประสิทธิภาพระบบตรวจสอบและแจ้งเตือน

การทดสอบ	จำนวน	Line	Email	Sheet	Accuracy
403	5	5	5	5	100%
404	5	5	5	5	100%
500	5	5	5	5	100%
501	5	5	5	5	100%
502	5	5	5	5	100%
503	5	5	5	5	100%
504	5	5	5	5	100%
Changed	5	5	5	5	100%

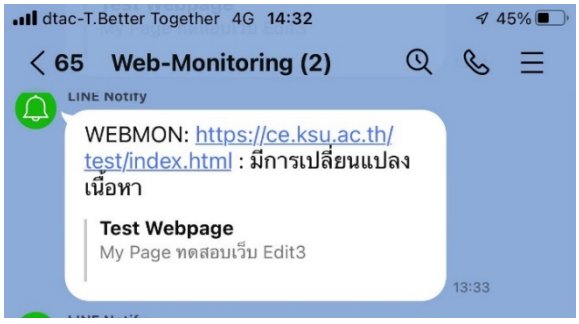
ตารางที่ 2 ผลของเวลาที่ใช้ในการแจ้งเตือนไปยังไลน์

การทดสอบ	จำนวน	เวลาเฉลี่ย (วินาที)	SD
403	5	2.80	0.45
404	5	2.60	0.55
500	5	2.80	0.45
501	5	2.60	0.55
502	5	2.60	0.55
503	5	2.40	0.55
504	5	2.60	0.55
Changed	5	3.60	0.55

ผลการทดสอบด้านเวลาที่ใช้ในการแจ้งเตือน โดยการทดลองนี้ผู้วิจัยได้ทำการทดสอบด้วยการเรียกใช้ App script แบบ manual โดยไม่ผ่านการใช้ทริกเกอร์เพื่อวัดเวลาที่ใช้ในการแจ้งเตือน โดยวัดจากผลต่างเวลาเริ่มรันสคริปต์กับเวลาที่ได้รับข้อความบนแอปพลิเคชั่นไลน์ ซึ่งทำการทดสอบจำนวน 5 ครั้ง พบว่าเวลาที่ใช้ระหว่างเริ่มรันสคริปต์กับเวลาที่ได้รับข้อความใช้เวลาอยู่ระหว่าง 2 – 4 วินาที รายละเอียดดังตารางที่ 2



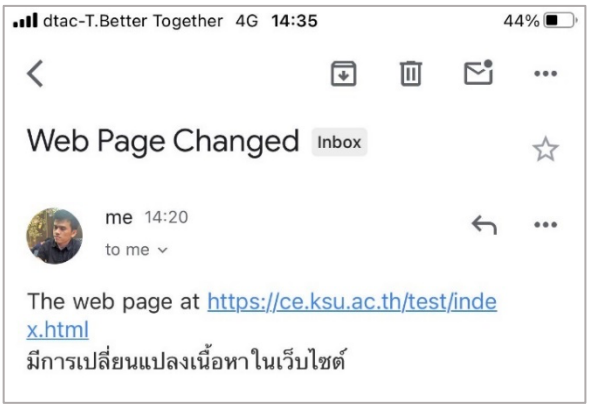
(ก)



(ข)

รูปที่ 5 ผลการแจ้งเตือนไปยังแอปพลิเคชั่นไลน์ (ก) HTTP Status Error (ข) Page Changed

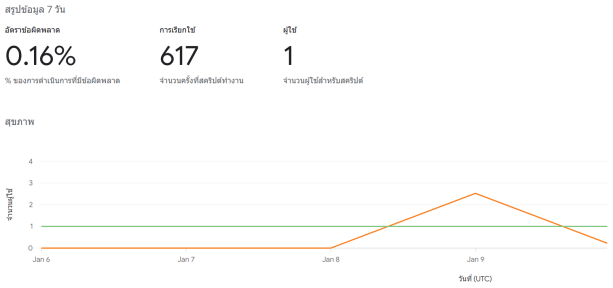
ผลการแจ้งเตือนไปยังอีเมล์ด้วยเมธอด sendEmail ของ Apps Script แสดงดังรูปที่ 6



รูปที่ 6 ผลการแจ้งข้อความไปยังอีเมล์

4.2 ผลการการทำงานของ Trigger

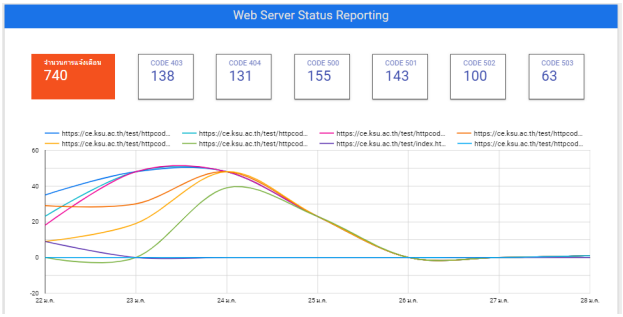
ประสิทธิภาพการทำงานของ Trigger ของ Apps Script ซึ่งใช้เรียกการทำงานระบบตรวจสอบและแจ้งเตือน ผลการทำงานของ Trigger ในระยะเวลา 7 วัน เกิดความผิดพลาด 0.16% ดังรูปที่ 7



รูปที่ 7 ผลการทำงานของ Trigger

4.3 ผลการแสดงรายงานด้วย Looker Studio

การศึกษานี้ได้ทำการประยุกต์ใช้บริการ Looker Studio เพื่อสร้างหน้าสรุปข้อมูล (dashboard) โดยมีแหล่งข้อมูลจาก Sheet ที่ทำการบันทึกข้อมูลทุกครั้งเมื่อเกิดข้อผิดพลาดของเว็บไซต์ขึ้น ซึ่งจากรูปที่ 8 เป็นผลการแสดงหน้ารายงานประวัติข้อผิดพลาดของเว็บไซต์กลุ่มตัวอย่างที่ทำการทดลอง โดยหน้าระบบรายงานจะแสดงสถิติจำนวนที่เกิดข้อผิดพลาดในแต่รหัส รวมถึงช่วงของวันที่ที่เกิดข้อผิดพลาดของแต่ละเว็บไซต์



รูปที่ 8 แสดงหน้าระบบรายงานข้อมูลด้วย Looker Studio

4.4 ข้อเสนอแนะ

การศึกษานี้มุ่งเสนอการใช้เทคโนโลยีคลาวน์เพื่อการตรวจสอบและแจ้งเตือนเมื่อเว็บไซต์เกิดปัญหา โดยการใช้เทคนิคการอ่านค่ารหัส HTTP Status และการตรวจสอบการเปลี่ยนแปลงข้อมูลของเว็บเพจโดยใช้เทคนิค hashing ของโค้ดเว็บไซต์ การใช้เทคนิค hashing เป็นวิธีการที่มีความเรียบง่าย ใช้งานฟรี และมีประสิทธิภาพสำหรับการตรวจสอบเว็บไซต์แบบคงที่ (Static Website) [6] อย่างไรก็ตาม หากเป็นเว็บไซต์แบบพลวัต (Dynamic Website) การใช้เทคนิค hashing อาจไม่เหมาะสม เนื่องจากมีการเปลี่ยนแปลงข้อมูลตลอดเวลา การทำการ hash ในทุกครั้งที่มีการเปลี่ยนแปลงเว็บเพจอาจส่งผลต่อประสิทธิภาพของระบบเฝ้าระวัง อย่างไรก็ตาม ระบบที่เราพัฒนามุ่งเน้นการใช้งานง่าย ทำงานเร็ว และเหมาะสำหรับเว็บไซต์หน่วยงานราชการที่มีเนื้อหาแบบคงที่หรือมีการเปลี่ยนแปลงไม่บ่อยครั้ง และในปัจจุบันการประยุกต์ใช้การเรียนรู้ของเครื่อง [21-22] อาจจะเป็นอีกเทคนิคที่สามารถนำมาใช้แก้ปัญหาดังกล่าวได้ แต่อย่างไรวิธีการเรียนรู้ของเครื่องยังมีความท้าทายในแง่ของความยุ่งยากต่อการนำมาประยุกต์ใช้งานจริง รวมถึงค่าใช้จ่ายด้านงบประมาณหากมีการใช้งานจริง

5. สรุปผล

จากการศึกษาการประยุกต์ใช้บริการคลาวด์แบบฟรีมาช่วยในการตรวจสอบและเฝ้าระวังความผิดพลาดของเว็บไซต์ในด้านของการให้บริการด้วยการตรวจสอบจากสถานะรหัส HTTP Status และด้านของการถูกโจมตีด้วยการเปลี่ยนแปลงข้อมูลบนหน้าเว็บไซต์

พบว่าระบบที่นำเสนอสามารถตรวจสอบและแจ้งเตือนได้ 100% ซึ่งระบบดังกล่าวสามารถพัฒนาได้อย่างง่ายและมีความเหมาะสมในการนำมาใช้งานสำหรับองค์กรที่มีห้องศูนย์ข้อมูล โดยการเฝ้าระวังจากระบบคลาวด์ซึ่งอยู่ภายนอกเครือข่ายขององค์กรจะช่วยให้การตรวจสอบและเฝ้าระวังสามารถทำงานได้ตลอดเวลา อย่างไรก็ตามการทำงาน Trigger อาจจะข้อผิดพลาดที่อาจจะเกิดขึ้นได้ แต่มีอัตราความผิดพลาดเพียงเล็กน้อย

6. เอกสารอ้างอิง

[1] A. A. Ganiyu, A. Mishra, J. Elijah, and U. M. Gana. "The Importance of Usability of a Website," *IUP Journal of Information Technology.*, Vol. 13 (No. 3), pp. 27-35, 2017.

[2] W. W. Schilling and M. Alam. "Measuring the reliability of existing web servers". *International Conference on Electro/Information Technology. 17-20 May. Chicago IL USA* : pp. 356-361, 2007. doi: 10.1109/EIT.2007.4374446.

[3] P. Patel, A. H. Ranabahu, and A. P. Sheth. *Service level agreement in cloud computing*. *Wright State University*, 2009.

[4] C. Cérin et al. "Downtime statistics of current cloud solutions," *International Working Group on Cloud Computing Resiliency, Tech. Rep.*, Vol. 1 (No. 2), p. 130, 2013.

[5] N. Albalawi, N. Alamrani, R. Aloufi, M. Albalawi, A. Aljaedi, and A. R. Alharbi. "The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities," *Electronics.*, Vol. 12 (No. 12), p. 2664, 2023.

[6] M. Albalawi, R. Aloufi, N. Alamrani, N. Albalawi, A. Aljaedi, and A. R. Alharbi. "Website Defacement Detection and Monitoring Methods: A Review," *Electronics.*, Vol. 11 (No. 21), p. 3573, 2022.

[7] Ramotion. (21 November 2023). *Website Defacement: Motivations Prevention and Recovery*. [Online] Available : https://www.ramotion.com/blog/website-defacement/

[8] M. Balduzzi et al. "A deep dive into defacement: How geopolitical events trigger web attacks," *Trend Micro Forward-Looking Threat Research (FTR) Team, Tech. Rep,* 2018.

[9] S. Limited. (21 November 2023). *Cybersecurity Statistics Report 2022*. [Online] Available : https://www.sitelock.com/resources/security-report/

[10] ไทยรัฐออนไลน์. (2 พฤศจิกายน 2566). *แก้ไขวุ่น! 9 เว็บไทยถูกแฮกเปลี่ยนหน้าเพจ โดยกลุ่มแฮกเกอร์ในตูนิเซีย*, [ระบบออนไลน์]. แหล่งที่มา: https://www.thairath.co.th/news/local/520485

[11] PPTV Online. (2 พฤศจิกายน 2566). *แฮกเกอร์ แสบ แฮกเว็บไซต์ ศาลรัฐธรรมนูญใส่เพลงหน้าเว็บ*, [ระบบออนไลน์]. แหล่งที่มา: https://www.pptvhd36.com/news/160314

[12] S. P. Mirashe and N. V. Kalyankar, *Cloud computing*. arXiv preprint arXiv:1003.4074, 2010.

[13] Google Apps Script. (12 December 2023). *Build web apps and automate tasks with Google Apps Script*. [Online] Available : https://www.google.com/script/start/

[14] LINE. (20 November 2023). *Receive web service notifications on LINE*. [Online] Available : https://notify-bot.line.me/en/

[15] D. Arunyagool, K. Chamnongthai, and D. Khawparisuth. "Monitoring and Energy Control Inside Home Using Google Sheets with Line Notification". *International Conference on Power, Energy and Innovations. 20-22 Oct. Nakhon*

Ratchasima Thailand : pp. 99-102, 2021. doi: 10.1109/ICPEI52436.2021.9690648

[16] Wanlapa Puttangkul and Mahasak Ketcham. "Management information system for server monitoring by geography information technology," *Science Technology and Innovation*., Vol. 1 (No. 2), pp. 21-31, 2020.

[17] Chet Sriyaem. *Development of computer network monitoring system : a case study of Phetchaburi Rajabhat University*. Master of Arts Program in Educational Informatics. Silpakorn University, 2014.

[18] G. Aceto, A. Botta, W. De Donato, and A. Pescapè. "Cloud monitoring: A survey," *Computer Networks.*, Vol. 57 (No. 9), pp. 2093-2115, 2013.

[19] S. Popa. "WEB Server monitoring," *Annals of University of Craiova-Economic Sciences Series.*, Vol. 2 (No. 36), pp. 710-715, 2008.

[20] Google. (2 November 2023). *Simple Triggers Apps Script.* [Online] Available : https://developers.google.com/apps-script/guides/triggers

[21] S. P. Ayunda, N. Qomariasih, R. B. Hadiprakoso, and H. Kabetta. "Comparative Analysis of Deep Learning Models for Web Defacement Detection Based on Textual Context". *International Conference on Cryptography, Informatics, and Cybersecurity. 22-24 Aug. Bogor Indonesia* : pp. 287-291, 2023. doi: 10.1109/ICoCICs58778.2023.10276697.

[22] A. Kumar and I. Sharma. "Performance Evaluation of Machine Learning Algorithms for Website Defacement Attack Detection". *International Conference on Smart Systems for applications in Electrical Sciences. 7-8 Jul. Tumakuru India* : pp. 1-6, 2023. doi: 10.1109/ICSSES58299.2023.10201194.