**Authentication vulnerabilities (ช่องโหว่การตรวจสอบสิทธิ์)**

**What is authentication?** การตรวจสอบความถูกต้องคืออะไร

การตรวจสอบความถูกต้อง คือ กระบวนการตรวจสอบตัวตนของผู้ใช้หรือลูกค้า เว็บไซต์อาจเปิดเผยต่อใครก็ตามที่เชื่อมต่อกับอินเทอร์เน็ต ซึ่งทำให้กลไกการตรวจสอบความถูกต้องที่แข็งแกร่งเป็นส่วนสำคัญของการรักษาความปลอดภัยบนเว็บที่มีประสิทธิภาพ

การตรวจสอบความถูกต้องมีสามประเภทหลัก

* สิ่งที่รู้ เช่น รหัสผ่านหรือคำตอบสำหรับคำถามด้านความปลอดภัย "knowledge factors"
* สิ่งที่มี physical object เช่น โทรศัพท์มือถือหรือโทเค็นความปลอดภัย "possession factors"
* สิ่งที่เป็นหรือทำ biometrics or patterns of behavior "inherence factors"

**What is the difference between authentication and authorization?**

Authentication การพิสูจน์ตัวตน คือ กระบวนการยืนยันว่าผู้ใช้เป็นใคร

Authorization การอนุญาต คือ การยืนยันว่าผู้ใช้ได้รับอนุญาตให้ทำบางสิ่งหรือไม่

ตัวอย่างเช่น การพิสูจน์ตัวตนจะกำหนดว่าบุคคลที่พยายามเข้าถึงเว็บไซต์โดยใช้ชื่อผู้ใช้ Carlos123 เป็นคนเดียวกับที่สร้างบัญชีนั้นจริงหรือไม่

เมื่อ Carlos123 ผ่านการพิสูจน์ตัวตนแล้ว การอนุญาตจะกำหนดว่าบุคคลนั้นได้รับอนุญาตให้ทำอะไร พวกเขาอาจได้รับอนุญาตให้เข้าถึงข้อมูลส่วนบุคคลเกี่ยวกับผู้ใช้รายอื่น เช่น การลบบัญชีของผู้ใช้รายอื่น

**How do authentication vulnerabilities arise?**

ช่องโหว่ส่วนใหญ่ในกลไกการยืนยันตัวตนเกิดขึ้นได้สองวิธีดังต่อไปนี้

* กลไกการยืนยันตัวตนอ่อนแอ เนื่องจากไม่สามารถป้องกันการโจมตีแบบ brute-force ได้อย่างเหมาะสม
* ข้อบกพร่องทางตรรกะหรือการเขียนโค้ดที่ไม่ดีในการใช้งานทำให้ผู้โจมตีสามารถข้ามกลไกการยืนยันตัวตนได้ทั้งหมด เรียกว่า "broken authentication"

ในหลายพื้นที่ของการพัฒนาเว็บ ข้อบกพร่องทางตรรกะทำให้เว็บไซต์ทำงานผิดปกติ ซึ่งอาจเป็นปัญหาความปลอดภัยหรือไม่ก็ได้ อย่างไรก็ตาม เนื่องจากการยืนยันตัวตนมีความสำคัญอย่างยิ่งต่อความปลอดภัย จึงมีความเป็นไปได้สูงที่ตรรกะการยืนยันตัวตนที่บกพร่องจะทำให้เว็บไซต์เผชิญกับปัญหาความปลอดภัย

**What is the impact of vulnerable authentication?**

ผลกระทบของช่องโหว่ในการตรวจสอบสิทธิ์อาจร้ายแรง หากผู้โจมตีสามารถหลีกเลี่ยงการตรวจสอบสิทธิ์หรือใช้วิธีการ brute-force เข้าสู่บัญชีผู้ใช้รายอื่น ผู้โจมตีจะสามารถเข้าถึงข้อมูลและฟังก์ชันต่างๆ ของบัญชีที่ถูกบุกรุกได้ทั้งหมด หากผู้โจมตีสามารถเจาะระบบบัญชีที่มีสิทธิ์ใช้งานสูง เช่น ผู้ดูแลระบบได้ ผู้โจมตีจะสามารถควบคุมแอปพลิเคชันทั้งหมดได้อย่างสมบูรณ์และอาจเข้าถึงโครงสร้างพื้นฐานภายในได้

**What is the impact of vulnerable authentication?**

แม้ว่าบัญชีที่มีสิทธิ์ใช้งานต่ำอาจทำให้ผู้โจมตีเข้าถึงข้อมูลที่ไม่ควรเข้าถึงได้ เช่น ข้อมูลทางธุรกิจที่ละเอียดอ่อนทางการค้า ถึงบัญชีจะไม่สามารถเข้าถึงข้อมูลที่ละเอียดอ่อนได้ แต่ผู้โจมตียังสามารถเข้าถึง additional pages ซึ่งจะเป็นช่องทางการโจมตีเพิ่มเติม การโจมตีที่มีความรุนแรงสูงมักไม่สามารถเกิดขึ้นได้จาก publicly accessible pages แต่อาจเกิดขึ้นได้จาก internal page

**Vulnerabilities in password-based login**

สำหรับเว็บไซต์ที่ใช้กระบวนการเข้าสู่ระบบโดยใช้รหัสผ่าน ผู้ใช้จะต้องลงทะเบียนเพื่อสร้างบัญชีของตนเองหรือได้รับบัญชีจากผู้ดูแล ซึ่งบัญชีนี้จะถูกเชื่อมโยงกับชื่อผู้ใช้และรหัสผ่านลับเฉพาะ เมื่อเข้าสู่ระบบผู้ใช้จะต้องกรอกข้อมูลเหล่านี้ในแบบฟอร์มเพื่อยืนยันตัวตน

ในกรณีนี้ การที่ผู้ใช้รู้รหัสผ่านลับนั้นถือเป็นหลักฐานเพียงพอในการยืนยันตัวตนของผู้ใช้ ซึ่งหมายความว่าความปลอดภัยของเว็บไซต์อาจถูกทำลายได้หากผู้โจมตีสามารถดึงข้อมูลหรือเดารหัสผ่านของผู้ใช้อื่นได้

วิธีการที่ผู้โจมตีอาจใช้ในการโจมตีมีหลายรูปแบบ ซึ่งบทความต่อไปนี้จะแสดงให้เห็นถึงวิธีการใช้การโจมตีแบบเดารหัสผ่าน (brute-force attack) และข้อบกพร่องในกระบวนการป้องกันการโจมตีแบบเดารหัส นอกจากนี้ยังอธิบายเกี่ยวกับช่องโหว่ของการยืนยันตัวตนแบบ HTTP basic authentication

**Brute-force attacks**

การโจมตีแบบเดารหัสผ่าน (Brute-force attack) คือ การที่ผู้โจมตีใช้วิธีการลองผิดลองถูกเพื่อเดาชื่อผู้ใช้และรหัสผ่านที่ถูกต้อง โดยการโจมตีประเภทนี้มักจะทำผ่านระบบอัตโนมัติที่ใช้รายการคำศัพท์ที่มีชื่อผู้ใช้และรหัสผ่านอยู่แล้ว การใช้กระบวนการอัตโนมัตินี้ โดยเฉพาะการใช้เครื่องมือเฉพาะทาง ทำให้ผู้โจมตีสามารถพยายามเข้าสู่ระบบจำนวนมากได้อย่างรวดเร็ว

การโจมตีแบบเดารหัสไม่ใช่การเดาสุ่มที่ไม่มีหลักการเสมอไป ผู้โจมตีอาจใช้หลักการพื้นฐานหรือข้อมูลที่มีอยู่สาธารณะในการช่วยเดาชื่อผู้ใช้และรหัสผ่าน ทำให้การโจมตีแบบเดารหัสมีประสิทธิภาพมากยิ่งขึ้น ซึ่งทำให้เว็บไซต์ที่ใช้การเข้าสู่ระบบด้วยรหัสผ่านเพียงอย่างเดียวอาจมีความเสี่ยงสูง หากไม่มีการป้องกันการโจมตีแบบเดารหัสที่เพียงพอ

**Brute-forcing usernames**

ชื่อผู้ใช้ (Username) มักเป็นสิ่งที่เดาได้ง่ายหากมีรูปแบบที่สามารถจดจำได้ เช่น อีเมล โดยเฉพาะในธุรกิจต่าง ๆ ที่มักใช้รูปแบบเช่น firstname.lastname@somecompany.com อย่างไรก็ตาม ถึงแม้ว่าจะไม่มีรูปแบบชัดเจน บางครั้งบัญชีที่มีสิทธิ์เข้าถึงสูง (high-privileged accounts) ก็ถูกสร้างด้วยชื่อผู้ใช้ที่คาดเดาได้ เช่น admin หรือ administrator

ในระหว่างการตรวจสอบความปลอดภัย ควรตรวจสอบว่าเว็บไซต์เปิดเผยชื่อผู้ใช้ที่เป็นไปได้หรือไม่ เช่น สามารถเข้าถึงโปรไฟล์ของผู้ใช้ได้โดยไม่ต้องเข้าสู่ระบบหรือไม่ แม้ว่าจะไม่สามารถเห็นเนื้อหาของโปรไฟล์ แต่ชื่อที่ใช้ในโปรไฟล์อาจเป็นชื่อเดียวกับชื่อผู้ใช้ที่ใช้เข้าสู่ระบบ นอกจากนี้ ควรตรวจสอบการตอบสนองของ HTTP เพื่อดูว่ามีการเปิดเผยที่อยู่อีเมลหรือไม่ บางครั้งการตอบสนองอาจมีที่อยู่อีเมลของผู้ใช้ที่มีสิทธิ์สูง เช่น ผู้ดูแลระบบหรือฝ่ายสนับสนุนด้านไอที

**Brute-forcing passwords**

การโจมตีแบบเดารหัสผ่าน (Brute-forcing passwords) สามารถใช้กับรหัสผ่านได้เช่นกัน ซึ่งระดับความยากง่ายในการเดาจะขึ้นอยู่กับความแข็งแกร่งของรหัสผ่าน หลายเว็บไซต์จึงมีนโยบายการตั้งรหัสผ่านที่บังคับให้ผู้ใช้สร้างรหัสผ่านที่มีความซับซ้อนสูง เพื่อให้ยากต่อการโจมตีแบบเดารหัสผ่านเพียงอย่างเดียว โดยปกติแล้วนโยบายนี้จะกำหนดให้รหัสผ่านต้องมี

- จำนวนอักขระขั้นต่ำ

- การผสมผสานระหว่างตัวอักษรเล็กและใหญ่

- อักขระพิเศษอย่างน้อยหนึ่งตัว

**Brute-forcing passwords - Continued**

แม้ว่ารหัสผ่านที่มีความซับซ้อนสูง (high-entropy) จะยากต่อการเดาด้วยการใช้คอมพิวเตอร์เพียงอย่างเดียว แต่เรายังสามารถใช้ความรู้พื้นฐานเกี่ยวกับพฤติกรรมมนุษย์ในการค้นหาช่องโหว่ที่ผู้ใช้อาจสร้างโดยไม่รู้ตัว แทนที่จะสร้างรหัสผ่านที่แข็งแกร่งด้วยการสุ่มอักขระ ผู้ใช้มักเลือกใช้รหัสผ่านที่จดจำง่ายและพยายามปรับให้ตรงตามนโยบายรหัสผ่าน เช่น หากใช้ "mypassword" ไม่ได้ ผู้ใช้อาจลองใช้ "Mypassword1!" หรือ "Myp4$$w0rd" แทน

ในกรณีที่นโยบายกำหนดให้ผู้ใช้ต้องเปลี่ยนรหัสผ่านอย่างสม่ำเสมอ ผู้ใช้มักจะเปลี่ยนเพียงเล็กน้อยและเป็นรูปแบบที่เดาได้ เช่น เปลี่ยนจาก "Mypassword1!" เป็น "Mypassword1?" หรือ "Mypassword2!"

ความรู้เกี่ยวกับรูปแบบรหัสผ่านที่น่าจะใช้และการเปลี่ยนแปลงที่คาดการณ์ได้นี้ทำให้การโจมตีแบบเดารหัสผ่านสามารถทำได้อย่างซับซ้อนและมีประสิทธิภาพมากขึ้นกว่าการลองทุกการรวมกันของอักขระ

**Username enumeration**

การระบุชื่อผู้ใช้ (Username enumeration) คือ กระบวนการที่ผู้โจมตีสามารถสังเกตการเปลี่ยนแปลงของพฤติกรรมเว็บไซต์เพื่อดูว่าชื่อผู้ใช้ที่กรอกนั้นมีอยู่จริงหรือไม่

การระบุชื่อผู้ใช้มักเกิดขึ้นในหน้าล็อกอิน เช่น เมื่อผู้ใช้กรอกชื่อผู้ใช้ที่ถูกต้องแต่รหัสผ่านผิด หรือในแบบฟอร์มลงทะเบียน เมื่อผู้ใช้กรอกชื่อผู้ใช้ที่มีคนใช้แล้ว ซึ่งทำให้ผู้โจมตีสามารถสร้างรายการชื่อผู้ใช้ที่ถูกต้องได้อย่างรวดเร็ว ลดเวลาและความพยายามที่จำเป็นในการโจมตีแบบเดารหัสผ่าน

ในระหว่างการโจมตีแบบเดารหัสผ่านบนหน้าล็อกอิน คุณควรให้ความสนใจกับความแตกต่างดังต่อไปนี้:

1. **รหัสสถานะ (Status codes):** ในการโจมตีแบบเดารหัสผ่าน รหัสสถานะ HTTP ที่ถูกส่งกลับมามักจะเหมือนกันในกรณีส่วนใหญ่ เนื่องจากชื่อผู้ใช้หรือรหัสผ่านส่วนใหญ่จะไม่ถูกต้อง หากมีการทายที่ส่งคืนรหัสสถานะที่แตกต่างไป อาจเป็นสัญญาณบ่งชี้ว่าชื่อผู้ใช้นั้นถูกต้อง หลักปฏิบัติที่ดีคือให้เว็บไซต์ส่งรหัสสถานะเดียวกันเสมอ ไม่ว่าผลลัพธ์จะเป็นอย่างไร แต่บางเว็บไซต์ไม่ได้ปฏิบัติตามหลักการนี้
2. **ข้อความแสดงข้อผิดพลาด (Error messages):** บางครั้ง ข้อความแสดงข้อผิดพลาดจะแตกต่างกัน ขึ้นอยู่กับว่าทั้งชื่อผู้ใช้และรหัสผ่านไม่ถูกต้องหรือมีแค่รหัสผ่านที่ไม่ถูกต้อง หลักปฏิบัติที่ดีคือให้เว็บไซต์ใช้ข้อความที่เหมือนกันในทั้งสองกรณี แต่บางครั้งอาจมีข้อผิดพลาดเล็กน้อยในการพิมพ์ ทำให้ข้อความทั้งสองต่างกัน แม้แต่การพิมพ์ผิดเพียงหนึ่งตัวอักษรก็ทำให้ข้อความแสดงผลแตกต่างกันได้ แม้ว่าตัวอักษรนั้นจะไม่แสดงบนหน้าเว็บก็ตาม
3. **เวลาตอบสนอง (Response times):** หากการร้องขอส่วนใหญ่ถูกประมวลผลด้วยเวลาตอบสนองที่คล้ายกัน แต่บางคำขอมีเวลาที่แตกต่างไป อาจแสดงว่ามีบางสิ่งที่แตกต่างเกิดขึ้นเบื้องหลัง เช่น เว็บไซต์อาจตรวจสอบความถูกต้องของรหัสผ่านก็ต่อเมื่อชื่อผู้ใช้ถูกต้อง ซึ่งอาจทำให้เวลาตอบสนองเพิ่มขึ้นเล็กน้อย วิธีนี้อาจละเอียดอ่อน แต่ผู้โจมตีสามารถทำให้ความล่าช้านี้ชัดเจนขึ้นได้โดยการป้อนรหัสผ่านที่ยาวมากจนเว็บไซต์ใช้เวลามากขึ้นในการประมวลผล