กรณีศึกษา : การออกแบบเว็บแอปพลิเคชันบน Cloud ให้มั่นคงปลอดภัย

ปัจจุบันมี Startup เกิดขึ้นบนระบบ Cloud เป็นส่วนใหญ่ ดังนั้นหากเราเริ่มวางแผนให้ดีตั้งแต่แรกก็จะเป็นรากฐานที่เข้มแข็งให้ธุรกิจเติบโตต่อไปได้โดยไม่เกิดเหตุการณ์รั่วไหลต่างๆ อย่างที่ปรากฏในข่าวหลายต่อหลายครั้ง วันนี้เราจึงขอสรุปบทความจาก F5 Labs ที่ได้เขียนขึ้นมาจากการสังเกตการณ์ Startup ที่ชื่อ Wanderlust Society โดยเกิดจากทีมงานมากประสบการณ์จาก AWS ร่วมกันสร้างสังคมออนไลน์ให้กับนักท่องเที่ยวในการวางแผนการเดินทาง

ตั้งเป้าหมาย เรื่องแรกเลยก่อนวางระบบใดๆ จะต้องเริ่มตั้งเป้าหมายก่อน โดยทางทีมงาน Wanderlust เองได้ตั้งเป้าหมายขึ้นมาดังนี้

1. มีความเหมาะสมกับการใช้งานบนมือถือ
2. มีความมั่นคงปลอดภัย
3. รวดเร็ว
4. ต้องให้บริการได้ต่อเนื่อง
5. ปรับขนาดได้ง่าย

ประเมินความเสี่ยง นอกจากมีเป้าหมายที่ดีแล้วเราต้องมีโมเดลที่สามารถปฏิบัติงานได้จริงสำหรับการออกแบบด้านความมั่นคงปลอดภัย โดยให้นักพัฒนาหรือผู้วางโครงสร้างของระบบใช้อ้างอิงได้ ยิ่งกว่านั้นความมั่นคงปลอดภัยในแต่ละบริษัทก็มีความหมายต่างกันออกไป ดังนั้น Startup แต่ละรายก็ควรลิสต์ปัจจัยความเสี่ยงของตนเพื่อให้คนในองค์กรเข้าใจภาพตรงกัน ในส่วนของ Wanderlust ได้นิยามความเสี่ยงในองค์กรของตนซึ่งสามารถแก้ไขเปลี่ยนแปลงได้หากต่อไปมีเงื่อนไขเปลี่ยนแปลง

1. ผู้ใช้งานที่ไม่ได้ถูกพิสูจน์ตัวตนควรจะทำได้แค่ อ่านและเขียนข้อมูล โดย APIs ที่ใช้งานต้องแสดงว่ามาจากการใช้งานแบบสาธารณะอย่างชัดเจน
2. ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วควรจะสามารถเปลี่ยนแปลง และเห็นข้อมูลของตนได้
3. ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วควรจะเห็นข้อมูลที่ถูกแชร์มาจากผู้ใช้งานอื่นได้
4. ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วไม่ควรจะสามารถเขียนหรืออ่านข้อมูลที่ไม่ใช่ของตนได้เช่น ข้อมูลระบบและข้อมูลสนับสนุนแอปพลิเคชันอื่นๆ
5. ผู้ไม่ประสงค์ดีไม่ควรเข้าถึงระบบเพื่อขโมย Credential ของผู้ใช้ได้
6. ผู้ไม่ประสงค์ดีไม่ควรขโมย Credential ของผู้ใช้ระหว่างทางได้

7. ผู้ไม่ประสงค์ดีไม่ควรขโมยหรือทำให้ข้อมูลของ Wanderlust Society ด่างพร้อยได้

8. ผู้ไม่ประสงค์ดีไม่ควรทำ แก้ไข ลดระดับ หรือทำระบบทำงานผิดพลาดได้

เหมาะสมกับการใช้งานบนมือถือ เพื่อให้ใช้งานได้อย่างรวดเร็ว Wanderlust ใช้การบีบอัด JavaScript ส่วนหลักเหลือเพียง 90 KB เท่านั้น อีกทั้งยังใช้การโหลดแบบ Asynchonous หรือการที่ไม่ต้องรอเพื่อดึงข้อมูลอื่นๆ มาแสดงผลพร้อมกันทำให้มันสามารถทำงานบนอินเทอร์เน็ต 3G ความเร็วต่ำได้

ความมั่นคงปลอดภัยของ Wanderlust

- ใช้ HTTPS ในการเชื่อมต่อเท่านั้น
- วาง Firewall ขวางทั้งขาเข้า และออก เพื่อลดการโจมตีภายนอก และป้องกันการรั่วไหล
- จำกัดการใช้งานฐานข้อมูล
- ไม่ใช้ Subnet แบบสาธารณะ
- จำกัด Firewall ให้ผ่านได้เพียงพอร์ตเดียว
- ใช้การควบคุมการเข้าถึงด้วยการพิสูจน์ตัวตน โดยได้เลือกดึงบัญชีผู้ใช้ที่มาจาก Facebook เนื่องจากมีผู้ใช้จำนวนมากและลงทะเบียนไว้แล้ว อีกทั้ง Facebook เองยังพิสูจน์ตัวเองแล้วว่ามีความมั่นคงปลอดภัย และเชื่อถือได้
- จำกัดการเข้าถึงผู้ใช้ที่พิสูจน์ตัวตนแล้วด้วยการติดตามการใช้งาน โดยสร้าง Token ให้ผู้ใช้เพื่อร้องขอเข้าถึงบริการ อีกทั้งสามารถจำกัดระยะเวลา และ Token จะถูกลบออกไปเมื่อผู้ใช้ Logout
- มีระบบที่ตั้งอยู่บนฝั่งเซิร์ฟเวอร์เพื่อตรวจสอบข้อมูล และ SQL statement ที่ได้รับเพื่อป้องกันการโจมตีแบบ Injection ด้วย

รวดเร็ว ใช้ระบบ CDN เพื่อเป็น Cache ให้พวกรูปภาพและข้อมูลที่ใช้บ่อยเพื่อลดการดาวน์โหลดโดยตรงจากเซิร์ฟเวอร์

สามารถให้บริการได้ต่อเนื่อง ใช้ระบบ Load Balancer เพื่อกระจายงานให้ EC2 เซิร์ฟเวอร์ และติดตามระบบด้วย Cloudwatch นอกจากนี้ยัง Deploy ระบบไว้ในหลายโซนด้วย

ปรับขนาดได้ง่าย Wanderlust ได้ใช้โมเดลแบบ Microservice กับแอปพลิเคชัน ซึ่งหมายความว่ามันประกอบมาจากระบบเล็กๆ หลายส่วนทำให้สามารถขยาย และติดตั้งแยกกันได้ รวมถึงใช้โค้ดบน Docker ด้วย

การจับคู่ระหว่างความเสี่ยงและการควบคุมของ Wanderlust

ความเสี่ยงด้านความมั่นคงปลอดภัย	การป้องกันที่ใช้			
	HTTPS	ควบคุมการเข้าถึงและการใช้ Token ติดตามผู้ใช้	ระบบตรวจสอบพารามิเตอร์และ SQL Statement	บริการที่ต่อเนื่องและระบบติดตาม
ผู้ใช้งานที่ไม่ได้ถูกพิสูจน์ตัวตนควรจะทำได้แค่ อ่านและเขียนข้อมูลโดย APIs ที่ใช้งานต้องแสดงว่ามาจากการใช้งานแบบสาธารณะอย่างชัดเจน		ป้องกัน		
ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วควรจะสามารถเปลี่ยนแปลงและเห็นข้อมูลของตนได้	ป้องกัน	ป้องกัน		
ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วควรจะเห็นข้อมูลที่ถูกแชร์มาจากผู้ใช้งานอื่นได้		ป้องกัน		
ผู้ใช้งานที่ผ่านการพิสูจน์ตัวตนแล้วไม่ควรจะสามารถเขียนหรืออ่านข้อมูลที่ไม่ใช่ของตนได้เช่น ข้อมูลระบบและข้อมูลสนับสนุนแอปพลิเคชันอื่นๆ		ป้องกัน		
ผู้ไม่ประสงค์ดีไม่ควรเข้าถึงระบบเพื่อขโมย Credential ของผู้ใช้ได้	ป้องกัน	ป้องกัน	ป้องกัน	
ผู้ไม่ประสงค์ดีไม่ควรขโมย Credential ของผู้ใช้ระหว่างการส่งได้	ป้องกัน			
ผู้ไม่ประสงค์ดีไม่ควรขโมยหรือทำให้ข้อมูลของ Wanderlust Society ด่างพร้อยได้	ป้องกัน	ป้องกัน		
ผู้ไม่ประสงค์ดีไม่ควรทำ แก้ไข ลดระดับ หรือทำระบบทำงานผิดพลาดได้			ป้องกัน	ป้องกัน

อย่างไรก็ดี Wanderlust ก็มีสิ่งที่ต้องชั่งน้ำหนักความสำคัญให้ดี ตัวอย่างเช่น มีผู้ใช้ที่ไม่เชื่อถือ Facebook จึงไม่ใช้งาน ดังนั้น Wanderlust ก็อาจจะต้องใช้บริการการระบุตัวตนจาก Google หรือแหล่งอื่นๆ หรือแม้แต่พิจารณาการสร้างระบบพิสูจน์ตัวตนของตัวเอง หากเป็นเช่นนั้นทาง Wanderlust ก็ต้องทดสอบระบบให้ดี พร้อมทั้งเลือกว่าจะใช้ Cloud เจ้าเดิมหรืออื่นๆ ตามความเหมาะสมด้านค่าใช้จ่ายหรือแม้แต่ On-premise ก็ตาม

ที่มา: www.techtalkthai.com