La firma de ciberseguridad runZero dio a conocer siete vulnerabilidades en FatFs, una biblioteca de sistema de archivos de tamaño reducido que permite a los dispositivos leer y escribir en los formatos FAT y exFAT, ampliamente utilizados en memorias USB y tarjetas SD.

La relevancia de estas fallas radica en la enorme adopción de FatFs. Esta biblioteca forma parte del firmware de una gran cantidad de dispositivos, entre ellos cámaras de videovigilancia, drones, controladores industriales, billeteras de criptomonedas por hardware y numerosos equipos basados en sistemas operativos de tiempo real.

En los escenarios más graves, un atacante podría introducir una memoria USB, una tarjeta SD o incluso un archivo de actualización especialmente manipulado para provocar corrupción de memoria y conseguir la ejecución de código arbitrario en el dispositivo afectado.

Muchos equipos embebidos carecen de mecanismos avanzados de protección de memoria, comunes en computadoras y teléfonos inteligentes. Debido a ello, runZero señala que *«any physical access leads to a jailbreak.»* En otras palabras, un quiosco público, una cámara con ranura SD, un cajero automático o incluso una máquina de votación equipada con un puerto USB podrían quedar completamente comprometidos tras un breve acceso físico.

Cómo funcionan las vulnerabilidades

Las siete vulnerabilidades comparten un mismo principio de explotación. El dispositivo intenta procesar un volumen de almacenamiento o una imagen de firmware creada de manera maliciosa, y FatFs administra incorrectamente esos datos corruptos, dando lugar a diferentes tipos de fallos.

En conjunto, runZero clasificó estas vulnerabilidades con niveles de severidad que van de Media a Alta según la escala CVSS, sin que ninguna alcance la categoría de Crítica.

La vulnerabilidad de mayor impacto es CVE-2026-6682, con una puntuación CVSS de 7.6. Se trata de un desbordamiento de enteros durante el proceso de montaje de volúmenes FAT32. Un error matemático provoca que el sistema calcule un tamaño de archivo incorrecto, el cual

posteriormente es utilizado como una longitud válida de lectura. En dispositivos reales, este comportamiento puede derivar en corrupción de memoria y, potencialmente, en la ejecución de código malicioso.

Resumen de las siete vulnerabilidades

- CVE-2026-6682 (CVSS 7.6 – Alta): desbordamiento de enteros durante el montaje de volúmenes FAT32 que puede provocar corrupción de memoria y ejecución de código. También puede explotarse mediante determinados procesos de actualización de firmware, además del uso de medios físicos.
- CVE-2026-6687 (CVSS 7.6 – Alta): un campo correspondiente a la etiqueta de volumen en exFAT desborda un pequeño búfer, facilitando la corrupción de memoria.
- CVE-2026-6688 (CVSS 7.6 – Alta): nombres de archivo largos pueden provocar desbordamientos en el código adicional que numerosos proyectos implementan alrededor de FatFs, por ejemplo al copiar `fno.fname` mediante funciones como `strcpy` hacia un búfer de tamaño fijo. Esta vulnerabilidad no puede solucionarse únicamente modificando FatFs.
- CVE-2026-6685 (CVSS 6.1 – Media): un error de cálculo durante la administración de la caché en volúmenes fragmentados puede ocasionar corrupción silenciosa de datos.
- CVE-2026-6683 (CVSS 4.6 – Media): una división entre cero en el manejo de exFAT puede bloquear completamente el dispositivo. Si ocurre durante un proceso de actualización, incluso podría inutilizar permanentemente el hardware. También es alcanzable mediante ciertos archivos de actualización.
- CVE-2026-6686 (CVSS 4.6 – Media): extender un archivo más allá de su tamaño real puede exponer información residual perteneciente a archivos eliminados anteriormente.
- CVE-2026-6684 (CVSS 4.6 – Media): una tabla de particiones GPT manipulada puede provocar que el dispositivo quede bloqueado durante el montaje del volumen. Es la única vulnerabilidad corregida oficialmente en FatFs R0.16.

Un problema sin solución oficial

Uno de los aspectos más preocupantes del informe es la falta de respuesta por parte del responsable del proyecto.

FatFs es mantenido por un único desarrollador, y runZero asegura que intentó establecer contacto en múltiples ocasiones. Incluso solicitó la colaboración del centro japonés de coordinación de incidentes JPCERT/CC, sin obtener respuesta.

Según la compañía, actualmente no existe una corrección oficial para las vulnerabilidades relacionadas con corrupción de memoria, tampoco una lista de distribución de seguridad que permita informar a los desarrolladores afectados ni un mecanismo centralizado para alertar a los numerosos fabricantes que integran FatFs en sus productos.

Actualizar a la versión más reciente únicamente protege contra la vulnerabilidad relacionada con GPT, mientras que el resto de las fallas deberán ser corregidas individualmente por cada proveedor que utilice esta biblioteca.

Plataformas afectadas

runZero identificó múltiples plataformas que incorporan FatFs, entre ellas:

- Espressif ESP-IDF
- STMicroelectronics STM32Cube
- Zephyr
- MicroPython
- ArduPilot
- RT-Thread
- Mbed
- Samsung TizenRT
- SWUpdate

Como consecuencia, el impacto potencial alcanza una enorme variedad de dispositivos del Internet de las Cosas (IoT), equipos industriales, drones, sistemas embebidos y billeteras de criptomonedas.

Hasta la publicación del informe, el 1 de julio de 2026, no existían evidencias de ataques que explotaran estas vulnerabilidades. Sin embargo, el material necesario para desarrollar exploits ya está disponible públicamente. runZero publicó imágenes de disco de prueba, un entorno para validar las vulnerabilidades y un ejemplo completamente funcional basado en QEMU.

Recomendaciones para fabricantes y usuarios

Los desarrolladores que construyen firmware compatible con dispositivos FAT o exFAT deberían localizar la implementación de FatFs utilizada en sus productos, revisar cuidadosamente el código adicional que la rodea, analizar el tratamiento de nombres de archivo y tamaños de archivos, y preparar actualizaciones de seguridad lo antes posible.

Por su parte, las organizaciones y usuarios que administran dispositivos potencialmente afectados deberían considerar los puertos físicos y los mecanismos de actualización como superficies de ataque. Limitar quién puede conectar medios de almacenamiento externos y mantenerse atento a nuevas versiones de firmware distribuidas por los fabricantes son medidas recomendables para reducir el riesgo.

La inteligencia artificial acelera el descubrimiento de vulnerabilidades

runZero explicó que ya había realizado una auditoría manual de FatFs en 2017, encontrando muy pocos problemas relevantes.

Cuando el equipo volvió a revisar el proyecto en marzo de 2026, decidió apoyarse en herramientas de inteligencia artificial de uso cotidiano, entre ellas Visual Studio Code y

GitHub Copilot configurado en modo automático, además de unas pocas instrucciones sencillas.

El modelo de lenguaje fue capaz de desarrollar un fuzzer, es decir, una herramienta diseñada para suministrar datos malformados a una aplicación hasta provocar fallos inesperados. Gracias a este enfoque aparecieron vulnerabilidades que la revisión manual nunca había detectado y, además, fue posible comprobar que varias de ellas eran explotables.

Este hallazgo coincide con una tendencia cada vez más evidente. A finales de 2024, el agente Big Sleep de Google descubrió una vulnerabilidad real de corrupción de memoria en SQLite que había pasado inadvertida para las técnicas tradicionales de fuzzing.

Más recientemente, un agente autónomo basado en inteligencia artificial identificó 21 vulnerabilidades relacionadas con seguridad de memoria en FFmpeg, otra biblioteca escrita en C ampliamente integrada en múltiples productos.

Para runZero, la conclusión es clara: si una cadena de herramientas basada casi por completo en inteligencia artificial comercial puede encontrar este tipo de errores, también pueden hacerlo actores maliciosos. Mantener estas vulnerabilidades en secreto ya no representa una estrategia de protección efectiva.

Un largo camino para la corrección

La empresa anticipa que la distribución de parches llevará años en lugar de días.

Como antecedente menciona PixieFail, el conjunto de nueve vulnerabilidades descubiertas en 2024 dentro del código de arranque por red de EDK II, utilizado por numerosos fabricantes de computadoras y servidores. En ese caso, la aplicación de correcciones por parte de los proveedores fue lenta y prolongada.

FatFs presenta un panorama aún más complejo, ya que comparte una enorme presencia en la industria, pero carece de un proceso de mantenimiento activo y de un canal oficial de

respuesta ante incidentes de seguridad.

Por ahora, será importante observar dos aspectos: si el mantenedor de FatFs publica finalmente correcciones para estas vulnerabilidades y cuál será la respuesta de los grandes fabricantes que incorporan esta biblioteca en sus plataformas. Hasta que eso ocurra, es razonable asumir que numerosos dispositivos actualmente en funcionamiento continúan procesando medios de almacenamiento no confiables mediante código que todavía no dispone de una solución oficial.