Se ha demostrado un nuevo ataque de hardware denominado PACMAN contra los conjuntos de chips del procesador M1 de Apple, lo que podría armar a un actor malicioso con la capacidad de obtener la ejecución de código arbitrario en los sistemas macOS.

Aprovecha los *«ataques de ejecución especulativa para eludir un importante mecanismo de protección de la memoria, la autenticación de puntero ARM, una función de seguridad que se utiliza para hacer cumplir la integridad del puntero»,* dijeron los investigadores del MIT, Joseph Ravichandran, Weon Taek Na, Jay Lang y Mengjia Yan.

Lo que es más preocupante es que *«si bien los mecanismos de hardware utilizados por PACMAN no se pueden parchear con funciones de software, los errores de corrupción de la memoria pueden serlo»,* agregaron los investigadores.

La vulnerabilidad tiene su origen en los códigos de autenticación de punteros (PAC), una línea de defensa introducida en la arquitectura arm64e que tiene como objetivo detectar y proteger contra cambios inesperados en los punteros, objetos que almacenan una dirección de memoria, en la memoria.

Los PAC tienen como objetivo resolver un problema común en la seguridad del software, como las vulnerabilidades de corrupción de la memoria, que por lo general se aprovechan al sobrescribir datos de control en la memoria (es decir, punteros) para redirigir la ejecución del código a una ubicación arbitraria controlada por el atacante.

Aunque se han diseñado estrategias como la aleatorización del diseño del espacio de direcciones (ASLR) para aumentar la dificultad de realizar ataques de desbordamiento de búfer, el objetivo de los PAC es determinar la *«validez de los punteros con un tamaño mínimo y un impacto en el rendimiento»,* evitando de forma efectiva que un adversario cree punteros para su uso en un exploit.

Esto se logra protegiendo un puntero con un hash criptográfico, llamado Código de Autentificación de Puntero (PAC), para garantizar su integridad. Apple explica la autenticación de puntero de la siguiente forma:

> *«La autenticación de puntero funciona al ofrecer una instrucción de CPU especial para agregar una firma criptográfica, o PAC, a los bits de alto orden no utilizados de un puntero antes de almacenar el puntero. Otra instrucción elimina y autentica la firma después de leer el puntero de la memoria. Cualquier cambio en el valor almacenado entre la escritura y la lectura invalida la firma. La CPU interpreta la falla de autenticación como corrupción de la memoria y establece un bit de orden alto en el puntero, lo que hace que el puntero no sea válido y que la aplicación se bloquee».*

Pero PACMAN *«elimina la barrera principal para realizar ataques de secuestro de flujo de control en una plataforma protegida mediante autenticación de puntero»*. Combina la corrupción de la memoria y la ejecución especulativa para eludir la función de seguridad, filtrando *«resultados de verificación de PAC a través de canales laterales de microarquitectura sin causar fallas».*

El método de ataque, en pocas palabras, hace posible distinguir entre un PAC correcto y una hash incorrecto, lo que permite a un mal actor *«aplicar con fuerza bruta el valor de PAC correcto mientras suprime los bloqueos y construye un ataque de secuestro de flujo de control en un PA habilitado o sistema operativo».*

La prevención de fallas, por su parte, tiene éxito porque cada valor de PAC se adivina especulativamente al explotar un canal lateral basado en el tiempo por medio del búfer de búsqueda lateral de traducción (TLB) usando un ataque Prime+Probe.

Las vulnerabilidades de ejecución especulativa, como se observó en el caso de Spectre y Meltdown, arman la ejecución desordenada, una técnica que se utiliza para lograr una mejora en el rendimiento de los microprocesadores modernos al predecir la ruta más probable del flujo de ejecución de un programa.

Sin embargo, cabe mencionar que el modelo de amenaza supone que ya existe una vulnerabilidad de corrupción de memoria explotable en un programa víctima (kernel), que a

su vez, permite que el atacante sin privilegios (una aplicación maliciosa) inyecte código no autorizado en ciertas ubicaciones de memoria en el proceso de la víctima.

> *«Este ataque tiene implicaciones importantes para los diseñadores que buscan implementar futuros procesadores con autenticación de puntero, y tiene amplias implicaciones para la seguridad de las futuras primitivas de integridad del flujo de control»*, agregaron los investigadores.

> *«Queremos agradecer a los investigadores por su colaboración, ya que esta prueba de concepto avanza en nuestra comprensión de estas técnicas»,* dijo Apple agregando que PACMAN tiene un nivel de explotación en la naturaleza muy reducido.

> *«Según nuestro análisis, así como los detalles compartidos con nosotros por los investigadores, hemos concluido que este problema no representa un riesgo inmediato para nuestros usuarios y es insuficiente para eludir las protecciones de seguridad del sistema operativo por sí solo».*

La vulnerabilidad se hace eco de otra amenaza irreparable denominada M1RACLES (CVE-2021-30747) que salió a la luz el año pasado, que permite que dos o más aplicaciones maliciosas instaladas en la máquina creen un canal encubierto para intercambiar datos entre ellos, sin usar memoria, sockets, archivos o cualquier otra característica típica del sistema operativo.