

วันที่ 27 ตุลาคม 2569

พบช่องโหว่ใน Brotli Decoder ที่อาจนำไปสู่การรันโค้ดจากระยะไกลบน Node.js และส่งผลกระทบต่อ Bun, Python รวมถึงซอฟต์แวร์ยอดนิยมอื่นๆ



พบช่องโหว่ด้านความปลอดภัยที่ทำให้หน่วยความจำเสียหายใน Brotli Decoder ของ Google ซึ่งเป็นส่วนประกอบที่ใช้คลายการบีบอัดข้อมูลในซอฟต์แวร์หลายประเภท โดยผลิตภัณฑ์ที่ได้รับผลกระทบประกอบด้วย Bun, .NET ของ Microsoft, แพ็กเกจ Brotli และ brotlicffi ของ Python รวมถึง Envoy Proxy ที่เปิดใช้งานการคลายการบีบอัดข้อมูลด้วย Brotli

ช่องโหว่นี้ได้รับรหัส CVE-2026-111497 โดยผู้โจมตีสามารถสร้างข้อมูลบีบอัดที่ถูกดัดแปลงเป็นพิเศษ เพื่อทำให้โปรแกรมเขียนข้อมูลเกินขอบเขตของพื้นที่หน่วยความจำที่จัดสรรไว้ ส่งผลให้ข้อมูลในหน่วยความจำเสียหาย และอาจทำให้โปรแกรมหยุดทำงาน เกิดการรั่วไหลของข้อมูลในหน่วยความจำ หรือเปิดโอกาสให้ผู้โจมตีรันโค้ดที่ต้องการได้

นักวิจัยจาก Imperva ได้สาธิตให้เห็นว่าสามารถใช้ช่องโหว่นี้รันโค้ดบน Node.js เวอร์ชันหนึ่งที่ยังไม่ได้รับการแก้ไขได้สำเร็จ อย่างไรก็ตาม ยังไม่มีการยืนยันว่าสามารถโจมตีในลักษณะเดียวกันบนซอฟต์แวร์อื่นๆ ได้ และยังไม่พบรายงานการนำช่องโหว่นี้ไปใช้โจมตีจริง

ช่องโหว่นี้ถูกเปิดเผยครั้งแรกเมื่อวันที่ 21 ตุลาคม โดย Node.js และ Google ได้ออกแพตช์แก้ไขในวันเดียวกัน ก่อนที่ผู้พัฒนาซอฟต์แวร์รายอื่นจะทยอยออกอัปเดตตามมา เนื่องจากช่องโหว่อยู่นี้ในตัวถอดรหัสของ Google และซอฟต์แวร์แต่ละตัวมีการนำโค้ดส่วนนี้ไปใช้งานแยกกัน จึงจำเป็นต้องอัปเดตส่วนประกอบดังกล่าวในแต่ละผลิตภัณฑ์โดยตรง

Brotli เป็นรูปแบบการบีบอัดข้อมูลที่พัฒนาโดย Google และถูกนำมาใช้อย่างแพร่หลายบนเว็บไซต์ เพื่อลดขนาดหน้าเว็บและข้อมูลที่รับส่งผ่านเครือข่าย โดยช่องโหว่นี้อยู่ในโค้ดภาษา C ที่ทำหน้าที่คลายการบีบอัดข้อมูล Brotli ทำให้ซอฟต์แวร์ใดก็ตามที่นำโค้ดดังกล่าวไปใช้งานอาจได้รับผลกระทบตามไปด้วย ไม่ว่าจะเป็นระบบสำหรับรันโปรแกรม Proxy หรือแพ็กเกจของ Python

ซอฟต์แวร์ที่ได้รับผลกระทบ

ผลิตภัณฑ์	เวอร์ชันที่ได้รับผลกระทบ	เวอร์ชันที่แก้ไขแล้ว	เงื่อนไขที่ทำให้เสี่ยงต่อการโจมตี
Google Brotli (C library)	1.0.3 ถึง 1.2.0	1.0.10, 1.1.1, 1.2.1	เมื่อมีการคลายการบีบอัดข้อมูล Brotli จากแหล่งที่ไม่น่าเชื่อถือ
Node.js	22.x, 24.x และ 26.x ก่อนเวอร์ชันแก้ไข รวมถึง 20.x และเวอร์ชันเก่ากว่าที่สิ้นสุดการสนับสนุนแล้ว	22.23.4, 24.22.1, 26.12.1	เมื่อ fetch() รับข้อมูลตอบกลับแบบ br หรือ Express และ body-parser คลายการบีบอัดข้อมูล br ใน Request Body
Bun	1.4.3 และเวอร์ชันก่อนหน้า	1.4.4	เมื่อ fetch() รับข้อมูลตอบกลับแบบ br หรือมีการใช้งาน node:zlib
.NET	8.0, 9.0 และ 10.0 ก่อนเวอร์ชันแก้ไข รวมถึง .NET 11 Preview ก่อน RC3	8.0.33, 9.0.22, 10.0.14, 11.0 RC3	เมื่อเปิดใช้ Brotli ใน ASP.NET Core Request Decompression หรือ HttpClient Decompression หรือใช้ Brotli API คลายการบีบอัดข้อมูลที่น่าเชื่อถือ
Envoy	1.39.2, 1.38.5, 1.37.7, 1.36.11	1.40.0, 1.39.3, 1.38.6, 1.37.8, 1.36.12	เมื่อมีการตั้งค่าให้ใช้งาน Brotli Decompressor
Istio	1.31.0 ถึง 1.31.1, 1.30.0 ถึง 1.30.5, 1.29.0 ถึง 1.29.8	1.31.2, 1.30.6, 1.29.9	เมื่อเปิดใช้งานการคลายการบีบอัดข้อมูล Brotli
Python Brotli	1.2.0 และเวอร์ชันก่อนหน้า	1.2.1	เมื่อติดตั้งแพ็คเกจและ HTTP Client มีการคลายการบีบอัดข้อมูล br
Python brotlicffi	1.2.0.x และเวอร์ชันก่อนหน้า	1.2.1.0	เมื่อติดตั้งแพ็คเกจและ HTTP Client มีการคลายการบีบอัดข้อมูล br
cdxgen	Standalone Binary และ Image เวอร์ชัน 13.4.0, 12.8.8 และเวอร์ชันก่อนหน้า	13.5.0, 12.8.9	เมื่อใช้ Standalone Binary หรือ Image สแกนข้อมูลจากแหล่งที่ไม่น่าเชื่อถือ

โปรแกรมจะมีความเสี่ยงต่อการโจมตีก็ต่อเมื่อมีการคลายการบีบอัดข้อมูล Brotli ที่ผู้โจมตีสามารถควบคุมได้ สำหรับ Node.js และ Bun ความเสี่ยงนี้อาจเกิดขึ้นเมื่อฟังก์ชัน fetch() รับข้อมูลตอบกลับที่ระบุ Content-Encoding: br หรือเมื่อส่วนประกอบที่จัดการ Request Body เช่น Express 5 และ body-parser 2 ทำการคลายการบีบอัดข้อมูล Brotli ที่ถูกส่งเข้ามา อย่างไรก็ตาม HTTP Server ทั่วไปที่สร้างด้วย Node.js จะไม่คลายการบีบอัด Request Body โดยอัตโนมัติ

สำหรับ Envoy และ Istio จะได้รับผลกระทบเฉพาะกรณีที่เปิดใช้งานการคลายการบีบอัดข้อมูล Brotli เช่น ผ่าน Envoy Filter หากไม่ได้เปิดใช้งานฟังก์ชันดังกล่าว ผู้โจมตีจะไม่สามารถเข้าถึงช่องโหว่ผ่านเส้นทางดังกล่าวได้

ในส่วนของ Python ความเสี่ยงเกิดจากการติดตั้งแพ็คเกจ Brotli หรือ brotlicffi เนื่องจากเมื่อมีแพ็คเกจใดแพ็คเกจหนึ่งอยู่ในระบบ HTTP Client เช่น requests, urllib3, httpx และ aiohttp จะสามารถคลายการบีบอัดข้อมูล Brotli ที่ได้รับมาได้โดยอัตโนมัติ ทำให้แอปพลิเคชันอาจได้รับผลกระทบ แม้จะไม่ได้เรียกใช้งาน Brotli โดยตรงในโค้ดก็ตาม

การใช้ Web Application Firewall (WAF) สามารถช่วยป้องกันได้เพียงบางส่วน โดย Vercel ซึ่งเพิ่มกฎ Firewall ในวันที่มีการเปิดเผยช่องโหว่ ระบุว่ากฎดังกล่าวไม่ครอบคลุมข้อมูลตอบกลับที่ฟังก์ชันของแอปพลิเคชันดึงมาจากเซิร์ฟเวอร์ภายนอก ดังนั้น การป้องกันการโจมตีผ่านช่องทางนี้จึงจำเป็นต้องอัปเดต Runtime หรือ Library ที่ได้รับผลกระทบ

ระดับความรุนแรงของช่องโหว่ขึ้นอยู่กับซอฟต์แวร์และรูปแบบการใช้งาน โดยทีม Red Team ของ Imperva ซึ่งตั้งชื่อการวิจัยครั้งนี้ว่า Node2Shell สามารถรันโค้ดที่ผู้โจมตีเตรียมไว้บน Node.js เวอร์ชัน 24.22.0 สำหรับ Linux ARM64 ที่ไม่ได้มีการดัดแปลงได้สำเร็จ ทั้งผ่านข้อมูลตอบกลับที่โปรแกรมดึงมาจากเซิร์ฟเวอร์ และผ่าน Request Body ของ Express อย่างไรก็ตาม การโจมตีต้องทดลองหลายครั้งจึงจะสำเร็จ โดยยังไม่มีเผยแพร่โค้ด Exploit และยังไม่ได้ทดสอบกับ Node.js เวอร์ชันอื่น

OpenAI ซึ่งเป็นผู้รายงานช่องโหว่ผ่านทีม Codex Security อธิบายว่าช่องโหว่ดังกล่าวอาจเปิดโอกาสให้ผู้โจมตีรันโค้ดจากระยะไกลโดยไม่ต้องยืนยันตัวตน (Unauthenticated Remote Code Execution) ใน Brotli และระบุว่า Node.js, Bun และ Envoy เป็นซอฟต์แวร์ที่ได้รับผลกระทบ โดยการโจมตีสามารถเกิดขึ้นได้ทั้งสองรูปแบบ ได้แก่

- เซิร์ฟเวอร์ที่ยังไม่ได้รับการแก้ไขและยอมรับข้อมูลบีบอัดจากภายนอก อาจถูกผู้โจมตีเข้าควบคุมผ่าน Client ที่ส่งข้อมูลอันตรายเข้ามา
- Client ที่ยังไม่ได้รับการแก้ไขและยอมรับข้อมูลบีบอัด เช่น การใช้งาน "curl --compressed" อาจถูกเข้าควบคุมผ่านเซิร์ฟเวอร์ที่ส่งข้อมูลอันตรายกลับมา

ขณะนี้ยังไม่มีรายงานการนำช่องโหว่นี้ไปใช้โจมตีจริง โดยผลการประเมินของ CISA เมื่อวันที่ 23 ตุลาคม ระบุว่ายังไม่พบการใช้ประโยชน์จากช่องโหว่ (Exploitation: None) และยังไม่พบการโจมตีแบบอัตโนมัติ (Automation: No) ขณะที่ Red Hat ระบุว่ายังไม่มีโค้ด Proof-of-Concept (PoC) ที่เผยแพร่ต่อสาธารณะและสามารถใช้รันโค้ดได้อย่างน่าเชื่อถือ

การประเมินระดับความรุนแรงแตกต่างกันไปตามแต่ละองค์กร โดย Google ซึ่งเป็นผู้กำหนดรหัส CVE ให้คะแนนความรุนแรง 9.2 ระดับ Critical ตามมาตรฐาน CVSS 4.0 ขณะที่ Node.js จัดให้อยู่ในระดับ High ส่วน Red Hat ให้คะแนน 8.1 ระดับ Important และ Microsoft ให้คะแนน 8.1 ระดับ High สำหรับ .NET

ที่น่าสนใจคือ การแก้ไขช่องโหว่นี้มีประวัติที่ค่อนข้างแตกต่างจากช่องโหว่ทั่วไป เนื่องจากโค้ดที่ใช้แก้ไขปัญหาก็ถูกเพิ่มเข้าไปในซอร์สโค้ดของ Brotli ตั้งแต่วันที่ 27 กุมภาพันธ์ แต่ยังไม่มีการออก Brotli เวอร์ชันใหม่ที่มีการแก้ไขดังกล่าว จนกระทั่งวันที่ 21 ตุลาคม

OpenAI อธิบายว่า สาเหตุของช่องโหว่นี้ได้รับการแก้ไขในซอร์สโค้ดของ Brotli ไปแล้วตั้งแต่เดือนกุมภาพันธ์ แต่ในขณะนั้นยังไม่มีใครทราบว่าเป็นช่องโหว่ด้านความปลอดภัย และยังไม่มีการเผยแพร่ Brotli เวอร์ชันใหม่หลังจากนั้น

Node.js ได้กล่าวขอบคุณ Khai Tran และ Sylvie Mayer จากทีม Codex Security ของ OpenAI ที่รายงานช่องโหว่ รวมถึง Matteo Collina และ Antoine du Hamel ซึ่งเป็นผู้ดูแลโครงการที่มีส่วนร่วมในการแก้ไขปัญหา

แนวทางแก้ไขและป้องกัน

ผู้ดูแลระบบควรอัปเดตซอฟต์แวร์เป็นเวอร์ชันที่ได้รับการแก้ไขตามตารางข้างต้น จากนั้นรีสตาร์ทบริการและสร้าง Container Image ใหม่หากมีการใช้งานผ่าน Container เนื่องจากโปรเซสที่กำลังทำงานอยู่จะยังคงใช้ Brotli Decoder เวอร์ชันเดิม จนกว่าจะมีการรีสตาร์ท

หากยังไม่สามารถอัปเดตได้ ควรหยุดการคลายการบีบอัดข้อมูล Brotli จากแหล่งที่ไม่น่าเชื่อถือ โดย OpenAI และบริษัทด้านความปลอดภัย Aikido ได้แนะนำแนวทางดังต่อไปนี้

- ปฏิเสธคำขอที่ส่งเข้ามาพร้อม Content-Encoding: br ก่อนที่ Proxy หรือส่วนประกอบที่ประมวลผลข้อมูลจะเริ่มคลายการบีบอัด Request Body
- ปิดการคลายการบีบอัด Brotli แบบอัตโนมัติใน HTTP Client และปฏิเสธข้อมูลตอบกลับแบบ Brotli ก่อนเข้าสู่กระบวนการคลายการบีบอัด
- สำหรับ Python ให้ถอนการติดตั้งแพ็คเกจ Brotli และ brotlicffi เพื่อให้ HTTP Client เปลี่ยนไปใช้การบีบอัดข้อมูลแบบ gzip และ deflate แทน

อย่างไรก็ตาม การลบ br ออกจากส่วนหัว Accept-Encoding เพียงอย่างเดียวไม่สามารถป้องกันช่องโหว่นี้ได้ เนื่องจากเซิร์ฟเวอร์ยังสามารถส่งข้อมูล Brotli กลับมาได้ แม้ Client จะไม่ได้ออกข้อผิดพลาด สำหรับ Node.js ยังไม่มีการเผยแพร่วิธีแก้ไข ปัญหาชั่วคราวอย่างเป็นทางการ ดังนั้น การอัปเดต Runtime จึงเป็นวิธีแก้ไขที่จำเป็น

แพ็คเกจของระบบปฏิบัติการ Linux แต่ละรายกำลังทยอยได้รับการแก้ไขในช่วงเวลาที่แตกต่างกัน โดย Debian ได้แก้ไขแพ็คเกจ brotli สำหรับ Debian 13 (trixie) เมื่อวันที่ 23 ตุลาคม ในเวอร์ชัน 1.1.0-2+deb13u1 และต่อมาได้แก้ไข Debian 12 (bookworm) ในเวอร์ชัน 1.0.9-2+deb12u1

ส่วน Amazon Linux 2023 ซึ่งก่อนหน้านี้เมื่อวันที่ 22 ตุลาคม ยังระบุว่าแพ็กเกจ brotli และ Node.js 24 อยู่ระหว่างรอการแก้ไข ได้เผยแพร่เวอร์ชันที่แก้ไขแล้วสำหรับทั้งสองแพ็กเกจเมื่อวันที่ 24 ตุลาคม

ขณะที่ openSUSE Tumbleweed ได้แก้ไขแพ็กเกจ brotli และ Node.js 24 แล้ว ส่วนการอัปเดต Node.js 22 สำหรับ SUSE Linux Enterprise 15 ยังอยู่ระหว่างการทดสอบ ณ วันที่ 26 ตุลาคม และ Ubuntu ยังคงระบุว่าแพ็กเกจ brotli และ nodejs ในทุกเวอร์ชันที่อยู่ในรายการจำเป็นต้องได้รับการประเมินเพิ่มเติม

นอกจากนี้ เครื่องมือสแกนช่องโหว่ใน Dependency อาจยังตรวจไม่พบแพ็กเกจ Python ที่ได้รับผลกระทบ เนื่องจากข้อมูล CVE ระบุถึง Repository ของ Google Brotli แต่ไม่ได้รับแพ็กเกจบน PyPI โดยตรง อีกทั้งข้อมูลคำแนะนำด้านความปลอดภัยของ GitHub ยังไม่ได้แสดงรายการแพ็กเกจที่ได้รับผลกระทบ ซึ่งเป็นข้อมูลที่ Dependabot ใช้ในการตรวจสอบ

อย่างไรก็ตาม มีการเพิ่มข้อมูลคำแนะนำด้านความปลอดภัยสำหรับแพ็กเกจ Python ทั้งสองรายการ ได้แก่ PYSEC-2026-4322 และ PYSEC-2026-4323 เมื่อวันที่ 23 ตุลาคม ดังนั้น ผู้ดูแลระบบจึงควรตรวจสอบเวอร์ชันของแพ็กเกจที่ติดตั้งอยู่ด้วยตนเองเพิ่มเติม

อัปเดต: Microsoft ออกแพตช์แก้ไข .NET พร้อมกับซอฟต์แวร์อื่นที่ทยอยได้รับการแก้ไข

Microsoft ได้เผยแพร่การอัปเดตด้านความปลอดภัยนอกกำหนดการ (Out-of-Band Update) เมื่อวันที่ 25 ตุลาคม เพื่อแก้ไขช่องโหว่ใน .NET 8, .NET 9 และ .NET 10 รวมถึง .NET 11 Release Candidate 3 โดยเวอร์ชัน Runtime ที่ได้รับการแก้ไขแล้ว ได้แก่ 8.0.33, 9.0.22 และ 10.0.14 ขณะที่ Microsoft ยืนยันว่า .NET Framework ไม่ได้รับผลกระทบ

แอปพลิเคชัน .NET จะมีความเสี่ยงก็ต่อเมื่อมีการคลายการบีบอัดข้อมูล Brotli ที่ผู้โจมตีสามารถควบคุมได้ เช่น การเปิดใช้งาน Brotli ใน ASP.NET Core Request Decompression หรือการเปิดใช้งานการคลายการบีบอัดข้อมูลตอบกลับแบบอัตโนมัติใน HttpClient ซึ่งทั้งสองฟังก์ชันไม่ได้เปิดใช้งานเป็นค่าเริ่มต้น

นอกจากนี้ ความเสี่ยงยังอาจเกิดขึ้นเมื่อแอปพลิเคชันส่งข้อมูลจากแหล่งที่ไม่น่าเชื่อถือไปยัง Brotli API เช่น BrotliStream และ BrotliDecoder โดย Microsoft ระบุว่า การติดตั้งแพ็กเกจที่มีช่องโหว่เพียงอย่างเดียวไม่ได้หมายความว่าแอปพลิเคชันจะสามารถถูกโจมตีได้ทันที

Microsoft อธิบายว่าช่องโหว่นี้อาจเปิดโอกาสให้ผู้โจมตีที่ไม่จำเป็นต้องยืนยันตัวตนสามารถอ่านหรือแก้ไขข้อมูลในหน่วยความจำของโปรเซสได้ ซึ่งอาจนำไปสู่การเปิดเผยข้อมูล การทำให้บริการหยุดทำงาน (Denial of Service) หรือการยกระดับสิทธิ์ (Elevation of Privilege) อย่างไรก็ตาม คำแนะนำด้านความปลอดภัยของ Microsoft ไม่ได้ระบุการรั่วไหลจากระยะไกลเป็นหนึ่งในผลกระทบของช่องโหว่นี้

สำหรับแอปพลิเคชันที่ใช้ Shared Runtime การติดตั้ง Runtime เวอร์ชันใหม่และรีสตาร์ทแอปพลิเคชันก็เพียงพอแล้ว แต่หากเป็นแอปพลิเคชันแบบ Self-contained จะต้อง Build และเผยแพร่แอปพลิเคชันใหม่ เช่นเดียวกับแอปพลิเคชัน NativeAOT เนื่องจาก Brotli Decoder ถูกฝังรวมอยู่ในไฟล์โปรแกรมโดยตรง ส่วน Container Image จะต้องสร้างใหม่โดยใช้ .NET Image ที่ได้รับการอัปเดตแล้ว

ในกรณีที่ยังไม่สามารถอัปเดตแอปพลิเคชันได้ Microsoft แนะนำให้ลบ br Provider ออกจากการตั้งค่า Request Decompression ที่ใช้งานอยู่ และนำ DecompressionMethods.Brotli ออกจาก HttpClient Handler ที่มีการตั้งค่าไว้ Microsoft ยังเตือนว่าไม่ควรเพิ่มการใช้งานฟังก์ชันคลายการบีบอัดข้อมูลให้กับแอปพลิเคชันที่เดิมไม่ได้ใช้งาน เนื่องจากอาจสร้างความเสี่ยงใหม่ที่ทำให้บริการหยุดทำงานจากการโจมตีแบบ Denial of Service ได้

การอัปเดตครั้งเดียวกันนี้ยังแก้ไขช่องโหว่ CVE-2026-114148 ซึ่งเป็นช่องโหว่ที่ไม่เกี่ยวข้องกับ Brotli โดยเกิดจากการกำหนดเจ้าของไฟล์ไม่ถูกต้องในไฟล์ .NET Tarball บางรายการสำหรับ Linux ซึ่งอาจเปิดโอกาสให้ผู้โจมตีที่เข้าถึงระบบภายในสามารถยกระดับสิทธิ์ของตนเองได้

ขณะเดียวกัน brotli gem สำหรับ Ruby ได้แก้ไขช่องโหว่ในเวอร์ชัน 0.9.1 เมื่อวันที่ 24 ตุลาคม โดยเวอร์ชันดังกล่าวรวม Brotli 1.2.1 ไว้ภายใน และจะไม่ใช้งาน Brotli Library ของระบบหากยังไม่ได้รับการแก้ไข แต่จะเปลี่ยนไปใช้ Library ที่มาพร้อมกับตัวเองแทน

Cloudflare ได้เพิ่มกฎสำหรับตรวจจับช่องโหว่นี้จำนวน 2 กฎ ลงใน Managed Firewall Ruleset เมื่อวันที่ 25 ตุลาคม โดยหนึ่งในกฎดังกล่าวใช้ตรวจจับคำขอที่ส่งเข้ามาพร้อม Content-Encoding: br

อย่างไรก็ตาม กฎทั้งสองถูกตั้งค่าเริ่มต้นให้บันทึกข้อมูลการรับส่งที่ตรงตามเงื่อนไขเท่านั้น และจะไม่มีกรบล็อกคำขอใดๆ จนกว่าลูกค้าจะเปลี่ยนการตั้งค่าการดำเนินการด้วยตนเอง

อัปเดต: JFrog พบช่องโหว่เดียวกันใน FFmpeg, ClickHouse และ Apache Arrow

นักวิจัยจาก JFrog รายงานเมื่อวันที่ 25 ตุลาคมว่า ช่องโหว่นี้ยังส่งผลกระทบต่อ FFmpeg, ClickHouse และ Apache Arrow โดยบางกรณีสามารถเริ่มต้นการโจมตีผ่านไฟล์ได้โดยตรง ไม่จำเป็นต้องอาศัยคำขอผ่านเว็บไซต์ ดังนั้น การปฏิเสธ Content-Encoding: br ที่จุดรับส่งข้อมูลเครือข่ายจึงไม่สามารถป้องกันการโจมตีผ่านไฟล์เหล่านี้ได้

สำหรับ FFmpeg ช่องทางการโจมตีเกี่ยวข้องกับ libjxl ซึ่งเป็น Library สำหรับจัดการภาพ JPEG XL โดยใช้ Brotli ในการคลายการบีบอัด Metadata ที่อยู่ภายในไฟล์ภาพ JPEG XL

นักวิจัยจาก JFrog พบว่าไฟล์ภาพที่ถูกสร้างขึ้นเป็นพิเศษสามารถทำให้ผู้โจมตีควบคุม Instruction Pointer ของโปรแกรมได้ ซึ่งเป็นส่วนที่กำหนดว่าโปรแกรมจะประมวลผลคำสั่งใดต่อไป อย่างไรก็ตาม JFrog ระบุว่าการพัฒนาการโจมตีให้สามารถรันโค้ดได้จริงยังต้องดำเนินการเพิ่มเติม

JFrog ยังไม่ได้รายงานว่ามีแพตช์ที่ออกมาเพื่อแก้ไข FFmpeg โดยเฉพาะ เนื่องจากช่องโหว่อยู่ใน Brotli Library ที่ libjxl นำมาใช้งาน ดังนั้น การอัปเดต Library ดังกล่าวเป็นเวอร์ชันที่ได้รับการแก้ไขแล้วจึงสามารถปิดช่องทางการโจมตีนี้ได้

ในส่วน ClickHouse ซึ่งเป็นระบบฐานข้อมูลแบบ Open Source มีการใช้ Brotli เพื่อคลายการบีบอัดข้อมูลจาก HTTP Request Body ที่ส่งมาพร้อม Content-Encoding: br รวมถึงไฟล์ .br และข้อมูล Parquet ตามรายละเอียดใน Pull Request ที่ใช้แก้ไขช่องโหว่นี้

นักวิจัยจาก JFrog สามารถทำให้เซิร์ฟเวอร์ ClickHouse หยุดทำงานได้ด้วยการส่งคำขอที่ถูกสร้างขึ้นเป็นพิเศษเพียงครั้งเดียว โดยมีขนาดประมาณ 23 MB ซึ่งถือเป็นการโจมตีแบบ Denial of Service (DoS) แต่ไม่ได้มีการยืนยันว่าสามารถรันโค้ดบนระบบได้

JFrog ระบุว่าในระยะเวลาที่เผยแพร่ผลการวิจัย ยังไม่มี ClickHouse เวอร์ชันที่ได้รับการแก้ไขออกมาอย่างเป็นทางการ อย่างไรก็ตาม เมื่อวันที่ 26 ตุลาคม มีการรวมการแก้ไขย้อนหลัง (Backport) ที่อัปเดต Brotli ซึ่งมาพร้อมกับ ClickHouse เป็นเวอร์ชัน 1.2.1 เข้าไปใน Branch 26.9 แล้ว

สำหรับ Apache Arrow พบว่า C++ Library ของโครงการมีการรวม Brotli เวอร์ชัน 1.0.9 ไว้สำหรับการ Build โปรแกรม ตามข้อมูลจาก Issue ที่เปิดไว้ในระบบติดตามปัญหาของโครงการ ซึ่ง ณ วันที่ 26 ตุลาคม ยังไม่มีการเชื่อมโยงไปยังแพตช์แก้ไขหรือการตอบกลับจากผู้ดูแลโครงการ

ข้อมูลใน Issue ระบุว่า การอ่านไฟล์ Parquet หรือ Arrow IPC ที่มีข้อมูลบีบอัดด้วย Brotli อาจทำให้เข้าถึงช่องโหว่นี้ได้ อย่างไรก็ตาม โปรแกรมที่ Build โดยเชื่อมโยงกับ Brotli Library ของระบบซึ่งได้รับการแก้ไขแล้วจะไม่ได้รับผลกระทบ

ในส่วนของ pyarrow ซึ่งเป็นแพ็คเกจสำหรับ Python นักวิจัยจาก JFrog พบว่าการส่งข้อมูลขนาดเพียง 534 ไบต์เข้าสู่กระบวนการคลายการบีบอัด Brotli สามารถทำให้โปรแกรมเข้าสู่การทำงานวนซ้ำไม่สิ้นสุด (Infinite Loop) และใช้ทรัพยากร CPU อย่างต่อเนื่อง ส่งผลให้เกิดการโจมตีแบบ Denial of Service แทนที่จะทำให้โปรแกรมหยุดทำงานจากข้อผิดพลาดโดยตรง

นอกจากนี้ JFrog ยังสามารถทำให้โปรแกรมทดสอบ .NET หยุดทำงานผ่าน Brotli API ได้เช่นกัน ซึ่งเป็นช่องทางการโจมตีที่ Microsoft ได้แก้ไขแล้วในการอัปเดตเมื่อวันที่ 25 ตุลาคม

(บทความนี้ได้รับการอัปเดตเพิ่มเติมเมื่อวันที่ 26 ตุลาคม หลังจากเผยแพร่ครั้งแรก โดยเพิ่มข้อมูลเกี่ยวกับแพตช์แก้ไข .NET ของ Microsoft ผลการวิจัยของ JFrog เกี่ยวกับซอฟต์แวร์อื่นที่ได้รับผลกระทบ การแก้ไข brotli gem สำหรับ Ruby กฎ Firewall ของ Cloudflare และแพ็คเกจเวอร์ชันใหม่ของระบบปฏิบัติการ Linux หลายรายการ)

ข้อมูลอ้างอิง

Oct 24, 2026, By Swati Khandelwal

- <https://thehackernews.com/2026/10/brotli-decoder-flaw-behind-nodejs-rce.html>