Se ha revelado una vulnerabilidad de seguridad de alta gravedad en Docker Engine que podría permitir a un atacante eludir los plugins de autorización (AuthZ) en determinadas condiciones.

La vulnerabilidad, identificada como CVE-2026-34040 (puntuación CVSS: 8.8), se origina a partir de una corrección incompleta de CVE-2024-41110, una falla de máxima gravedad en el mismo componente que se dio a conocer en julio de 2024.

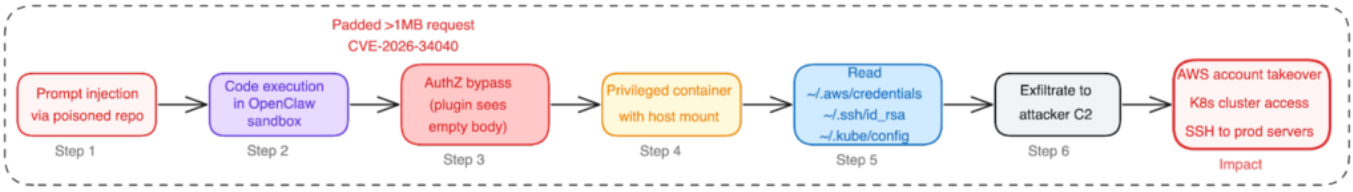
«Mediante una solicitud API especialmente diseñada, un atacante podría lograr que el demonio de Docker reenvíe la petición a un plugin de autorización sin incluir el cuerpo,» indicaron los responsables de Docker Engine en un aviso publicado a finales del mes pasado. *«El plugin de autorización podría aceptar una solicitud que normalmente habría rechazado si hubiera recibido el cuerpo completo.»*

«Cualquier persona que dependa de plugins de autorización que analicen el cuerpo de la solicitud para tomar decisiones de control de acceso podría verse afectada.»

Diversos investigadores de seguridad, entre ellos Asim Viladi Oglu Manizada, Cody, Oleh Konko y Vladimir Tokarev, han sido reconocidos por descubrir y reportar el fallo de forma independiente. El problema fue corregido en la versión 29.3.1 de Docker Engine.

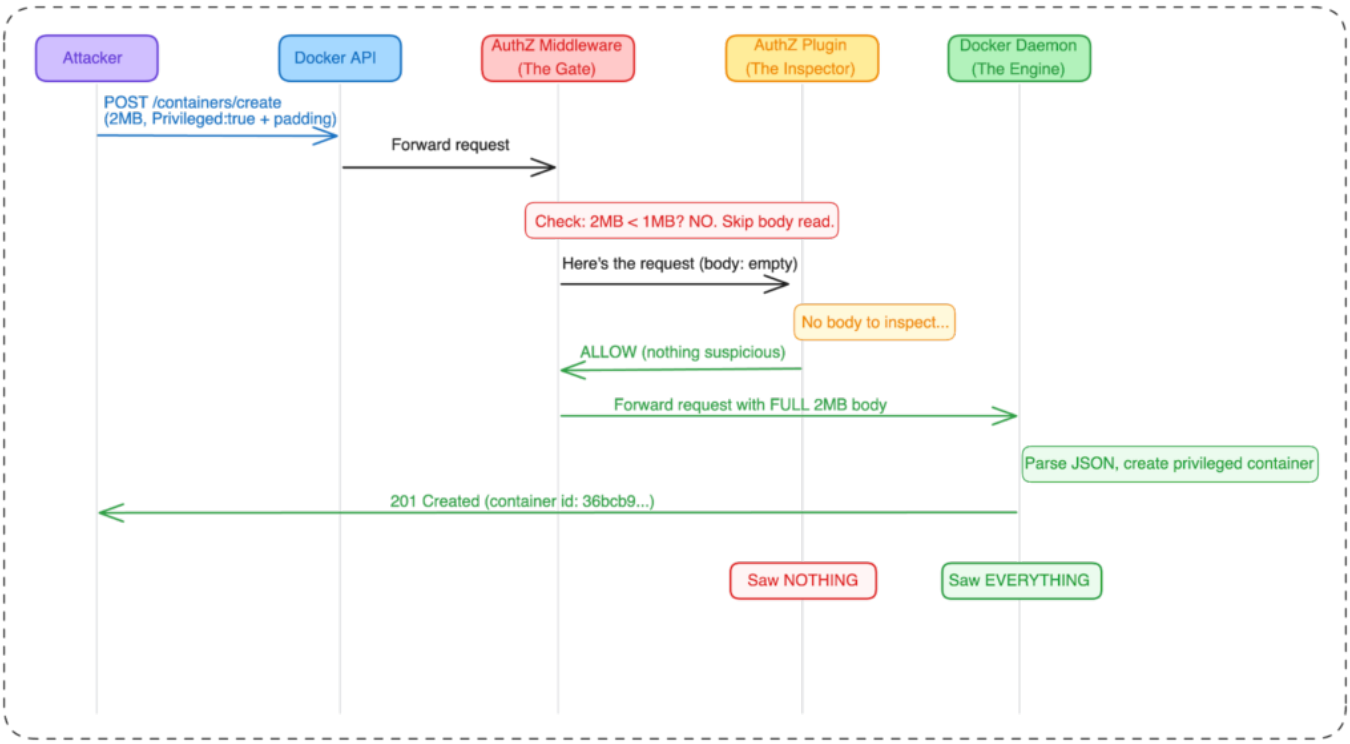
De acuerdo con un informe publicado por el investigador Tokarev de Cyera Research Labs, la vulnerabilidad se debe a que la solución aplicada a CVE-2024-41110 no gestionaba correctamente los cuerpos de solicitudes HTTP de gran tamaño, lo que abre la posibilidad de que una única solicitud HTTP con relleno adicional permita crear un contenedor con privilegios y acceso al sistema de archivos del host.

En un escenario de ataque hipotético, un actor malicioso con acceso a la API de Docker restringido por un plugin AuthZ podría evadir el mecanismo inflando una solicitud de creación de contenedor a más de 1 MB, provocando que esta sea descartada antes de llegar al plugin.



«El plugin permite la solicitud porque no detecta nada que bloquear,» explicó Tokarev en un informe. *«El demonio de Docker procesa la solicitud completa y crea un contenedor privilegiado con acceso root al host: credenciales de AWS, claves SSH, configuraciones de Kubernetes y cualquier otro recurso de la máquina. Esto funciona contra todos los plugins AuthZ del ecosistema.»*

Además, un agente de programación basado en inteligencia artificial como OpenClaw, ejecutándose dentro de un entorno aislado basado en Docker, puede ser manipulado para ejecutar una inyección de instrucciones oculta en un repositorio de GitHub especialmente preparado dentro de un flujo de trabajo normal de desarrollo. Esto puede derivar en la ejecución de código malicioso que explote CVE-2026-34040 para saltarse la autorización utilizando el método descrito y crear un contenedor privilegiado con acceso al sistema del host.



Con este nivel de acceso, el atacante podría extraer credenciales de servicios en la nube y utilizarlas para tomar control de cuentas cloud, clústeres de Kubernetes e incluso acceder por SSH a servidores en producción.

Y eso no es todo. Cyera también advirtió que los agentes de IA pueden descubrir por sí mismos cómo evadir esta protección y activarla generando una solicitud HTTP con relleno adicional al encontrar errores al intentar acceder a archivos como kubeconfig durante tareas legítimas de depuración (por ejemplo, resolver un problema de memoria en Kubernetes). Este enfoque elimina la necesidad de introducir un repositorio comprometido con instrucciones maliciosas.

«El plugin AuthZ rechazó la solicitud de montaje,» explicó Cyera. *«El agente tiene acceso a la API de Docker y comprende cómo funciona HTTP. CVE-2026-34040 no requiere código de explotación, privilegios ni herramientas especiales. Se trata de una única solicitud HTTP con relleno adicional. Cualquier agente capaz de leer la documentación de la API de Docker*

puede construirla.»

Como medidas temporales, se recomienda evitar el uso de plugins AuthZ que dependan del análisis del cuerpo de la solicitud para tomar decisiones de seguridad, restringir el acceso a la API de Docker únicamente a entidades confiables siguiendo el principio de privilegio mínimo, o ejecutar Docker en modo rootless.

«En modo rootless, incluso el usuario 'root' de un contenedor privilegiado se asigna a un UID sin privilegios en el host,» señaló Tokarev. *«El impacto potencial se reduce de un 'compromiso total del host' a 'compromiso de un usuario sin privilegios'. Para entornos que no puedan adoptar completamente el modo rootless, –userns-remap ofrece un mapeo de UID similar.»*