หมายเลขโครงงาน	

แบบฟอร์มขอบเขตโครงงาน TEAM-PROJECT 3

ชื่อภาษาไทย การยับยั้งการโจมตีแบบปฏิเสธการให้บริการในระดับแอปพลิเคชัน โดยใช้โปรโตคอลการชำระเงินขนาดย่อม

ชื่อภาษาอังกฤษ APPLICATION-LAYER DDOS MITIGATION USING HTTP 402 MICROPAYMENT PROTOCOL

โดย

นายธนวิชญ์	มุริจันทร์	รหัสนักศึกษา	67015058
นายมอ	รักไทย	รหัสนักศึกษา	67015116
นายวรเทพ	สีนวล	รหัสนักศึกษา	67015124

อาจารย์ที่ปรึกษาหลัก _____

(ผศ.ดร.นภัทร สระเอี่ยม) ลงนามวันที่ 25/11/68

วัตถุประสงค์โดยคร่าวของการนำเสนอโครงงาน TEAM-PROJECT 3

1. เพื่อศึกษาและพัฒนาระบบต้นแบบ (Prototype) สำหรับการป้องกันการโจมตีแบบปฏิเสธการให้บริการในระดับแอปพลิเคชัน (Application-Layer DDoS) โดยประยุกต์ใช้มาตรฐาน HTTP Status Code 402 (Payment Required)
2. เพื่อเปรียบเทียบประสิทธิภาพการทำงานและเสถียรภาพของเซิร์ฟเวอร์ ระหว่างระบบที่ไม่มีกลไกป้องกัน (Traditional Server) และระบบที่มีกลไกป้องกันด้วย HTTP 402 (402 Shield) ภายใต้สภาวะการถูกโจมตี
3. เพื่อพิสูจน์สมมติฐานเรื่องการใช้ต้นทุนทางเศรษฐศาสตร์ (Economic Denial of Sustainability) ในการยับยั้งการก่อกวนระบบเครือข่าย โดยจำลองระบบชำระเงินขนาดเล็ก (Micropayment)
4. เพื่อนำเสนอแนวทางเลือกใหม่ในการออกแบบระบบควบคุมการเข้าถึง (Access Control) ของ API ที่มีความยืดหยุ่นและสามารถลดภาระการประมวลผลของเซิร์ฟเวอร์ได้

หมายเลขโครงงาน	

ขอบเขตของโครงงาน TEAM-PROJECT 3

1. ขอบเขตด้านการพัฒนาระบบ แบ่งออกเป็น 3 ส่วน 1) ส่วนเกตเวย์ผู้ให้บริการ พัฒนา API Gateway ที่ทำหน้าที่ตรวจสอบสิทธิ์การเข้าถึงทรัพยากร โดยระบบจะตอบกลับด้วยสถานะ HTTP 402 Payment Required หากคำร้องขอ ขาดโทเคนยืนยันการชำระเงิน 2) ส่วนธนาคารจำลอง พัฒนาระบบบัญชีแบบรวมศูนย์ โดยใช้ฐานข้อมูล SQLite เพื่อจำลองการทำธุรกรรมและออกโทเคนชำระเงิน และ 3) จำลองผู้ใช้งาน 2 ประเภท คือ ผู้ใช้ปกติสามารถดำเนินการชำระเงินและแนบโทเคนเพื่อเข้าถึงข้อมูลได้อัตโนมัติและผู้โจมตีจำลองบอทที่ส่งคำร้องขอปริมาณมากโดยไม่มีการชำระเงิน
2. ขอบเขตด้านการทดสอบและวิเคราะห์ จะทดสอบเฉพาะการโจมตีประเภท HTTP Flood Attack (Layer 7) ซึ่งมุ่งเน้นให้เซิร์ฟเวอร์ใช้ทรัพยากรประมวลผลสูง จะไม่ครอบคลุมการโจมตีระดับ Network Layer จะดำเนินการทดสอบบนระบบจำลอง (Localhost) หรือ Cloud Server จำนวน 1-2 เครื่อง มีการเปรียบเทียบประสิทธิภาพระหว่างระบบที่ไม่มีการป้องกันและระบบที่มี 402 Shield โดยวัดจากอัตราการใช้งาน CPU และความสามารถในการให้บริการผู้ใช้ปกติขณะถูกโจมตี
3. ระบบนี้เป็นเพียงต้นแบบเพื่อพิสูจน์แนวคิดการป้องกันเชิงเศรษฐศาสตร์ ไม่รองรับการเชื่อมต่อกับระบบการเงินจริง
4. เน้นการทำงานของระบบหลังบ้าน และการสื่อสารระหว่างเซิร์ฟเวอร์เป็นหลัก ไม่เน้นการพัฒนาส่วนติดต่อผู้ใช้งาน

บล็อกไดอะแกรมของโครงงานที่นำเสนอ

หมายเลขโครงงาน	

แผนการปฏิบัติงานตลอดภาคการศึกษา

ช่วงการดำเนินการ	แผนงานที่จะดำเนินการ	
เดือนที่ 1 (ธ.ค. 2568)	สัปดาห์ที่ 1	ศึกษาทฤษฎี HTTP 402, FastAPI, ออกแบบโครงสร้างฐานข้อมูล (Database Schema) และติดตั้งเครื่องมือ (Environment Setup)
	สัปดาห์ที่ 2	พัฒนาส่วน Mock Banking System (ระบบบัญชีจำลอง) โดยสร้าง API สำหรับเติมเงิน, โอนเงิน และออก Token
	สัปดาห์ที่ 3	พัฒนาส่วน Provider Gateway (ระบบด่านหน้า) โดยสร้าง Middleware สำหรับตรวจสอบ Token และตอบกลับ HTTP 402 และ เชื่อมต่อ Gateway เข้ากับ Bank (Integration Test)
	สัปดาห์ที่ 4	จัดทำและส่งรายงานความก้าวหน้าครั้งที่ 1
เดือนที่ 2 (ม.ค. 2569)	สัปดาห์ที่ 1	พัฒนา Smart Client Agent โดยเขียนโปรแกรมจำลองผู้ใช้ปกติที่สามารถอ่าน 402 Header และจ่ายเงินอัตโนมัติ
	สัปดาห์ที่ 2	พัฒนา Attacker Bot โดยเขียนสคริปต์จำลองการโจมตีแบบ HTTP Flood (ยิง Request รัวๆ โดยไม่จ่ายเงิน)
	สัปดาห์ที่ 3	พัฒนาระบบ Monitoring โดยเขียนสคริปต์เก็บค่า CPU Usage, RAM และ Response Time ลงไฟล์ Log (CSV/Database)
	สัปดาห์ที่ 4	จัดทำและส่งรายงานความก้าวหน้าครั้งที่ 2

หมายเลขโครงงาน	

แผนการปฏิบัติงานตลอดภาคการศึกษา (ต่อ)

ช่วงการดำเนินการ	แผนงานที่จะดำเนินการ	
เดือนที่ 3 (ก.พ. 2569)	สัปดาห์ที่ 1	ดำเนินการทดลอง Scenario A (ไม่มีระบบป้องกัน) โดยการยิงโจมตีใส่ Server เปล่า และบันทึกผลประสิทธิภาพ
	สัปดาห์ที่ 2	ดำเนินการทดลอง Scenario B (มีระบบ 402 Shield) โดยการยิงโจมตีใส่ Server ที่มีระบบป้องกัน และบันทึกผลประสิทธิภาพ
	สัปดาห์ที่ 3	นำข้อมูล Log มาประมวลผลและสร้างกราฟเปรียบเทียบ (Data Visualization) และวิเคราะห์ผลการทดลองตามสมมติฐาน EDoS (Economic Denial of Sustainability)
	สัปดาห์ที่ 4	จัดทำและส่งรายงานความก้าวหน้าครั้งที่ 3
เดือนที่ 4 (มี.ค. 2569)	สัปดาห์ที่ 1	เรียบเรียงและจัดทำเอกสารรายงานฉบับสมบูรณ์ (บทที่ 1-5)
	สัปดาห์ที่ 2	จัดเตรียมสื่อนำเสนอ (Presentation Slide) และเตรียม Demo ระบบจริง และตรวจสอบความถูกต้องของรูปเล่มและบรรณานุกรม
	สัปดาห์ที่ 3	จัดทำและส่งรายงานฉบับสมบูรณ์

หมายเหตุ

รายงานความก้าวหน้าจะมีกำหนดส่งของทุกๆ เดือน ตามประกาศของภาควิชาฯ โดยในรายงานจะต้องมีหลักฐานผลการดำเนินงานเชิงประจักษ์ เช่น ผลการทดลอง ผลการทดสอบ และสอดคล้องตามแผนการปฏิบัติงานที่ได้แสดงไว้

หมายเลขโครงงาน	

บทคัดย่อโครงงาน TEAM-PROJECT 3

ชื่อภาษาไทย การยับยั้งการโจมตีแบบปฏิเสธการให้บริการในระดับแอปพลิเคชัน โดยใช้โปรโตคอลการชำระเงินขนาดย่อม

ชื่อภาษาอังกฤษ APPLICATION-LAYER DDOS MITIGATION USING HTTP 402 MICROPAYMENT PROTOCOL

บทคัดย่อ

การโจมตีแบบปฏิเสธการให้บริการแบบกระจายในระดับแอปพลิเคชัน (Application-Layer DDoS) เป็นภัยคุกคามทางไซเบอร์ที่สร้างความเสียหายอย่างรุนแรงและยากต่อการตรวจจับ เนื่องจากรูปแบบของคำร้องขอมีความคล้ายคลึงกับการใช้งานของผู้ใช้ปกติ ส่งผลให้ทรัพยากรของเซิร์ฟเวอร์ถูกใช้งานจนหมดและระบบหยุดทำงาน โครงงานนี้จึงนำเสนอแนวทางการป้องกันเชิงรุกโดยประยุกต์ใช้แนวคิดการปฏิเสธความยั่งยืนทางเศรษฐศาสตร์ (Economic Denial of Sustainability - EDoS) ผ่านมาตรฐาน HTTP Status Code 402 (Payment Required) เพื่อสร้างกำแพงต้นทุนในการเข้าถึงทรัพยากร ในการทดลองดำเนินการโดยเปรียบเทียบประสิทธิภาพของระบบระหว่างเซิร์ฟเวอร์ที่ไม่มีกลไกป้องกันและเซิร์ฟเวอร์ที่มีกลไก HTTP 402 ภายใต้สภาวะการถูกโจมตีแบบ HTTP Flood ผลการดำเนินงานมุ่งหวังที่จะแสดงให้เห็นว่ากลไกดังกล่าวสามารถลดอัตราการใช้งานหน่วยประมวลผลกลาง (CPU Usage) และรักษาเสถียรภาพในการให้บริการแก่ผู้ใช้ปกติ (Availability) ได้อย่างมีนัยสำคัญ ซึ่งจะเป็นแนวทางใหม่ในการออกแบบความปลอดภัยของ API ในอนาคต

Abstract

Application-Layer Distributed Denial of Service (Application-Layer DDoS) attacks are severe cybersecurity threats that are difficult to detect because their request patterns closely resemble legitimate user behavior, leading to server resource exhaustion and system failure. Consequently, this project proposes a proactive defense mechanism by applying the concept of Economic Denial of Sustainability (EDoS) via the HTTP Status Code 402 (Payment Required) standard to establish a cost barrier for resource access. The experiment involves comparing system performance between a server without defense mechanisms and a server equipped

with the HTTP 402 mechanism under HTTP Flood attack conditions. The expected results aim to demonstrate that this mechanism can significantly reduce CPU Usage and maintain service Availability for legitimate users, offering a novel approach for future API security design.