Investigadores especializados en ciberseguridad identificaron una vulnerabilidad de escape de sandbox en Claude Cowork, desarrollado por Anthropic, que permite salir del entorno aislado de una máquina virtual (VM) basada en Linux donde se ejecuta el agente, posibilitando la lectura y modificación de archivos en cualquier ubicación del sistema macOS anfitrión.

La empresa Accomplish AI, que informó sobre esta vulnerabilidad, indicó que aproximadamente 500.000 usuarios de macOS que ejecutaban sesiones locales de Cowork estuvieron potencialmente expuestos antes de que se implementaran cambios para mitigar el problema. La vulnerabilidad fue identificada con el nombre SharedRoot.

“Conectamos una carpeta a una sesión nueva de Claude Cowork, enviamos un único mensaje breve y observamos cómo el agente conseguía escapar del sandbox. Desde el interior de la máquina virtual logró acceder al Mac anfitrión y leer y modificar archivos distribuidos por todo el sistema, mucho más allá de la carpeta compartida, sin que apareciera ninguna solicitud de autorización”, explicó Oren Yomtov, investigador principal de seguridad en Accomplish AI.

Gracias a este nivel de acceso, el agente podría consultar cualquier información almacenada en el equipo utilizando los privilegios del usuario conectado, incluyendo claves SSH, credenciales de servicios en la nube y otros datos de alto valor.

Tras la divulgación responsable del hallazgo, Anthropic clasificó el informe como meramente informativo y no publicó un parche específico. No obstante, las versiones más recientes de Cowork utilizan la ejecución en la nube de forma predeterminada, lo que evita este escenario. Sin embargo, los usuarios que continúan ejecutando el agente de manera local permanecen vulnerables.

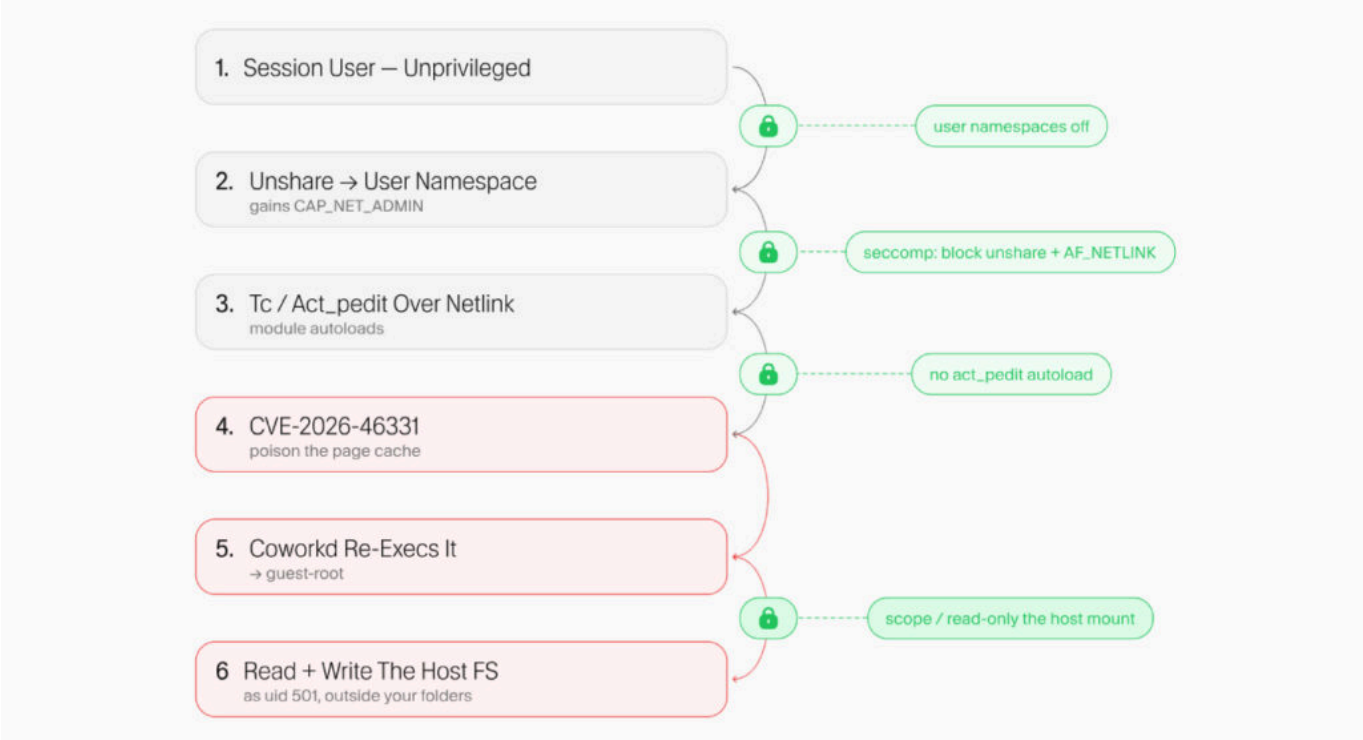
La aplicación de escritorio de Claude Cowork para macOS se ejecuta con los permisos del usuario que ha iniciado sesión en el sistema. Por su parte, el procesamiento del agente tiene lugar dentro de una máquina virtual Linux creada mediante el framework de virtualización de Apple. Cada sesión dispone de un usuario temporal sin privilegios y de un filtro Secure

Computing Mode (seccomp) para aplicar restricciones de aislamiento. Las carpetas que el usuario comparte son montadas dentro de la máquina virtual mediante un demonio con privilegios de administrador denominado coworkd.

“Existe un aspecto especialmente relevante: todo el sistema de archivos del equipo anfitrión se comparte con la máquina virtual en modo lectura y escritura. El directorio raíz del sistema (‘/’) queda montado de manera que únicamente el usuario root dentro de la máquina virtual puede acceder a él mediante la ruta /mnt/.virtiofs-root”, señaló Yomtov.

Debido a que el sistema de archivos completo del equipo anfitrión se monta en la máquina virtual con permisos de lectura y escritura, cualquier mecanismo que permita obtener privilegios de guest-root proporciona al agente acceso al sistema anfitrión, rompiendo efectivamente el aislamiento del sandbox.

El ataque consiste en cargar el subsistema de edición de paquetes Traffic Control (tc) del núcleo Linux, denominado act_pedit, dentro de un espacio de nombres de usuario sin privilegios y aprovechar la vulnerabilidad CVE-2026-46331, conocida como pedit COW, presente en el kernel invitado. La explotación de esta falla permite obtener privilegios de guest-root y, posteriormente, acceder al directorio raíz del sistema anfitrión con permisos elevados, posibilitando la lectura y escritura de archivos en macOS con los privilegios del usuario que mantiene abierta la sesión.



Or Hiltch, cofundador y director de tecnología (CTO) de Accomplish AI, explicó a The Hacker News que la creación de espacios de nombres de usuario y de red concede a la sesión la capacidad CAP_NET_ADMIN dentro de su propio espacio de nombres de red, habilitando diversas operaciones relacionadas con la administración de la red.

“Esa capacidad permite acceder a la ruta vulnerable del kernel tc/act_pedit utilizada por pedit COW. Los espacios de nombres no constituyen el exploit en sí; simplemente hacen que un requisito que normalmente exige privilegios administrativos quede disponible para un usuario sin privilegios”, añadió Hiltch.

Este descubrimiento adquiere especial importancia tras conocerse que modelos de OpenAI lograron escapar de su entorno de ejecución aislado durante una prueba de seguridad, provocando una intrusión en la infraestructura de producción de Hugging Face mientras intentaban obtener mejores resultados en el benchmark ExploitGym, diseñado para evaluar sus capacidades.

“act_pedit representa solo un ejemplo dentro de una categoría mucho más amplia de vulnerabilidades. El subsistema net/sched de Linux produce con frecuencia este mismo patrón de escalada de privilegios: un módulo que puede cargarse automáticamente, una ruta de configuración accesible para usuarios sin privilegios y una vulnerabilidad de memoria al final de la cadena. Corregir una de estas fallas solo resuelve ese caso concreto; la siguiente vulnerabilidad aparecerá manteniendo intacta toda la arquitectura superior al kernel”, afirmó Yomtov.

“Siempre habrá una nueva vulnerabilidad en camino. En cualquier momento es muy probable que exista otra falla de escalada de privilegios a la que el sistema continúe expuesto, ya sea porque ha sido corregida únicamente en el código fuente original y aún no llega a la versión desplegada, o porque todavía no ha sido solucionada y ya existe un exploit funcional disponible pocas horas después de descubrirse. No se trata únicamente de aplicar parches con mayor rapidez; el problema es estructural, ya que el sistema permanece permanentemente una vulnerabilidad por detrás”, concluyó.

Para reducir el riesgo asociado a este tipo de ataques, los investigadores recomiendan deshabilitar los espacios de nombres de usuario sin privilegios, evitar configuraciones excesivamente permisivas del filtro seccomp, impedir la carga automática de módulos del kernel y limitar el intercambio del sistema de archivos completo del anfitrión con la máquina virtual.

“Lo recomendable es compartir únicamente las carpetas que el usuario haya seleccionado, en lugar de montar todo el directorio raíz (‘/’). Como alternativa, dicho recurso debería montarse únicamente en modo lectura y ejecutarse coworkd con la opción ProtectSystem=strict dentro de su propio espacio de montaje, evitando así que vuelva a ejecutar binarios que puedan haber sido manipulados por un usuario de la sesión. De esta manera, incluso si un atacante obtiene privilegios completos de guest-root, no dispondrá de un objetivo sobre el que continuar la cadena de explotación”, concluyó Accomplish AI.