Alerta Vulnerabilidad
“Vulnerabilidad “Copy Fail” en Linux.

CSIRTSALUD-AV-20260504-44 **TLP: CLEAR**

Alerta ID:	044
Fecha del reporte:	04/05/2026
Entidad:	Todas las entidades del ecosistema digital
Título:	Vulnerabilidad “Copy Fail” en Linux.
Herramienta de detección	N/A
Activo involucrado:	Kernel de Linux
Tipo de alerta:	Alerta
Nivel de riesgo:	Alto

Objetivo:

Informar a las entidades del ecosistema digital sobre la vulnerabilidad **CVE-2026-31431**, conocida como “Copy Fail”, que afecta el kernel de Linux y permite a un atacante con acceso local escalar privilegios hasta obtener control como root en sistemas vulnerables. Esta vulnerabilidad representa un riesgo significativo para servidores Linux, entornos cloud, nodos Kubernetes, contenedores y plataformas CI/CD, debido a que puede ser aprovechada para comprometer la integridad del sistema, ejecutar acciones con privilegios elevados y facilitar movimientos laterales dentro de la infraestructura. La aplicación oportuna de las actualizaciones de seguridad y medidas de mitigación recomendadas es esencial para reducir la exposición, prevenir posibles explotaciones y garantizar la continuidad y seguridad de los servicios tecnológicos.

Alerta Vulnerabilidad
“Vulnerabilidad “Copy Fail” en Linux.

CSIRTSALUD-AV-20260504-44 **TLP: CLEAR**

Descripción:

Copy Fail es una vulnerabilidad de escalamiento local de privilegios en el subsistema criptográfico del kernel de Linux, específicamente asociada al módulo algif_aead de la interfaz criptográfica de usuario AF_ALG. Microsoft la clasifica como una vulnerabilidad de alta severidad que afecta distribuciones Linux ampliamente utilizadas, incluyendo Red Hat, SUSE, Ubuntu y Amazon Linux, con impacto relevante en cargas de trabajo cloud y clústeres Kubernetes.

La vulnerabilidad permite que un usuario local no privilegiado corrompa la caché de página del kernel sobre archivos legibles, incluidos binarios privilegiados tipo setuid, lo que puede derivar en ejecución de código con privilegios de root. El CVSS reportado por el CNA de kernel.org es 7.8 High, con vector local, baja complejidad, bajos privilegios requeridos y sin interacción del usuario.

Aunque no es una vulnerabilidad explotable remotamente por sí sola, su criticidad aumenta cuando se combina con un acceso inicial previo, por ejemplo, acceso SSH limitado, ejecución de código en un contenedor comprometido o ejecución de trabajos maliciosos en pipelines CI/CD.

Consecuencias de la explotación

La explotación exitosa de la vulnerabilidad **CVE-2026-31431** “Copy Fail” puede generar impactos significativos sobre la confidencialidad, integridad y disponibilidad de los sistemas Linux afectados. Aunque el vector de ataque es local, su criticidad aumenta cuando el atacante ya cuenta con algún nivel de acceso previo al sistema, por ejemplo, mediante una cuenta comprometida, una aplicación vulnerable, un contenedor expuesto o un proceso ejecutado dentro de una plataforma CI/CD.

Entre las principales consecuencias se encuentran:

1. Escalamiento de privilegios a root

Un atacante con permisos limitados podría elevar sus privilegios hasta obtener acceso como root, lo que le permitiría tomar control total del sistema operativo afectado. Con este nivel de acceso, el actor malicioso podría ejecutar comandos administrativos, modificar configuraciones críticas, manipular servicios del sistema, crear nuevas cuentas privilegiadas o desactivar controles de seguridad.

2. Compromiso total del sistema afectado

Una vez obtenidos privilegios elevados, el atacante podría alterar archivos del sistema, modificar binarios, instalar herramientas maliciosas, manipular registros de auditoría o implementar mecanismos de persistencia. Esto compromete la integridad del sistema y dificulta la identificación del alcance real del incidente.

3. Riesgo de movimiento lateral dentro de la infraestructura

El acceso como root puede facilitar la recolección de credenciales, llaves SSH, tokens de acceso, secretos de aplicaciones, variables de entorno y archivos de configuración. Esta información puede ser utilizada para desplazarse hacia otros servidores, bases de datos, plataformas cloud, repositorios de código o servicios internos de la organización.

4. Afectación de entornos contenerizados

En ambientes que utilizan contenedores, la vulnerabilidad representa un riesgo especial debido a que los contenedores comparten el kernel del host. Si un atacante logra ejecutar código dentro de un contenedor vulnerable o mal configurado, podría intentar aprovechar la falla para afectar el host subyacente, comprometer otros contenedores o acceder a recursos fuera del entorno inicialmente comprometido.

5. Impacto en clústeres Kubernetes

En clústeres Kubernetes, la explotación podría permitir que un atacante comprometa nodos de trabajo, acceda a secretos del clúster, manipule cargas de trabajo, afecte pods de otros namespaces o interfiera con servicios críticos. Esto puede derivar en un compromiso más amplio de la plataforma de orquestación y de las aplicaciones desplegadas sobre ella.

6. Exposición de información sensible

El atacante podría acceder a información almacenada en el sistema, incluyendo archivos de configuración, credenciales, certificados, llaves privadas, registros de aplicaciones, datos operativos o información sensible de usuarios. Este acceso no autorizado puede derivar en fuga de información, uso indebido de credenciales o exposición de datos protegidos.

7. Alteración o interrupción de servicios

Con privilegios administrativos, el atacante podría detener servicios, modificar parámetros del sistema, eliminar archivos críticos, interferir con procesos productivos o degradar el rendimiento del servidor. En escenarios críticos, esto podría ocasionar indisponibilidad de aplicaciones, interrupción de servicios institucionales o afectación de plataformas misionales.

8. Evasión de mecanismos de seguridad

El acceso privilegiado permitiría al atacante intentar desactivar herramientas de monitoreo, modificar logs, eliminar evidencias, detener agentes EDR/XDR, cambiar reglas de firewall o alterar políticas de seguridad. Esto incrementa la dificultad para detectar el incidente y reconstruir la línea de tiempo de la explotación.

9. Compromiso de plataformas CI/CD

En entornos de integración y despliegue continuo, un atacante podría aprovechar la vulnerabilidad desde un runner o agente vulnerable para acceder a secretos de despliegue, modificar pipelines, insertar código malicioso, alterar artefactos de compilación o comprometer ambientes productivos mediante la cadena de suministro de software.

10. Persistencia y control prolongado del entorno

Después de obtener privilegios elevados, el atacante podría establecer mecanismos de persistencia como usuarios ocultos, llaves SSH no autorizadas, tareas programadas, servicios maliciosos o modificaciones en scripts de inicio. Esto le permitiría mantener acceso al sistema incluso después de reinicios o cambios menores en la configuración.

11. Incremento del impacto regulatorio y reputacional

Si la explotación deriva en pérdida de información, interrupción de servicios o compromiso de datos sensibles, la organización podría enfrentar consecuencias regulatorias, contractuales, operativas y reputacionales. Esto es especialmente relevante para entidades que administran información crítica, datos personales o servicios esenciales.

12. Dificultad en la contención del incidente

Debido a que la vulnerabilidad permite elevar privilegios dentro del sistema, la respuesta ante incidentes debe considerar la posibilidad de compromiso total del host. En muchos casos, no basta con eliminar el archivo o proceso malicioso identificado; puede ser necesario aislar el sistema, preservar evidencia, rotar credenciales y reconstruir el servidor desde una imagen confiable.

La explotación de CVE-2026-31431 puede transformar un acceso local limitado en un compromiso completo del sistema afectado. Por esta razón, la vulnerabilidad debe ser priorizada en servidores Linux, nodos Kubernetes, entornos cloud, plataformas CI/CD y sistemas donde existan usuarios o cargas de trabajo no confiables.

Modo de explotación del ataque

La explotación de la vulnerabilidad CVE-2026-31431 "Copy Fail" se desarrolla como un ataque de escalamiento local de privilegios, es decir, requiere que el atacante ya cuente con algún nivel de acceso al sistema afectado. Este acceso inicial puede provenir de una cuenta limitada, una aplicación comprometida, un contenedor vulnerable, una sesión SSH con permisos bajos o un proceso ejecutado dentro de un entorno CI/CD. La vulnerabilidad afecta el subsistema criptográfico del kernel de Linux, específicamente el módulo algif_aead asociado a la interfaz criptográfica de usuario AF_ALG.

A continuación, se describe el modo de explotación por fases, en un nivel técnico defensivo y no operativo:

Fase 1. Obtención de acceso local limitado

El ataque inicia cuando el actor malicioso logra ejecutar código en el sistema Linux vulnerable con permisos bajos. Esta condición puede darse mediante el compromiso de una cuenta de usuario, la explotación previa de una aplicación expuesta, una mala configuración en un contenedor, un runner CI/CD comprometido o cualquier otro mecanismo que permita ejecución local.

En esta fase, el atacante todavía no tiene privilegios administrativos. Sin embargo, la vulnerabilidad es especialmente relevante porque el vector de ataque es local, requiere bajos privilegios y no necesita interacción del usuario.

Fase 2. Reconocimiento del entorno vulnerable

Una vez dentro del sistema, el atacante busca determinar si el host ejecuta una versión vulnerable del kernel de Linux y si el componente afectado está disponible. Para ello, puede revisar información general del sistema, versión del kernel, distribución Linux, presencia del módulo vulnerable y condiciones del entorno donde se ejecuta el proceso.

Esta fase es crítica porque la vulnerabilidad no afecta únicamente servidores tradicionales, sino también entornos cloud, nodos Kubernetes, hosts de contenedores y plataformas de integración continua que comparten el kernel del sistema operativo subyacente. Microsoft advierte que el impacto es relevante en cargas de trabajo cloud y Kubernetes.

Fase 3. Identificación de un objetivo privilegiado en el sistema

Después de confirmar que el sistema puede ser vulnerable, el atacante identifica componentes del sistema que, al ser manipulados en memoria, podrían permitir una elevación de privilegios. En este tipo de explotación, el objetivo suele estar relacionado con binarios o procesos que se ejecutan con permisos elevados, especialmente aquellos que pueden interactuar con privilegios de administrador.

El riesgo principal se presenta cuando la falla permite afectar la representación en memoria de archivos o componentes privilegiados sin modificar necesariamente el contenido persistente en disco. Esto dificulta la detección por controles tradicionales que se basan únicamente en verificar integridad de archivos almacenados.

Fase 4. Abuso del subsistema criptográfico del kernel

En esta fase, el atacante intenta abusar de la falla lógica ubicada en el módulo algif_aead, perteneciente a la API criptográfica de espacio de usuario AF_ALG. La vulnerabilidad se origina

en una optimización introducida años atrás, relacionada con operaciones "in-place", donde determinadas páginas de memoria podían ser tratadas indebidamente dentro de estructuras internas del kernel.

Desde una perspectiva defensiva, lo importante es comprender que el ataque no depende de una contraseña de administrador ni de una mala configuración puntual del usuario final, sino de una condición vulnerable dentro del kernel. Por eso, las medidas de mitigación deben centrarse en actualizar el kernel, restringir la exposición del componente afectado y endurecer los entornos donde se ejecutan cargas no confiables.

Fase 5. Manipulación de memoria o caché de página

El atacante aprovecha la falla para provocar una alteración controlada en estructuras de memoria asociadas al kernel, particularmente relacionadas con la caché de página. Esta manipulación puede permitir que ciertos datos se comporten de forma distinta a como deberían, generando una diferencia entre lo que existe en disco y lo que el sistema interpreta o ejecuta desde memoria.

Esta característica hace que la explotación sea especialmente delicada, porque la modificación puede no quedar reflejada como un cambio tradicional de archivo. En consecuencia, algunas herramientas de verificación basadas únicamente en hash, integridad de disco o comparación estática de archivos podrían no detectar el abuso de forma inmediata.

Fase 6. Alteración del flujo de ejecución privilegiado

Una vez manipulada la memoria o la representación temporal de un componente privilegiado, el atacante busca que el sistema ejecute instrucciones bajo un contexto de mayor privilegio. El objetivo final es transformar un proceso inicialmente limitado en una sesión o ejecución con permisos administrativos.

En términos prácticos, esta fase puede derivar en que un usuario sin privilegios obtenga capacidades equivalentes a root, lo que le permitiría controlar el sistema afectado. Microsoft clasifica la vulnerabilidad como una falla de escalamiento local de privilegios que puede permitir obtener privilegios elevados en distribuciones Linux afectadas.

Fase 7. Obtención de privilegios de root

Si la explotación es exitosa, el atacante consigue ejecutar acciones como root. En este punto, la barrera entre usuario limitado y administrador del sistema queda superada. El atacante puede crear usuarios, modificar permisos, acceder a archivos protegidos, alterar configuraciones, detener servicios, instalar herramientas maliciosas o manipular mecanismos de seguridad.

Esta fase representa el punto de mayor impacto técnico, porque el compromiso deja de limitarse al usuario inicial y pasa a afectar la totalidad del host Linux.

Fase 8. Consolidación del acceso

Después de obtener privilegios elevados, el atacante puede intentar mantener el control del sistema. Para ello, podría crear mecanismos de persistencia, modificar tareas programadas, agregar claves SSH no autorizadas, alterar servicios del sistema, instalar puertas traseras o manipular scripts de inicio.

En esta fase también puede intentar borrar rastros, modificar logs o desactivar herramientas de monitoreo. Por esta razón, ante sospecha de explotación, se recomienda tratar el sistema como potencialmente comprometido en su totalidad y no limitar la respuesta a eliminar un único proceso sospechoso.

Fase 9. Movimiento lateral y expansión del compromiso

Con privilegios de root, el atacante puede buscar credenciales, tokens, secretos de aplicaciones, claves privadas, archivos de configuración, variables de entorno y accesos almacenados en el sistema. Esta información puede ser utilizada para comprometer otros servidores, bases de datos, repositorios, servicios cloud, plataformas Kubernetes o sistemas internos.

En entornos Kubernetes y CI/CD, esta fase es especialmente crítica, porque un nodo o runner comprometido puede servir como punto de pivote hacia cargas de trabajo, secretos, pipelines de despliegue o ambientes productivos. Microsoft y CERT-EU destacan la importancia de priorizar la mitigación en entornos cloud, Kubernetes, contenedores y CI/CD.

Fase 10. Evasión, persistencia o sabotaje

Finalmente, el atacante puede intentar permanecer oculto, dificultar la investigación o afectar la disponibilidad del servicio. Con privilegios administrativos podría eliminar evidencias, modificar registros, detener agentes de seguridad, alterar reglas de firewall, manipular servicios críticos o preparar acciones destructivas.

En escenarios de mayor impacto, la explotación puede derivar en interrupción de servicios, fuga de información, compromiso de infraestructura crítica, manipulación de artefactos de software o afectación de la cadena de suministro tecnológica.

Vulnerabilidades asociadas

La vulnerabilidad principal asociada al presente boletín es CVE-2026-31431, conocida como "Copy Fail", una falla de escalamiento local de privilegios en el kernel de Linux. Aunque se registra bajo un único identificador CVE, su impacto se relaciona con varios

componentes críticos del sistema operativo, entre ellos el subsistema criptográfico del kernel, la interfaz AF_ALG, el módulo algif_aead, la caché de página y los binarios privilegiados tipo setuid.

Esta vulnerabilidad fue clasificada como de alta severidad, con una puntuación CVSS 3.1 de 7.8, debido a que permite que un usuario local sin privilegios pueda escalar hasta obtener permisos de root, sin requerir interacción del usuario y con baja complejidad de ataque. SUSE registra el vector como AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H, lo que indica impacto alto sobre confidencialidad, integridad y disponibilidad.

- **Vulnerabilidad principal identificada**

CVE	Nombre común	Severidad	Tipo de vulnerabilidad	Componente afectado	Impacto
CVE-2026-31431	Copy Fail	Alta — CVSS 7.8	Escalamiento local de privilegios	Kernel de Linux / algif_aead / AF_ALG	Permite obtener privilegios de root desde un usuario local sin privilegios

De acuerdo con NVD, la vulnerabilidad se encuentra en el kernel de Linux y está relacionada con la reversión de una operación "in-place" en crypto: algif_aead, debido a que no existía beneficio en operar de esa forma cuando el origen y destino provenían de diferentes asignaciones de memoria.

- **Componentes técnicos asociados**

Aunque el CVE corresponde a una sola vulnerabilidad, el riesgo se materializa mediante la interacción de varios componentes del sistema:

Componente	Relación con la vulnerabilidad	Riesgo asociado
Kernel de Linux	Es el componente central afectado. La falla reside en una lógica interna del kernel.	Compromiso completo del sistema operativo si se logra escalamiento a root.
algif_aead	Módulo relacionado con la API criptográfica del kernel expuesta al espacio de usuario.	Puede ser abusado para provocar una escritura indebida en memoria asociada a archivos legibles.
AF_ALG	Interfaz que permite a procesos de usuario acceder a funciones criptográficas del kernel.	Amplía la superficie de ataque cuando está disponible para usuarios o contenedores.
Caché de página del kernel	Representa en memoria contenido de archivos utilizados por el sistema.	Su manipulación puede alterar el comportamiento de binarios sin modificar directamente el archivo en disco.
Binarios setuid-root	Archivos ejecutables que pueden correr con privilegios elevados.	Si se altera su representación en memoria, pueden ser utilizados para obtener una shell con privilegios de root.
Contenedores y nodos Kubernetes	Comparten el kernel del host.	Un proceso dentro de un contenedor podría convertirse en punto de partida para comprometer el host si existen condiciones vulnerables.

Microsoft describe la falla como una vulnerabilidad en el subsistema criptográfico del kernel que permite corromper la caché de cualquier archivo legible, incluidos binarios setuid, lo que puede derivar en ejecución de código con privilegios de root.

- **Productos y distribuciones potencialmente afectadas**

La vulnerabilidad afecta principalmente a sistemas Linux que ejecutan kernels construidos desde 2017 y que no cuentan con las correcciones o mitigaciones correspondientes. CERT-EU indica que la falla impacta distribuciones Linux convencionales que integran kernels afectados y menciona como entornos verificados Ubuntu, Amazon Linux, RHEL y SUSE, además de otras distribuciones como Debian, Arch Linux, Fedora, Rocky Linux, AlmaLinux, Oracle Linux y distribuciones Linux embebidas.

Distribución / entorno	Estado de exposición
Ubuntu	Versiones anteriores a Ubuntu 26.04 pueden estar afectadas; Ubuntu informó mitigaciones mediante kmod para deshabilitar el módulo vulnerable mientras se liberan paquetes de kernel corregidos. (Ubuntu)
Amazon Linux 2023	Reportado dentro de los entornos afectados o verificados por análisis técnicos. (CERT-EU)
Red Hat Enterprise Linux — RHEL	Microsoft y CERT-EU lo incluyen dentro de las distribuciones relevantes afectadas o verificadas. (Microsoft)
SUSE Linux Enterprise	SUSE registra el CVE con severidad importante y CVSS 7.8. (SUSE)
Debian, Fedora, Arch Linux, Rocky Linux, AlmaLinux, Oracle Linux	CERT-EU las menciona como distribuciones implícitamente afectadas si ejecutan kernels dentro del rango vulnerable. (CERT-EU)
Kubernetes, Docker, LXC y hosts de contenedores	Riesgo elevado por compartir el kernel del host con cargas de trabajo potencialmente no confiables. (The Hacker News)
Runners CI/CD	Riesgo elevado si ejecutan jobs no confiables o código de terceros sobre kernels vulnerables. (CERT-EU)

- **Condiciones que aumentan el riesgo de explotación**

La vulnerabilidad no debe analizarse únicamente como una falla aislada del kernel. Su impacto aumenta cuando se combina con condiciones operativas comunes en entornos empresariales:

Condición asociada	Cómo aumenta el riesgo
Usuarios locales con permisos bajos	Cualquier cuenta comprometida o mal administrada puede convertirse en punto de partida para escalar privilegios.
Acceso SSH limitado	Un atacante que obtenga acceso a una cuenta no privilegiada podría intentar explotar la vulnerabilidad para obtener root.
Aplicaciones vulnerables con ejecución local	Una aplicación comprometida puede proporcionar el acceso inicial necesario para ejecutar código en el host.
Contenedores con cargas no confiables	Si el host comparte un kernel vulnerable, el contenedor puede ser usado como punto de entrada hacia el sistema base.
Runners CI/CD expuestos	Los pipelines pueden ejecutar código de repositorios externos, ramas no confiables o artefactos maliciosos.
Sistemas multiusuario	A mayor número de usuarios locales, mayor probabilidad de que una cuenta limitada sea utilizada para escalar privilegios.
Servidores sin actualización de kernel	La falta de parcheo mantiene expuesta la superficie vulnerable.
Ausencia de controles EDR/XDR en Linux	Reduce la capacidad de detectar explotación, elevación de privilegios o comportamiento anómalo.

Microsoft advierte que la vulnerabilidad no es remotamente explotable por sí sola, pero se vuelve altamente impactante cuando se encadena con un acceso inicial como SSH, ejecución maliciosa en CI/CD o presencia dentro de un contenedor

- **Relación con explotación activa y exposición pública**

La vulnerabilidad también está asociada a un riesgo elevado por la disponibilidad de pruebas de concepto públicas y reportes de explotación. CISA la incorporó a su catálogo de **Known Exploited Vulnerabilities — KEV**, con base en evidencia de explotación activa

Factor asociado	Relevancia
PoC pública disponible	Facilita que actores maliciosos repliquen o adapten la técnica de explotación.
Inclusión en CISA KEV	Indica que la vulnerabilidad ya presenta explotación conocida y debe ser priorizada.
Afectación amplia en Linux	Incrementa la probabilidad de encontrar sistemas vulnerables en infraestructuras empresariales.
Baja complejidad de explotación	Reduce la barrera técnica para actores con acceso local limitado.
Impacto en cloud y Kubernetes	Amplía el alcance del riesgo hacia entornos modernos de infraestructura.

- **Vulnerabilidades operativas relacionadas**

Además de la vulnerabilidad técnica principal, existen debilidades de configuración y operación que pueden facilitar su aprovechamiento:

Debilidad operativa	Riesgo generado
Falta de inventario de kernels Linux	Impide identificar rápidamente los sistemas expuestos.
Gestión deficiente de parches	Retrasa la aplicación de actualizaciones críticas.
Contenedores con privilegios excesivos	Aumenta el impacto de una explotación desde cargas contenerizadas.
Ausencia de políticas seccomp, AppArmor o SELinux	Reduce la capacidad de limitar llamadas o comportamientos peligrosos.
CI/CD sin aislamiento adecuado	Permite que código no confiable se ejecute sobre hosts críticos.
Cuentas locales innecesarias	Amplían la superficie de ataque para escalamiento de privilegios.
Monitoreo limitado de Linux	Dificulta detectar procesos que pasan de usuario bajo a root de forma anómala.

Estas condiciones no son CVE independientes, pero sí son **factores de exposición** que incrementan la probabilidad y el impacto de explotación de **CVE-2026-31431**.

- **Versiones corregidas y mitigaciones asociadas**

El parche principal revierte la optimización vulnerable introducida en el componente afectado. The Hacker News reporta que existen correcciones en versiones del kernel 6.18.22, 6.19.12 y 7.0, mientras que CERT-EU indica que el fix upstream corresponde al commit principal a664bf3d603d, orientado a revertir la optimización introducida en 2017

Acción	Propósito
Actualizar el kernel	Corregir la vulnerabilidad en el componente afectado.

Acción	Propósito
Reiniciar el sistema	Garantizar que el kernel corregido quede cargado en memoria.
Aplicar mitigaciones del proveedor	Reducir exposición mientras se despliega el parche definitivo.
Deshabilitar o bloquear el módulo vulnerable cuando aplique	Limitar el abuso de algif_aead / AF_ALG en sistemas donde sea viable.
Endurecer contenedores	Reducir el riesgo de explotación desde cargas no confiables.
Priorizar Kubernetes y CI/CD	Atender primero entornos donde se ejecuta código de múltiples fuentes.

Recomendaciones de detección y mitigación

Los equipos de seguridad, infraestructura, operaciones y administración de plataformas Linux deben priorizar la actualización inmediata de los sistemas afectados por la vulnerabilidad CVE-2026-31431 "Copy Fail", aplicando los parches de seguridad publicados por los fabricantes y distribuciones correspondientes. Esta actualización debe contemplar servidores físicos, máquinas virtuales, instancias cloud, nodos Kubernetes, hosts de contenedores, appliances basados en Linux, estaciones administrativas y runners de CI/CD que ejecuten versiones vulnerables del kernel.

Adicionalmente, se recomienda realizar un inventario técnico detallado de todos los activos Linux presentes en la infraestructura, identificando la versión del kernel instalada, la distribución utilizada, el nivel de exposición del sistema, los servicios publicados, los usuarios

locales existentes y la función que cumple cada servidor dentro de la operación. Este inventario debe permitir priorizar la remediación de los sistemas más críticos, especialmente aquellos que procesan información sensible, alojan servicios misionales, ejecutan cargas de trabajo compartidas o permiten acceso de usuarios no administrativos.

Los equipos técnicos deben verificar que la actualización del kernel haya sido aplicada correctamente y que el sistema haya sido reiniciado cuando corresponda, ya que en Linux la instalación del paquete corregido no siempre implica que el kernel seguro se encuentre cargado en memoria. Por esta razón, después de aplicar los parches, se debe validar la versión activa del kernel y confirmar que no permanezcan sistemas ejecutando versiones vulnerables, especialmente en ambientes de producción, contingencia, pruebas, desarrollo y laboratorios conectados a la red institucional.

En entornos contenerizados, se recomienda revisar de forma prioritaria los hosts que ejecutan Docker, containerd, CRI-O, LXC u otras tecnologías de virtualización a nivel de sistema operativo. Debido a que los contenedores comparten el kernel del host, una carga de trabajo comprometida podría incrementar el riesgo de explotación si el nodo subyacente permanece vulnerable. Por lo tanto, los administradores deben verificar tanto las imágenes base como los nodos donde se ejecutan los contenedores, evitando que cargas no confiables operen sobre kernels sin corregir.

Para plataformas Kubernetes, se recomienda validar todos los nodos worker y control plane, revisar los pools de nodos, las imágenes utilizadas por las cargas de trabajo, los permisos asignados a los pods y la exposición de workloads que ejecuten código de terceros o procesos no confiables. En caso de identificar nodos vulnerables, estos deben ser drenados, actualizados, reiniciados y reincorporados al clúster de manera controlada, garantizando que las aplicaciones críticas mantengan disponibilidad durante el proceso de mitigación.

En ambientes de integración y despliegue continuo, los equipos DevOps deben revisar los runners, agentes de compilación, servidores de automatización y pipelines que ejecuten código

proveniente de ramas, repositorios, dependencias o colaboradores externos. Estos entornos deben considerarse de alto riesgo, ya que un atacante podría aprovechar una ejecución limitada dentro de un pipeline para intentar escalar privilegios sobre el host vulnerable y acceder a secretos, tokens, credenciales de despliegue o artefactos de software.

Como medida de mitigación adicional, se recomienda endurecer las políticas de ejecución en contenedores y cargas de trabajo no confiables mediante controles como seccomp, AppArmor o SELinux, restringiendo llamadas al sistema y capacidades innecesarias. Cuando sea técnicamente viable y no afecte la operación, también se debe evaluar el bloqueo o deshabilitación temporal de componentes relacionados con AF_ALG o algif_aead, mientras se completa el proceso de actualización del kernel en todos los activos afectados.

Desde el punto de vista de detección, se recomienda fortalecer el monitoreo sobre eventos de escalamiento de privilegios, ejecución anómala de procesos con UID 0, creación inesperada de usuarios privilegiados, modificación de archivos sensibles, cambios en permisos de binarios críticos y actividad sospechosa en directorios del sistema. Estos eventos deben ser correlacionados con registros de autenticación, auditoría del sistema, logs de contenedores, alertas EDR/XDR y telemetría de plataformas cloud o Kubernetes.

Los equipos de seguridad deben prestar especial atención a comportamientos inusuales posteriores a la ejecución de procesos por usuarios sin privilegios, especialmente cuando dichos procesos deriven en shells administrativas, ejecución de comandos como root, acceso a archivos restringidos o modificación de configuraciones críticas. También se recomienda revisar intentos de desactivación de agentes de seguridad, limpieza de logs, cambios en reglas de firewall, alteración de servicios del sistema o instalación de mecanismos de persistencia.

Finalmente, se recomienda establecer una estrategia de validación posterior a la mitigación, que incluya verificación de versiones corregidas, revisión de alertas recientes, análisis de cuentas locales, validación de integridad de configuraciones críticas y rotación de credenciales en sistemas donde exista sospecha de explotación. En caso de detectar indicios de

compromiso, el host debe ser aislado, preservado para análisis forense y, preferiblemente, reconstruido desde una imagen confiable antes de ser reincorporado a la operación.

Recomendaciones de prevención y mitigación

Los equipos de seguridad, infraestructura, operaciones, DevOps y administración de plataformas Linux deben establecer como prioridad la actualización del kernel en todos los sistemas potencialmente afectados por la vulnerabilidad CVE-2026-31431 "Copy Fail". Esta falla afecta el subsistema criptográfico del kernel de Linux, específicamente el módulo algif_aead asociado a la API criptográfica de espacio de usuario AF_ALG, y permite el escalamiento local de privilegios hasta obtener acceso como root cuando el sistema ejecuta una versión vulnerable del kernel. La vulnerabilidad fue calificada con severidad alta, con CVSS 3.1 de 7.8, bajo el vector AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H.

La primera medida preventiva debe ser la identificación de todos los activos Linux dentro de la infraestructura institucional. Este inventario debe incluir servidores físicos, máquinas virtuales, instancias cloud, nodos Kubernetes, hosts de contenedores, appliances, estaciones administrativas, sistemas embebidos y runners de CI/CD. Microsoft señala que la vulnerabilidad impacta múltiples distribuciones Linux, incluidas Red Hat, SUSE, Ubuntu y AWS Linux, y que puede afectar de forma significativa cargas Linux en la nube y clústeres Kubernetes.

Una vez identificados los sistemas expuestos, se debe aplicar el parche de seguridad correspondiente según el proveedor de la distribución utilizada. La mitigación no debe limitarse a instalar paquetes, ya que en muchos sistemas Linux es necesario reiniciar el host para que el kernel corregido quede efectivamente cargado en memoria. Después de la actualización, se debe validar la versión activa del kernel, confirmar que el sistema dejó de ejecutar la versión vulnerable y documentar la remediación dentro del proceso de gestión de vulnerabilidades.

En los casos donde aún no exista un parche definitivo disponible para una distribución específica, se deben aplicar mitigaciones temporales. Microsoft recomienda identificar todos los productos y versiones afectados, aplicar parches cuando estén disponibles y, si no existen actualizaciones inmediatas, implementar medidas provisionales como deshabilitar la funcionalidad afectada, aplicar aislamiento de red y reforzar controles de acceso.

Adicionalmente, se recomienda evaluar la deshabilitación temporal del módulo vulnerable algif_aead cuando no sea indispensable para la operación del sistema. Ubuntu informó que su equipo de seguridad liberó mitigaciones mediante el paquete kmod para deshabilitar el módulo afectado, mientras que las correcciones definitivas se distribuyen a través de paquetes de imagen del kernel. Esta medida debe aplicarse con validación previa, ya que el módulo puede estar relacionado con funciones de criptografía acelerada por hardware y algunas aplicaciones podrían requerir reinicio o ajuste para operar correctamente.

En ambientes contenerizados, la prevención debe enfocarse en reducir la posibilidad de que cargas no confiables interactúen con superficies sensibles del kernel. Debido a que los contenedores comparten el kernel del host, un contenedor comprometido o mal configurado puede incrementar el riesgo operativo si el nodo subyacente permanece vulnerable. Por esta razón, se recomienda actualizar los hosts de contenedores, restringir contenedores privilegiados, eliminar capacidades innecesarias, aplicar perfiles seccomp, reforzar AppArmor o SELinux, limitar montajes sensibles del host y evitar la ejecución de imágenes no verificadas.

Para plataformas Kubernetes, se recomienda priorizar la revisión y mitigación de todos los nodos worker y control plane, especialmente aquellos que ejecuten cargas de terceros, workloads multi-tenant o procesos con alto nivel de automatización. CERT-EU recomienda aplicar mitigaciones provisionales de forma inmediata y priorizar nodos Kubernetes y runners CI/CD expuestos a cargas de trabajo no confiables.

En clústeres Kubernetes, los administradores deben drenar los nodos vulnerables, aplicar actualizaciones del kernel, reiniciar los sistemas, verificar la versión corregida y reincorporar los nodos al clúster de forma controlada. También se recomienda revisar políticas de seguridad de pods, evitar contenedores privilegiados, restringir hostPath, limitar hostPID, hostIPC y hostNetwork, controlar el uso de cuentas de servicio y validar que los secretos del clúster no estén innecesariamente expuestos a cargas de trabajo no confiables.

En entornos de integración y despliegue continuo, los equipos DevOps deben reforzar la separación entre runners confiables y no confiables. Los runners que ejecutan código de ramas externas, dependencias de terceros, pruebas automatizadas o artefactos no validados deben considerarse de mayor exposición. Se recomienda evitar que estos runners compartan infraestructura con ambientes productivos, limitar el acceso a secretos, usar runners efímeros cuando sea posible, rotar credenciales de despliegue y aplicar controles estrictos sobre los permisos otorgados a los pipelines.

Como medida preventiva adicional, se recomienda implementar una política de mínimo privilegio para usuarios, servicios y procesos locales. Debido a que la vulnerabilidad requiere ejecución local con bajos privilegios, cualquier cuenta innecesaria, sesión SSH expuesta, usuario compartido, credencial débil o servicio mal configurado puede convertirse en punto de partida para la explotación. Por esta razón, se deben deshabilitar cuentas no utilizadas, restringir accesos SSH, exigir autenticación fuerte, limitar grupos administrativos y revisar periódicamente permisos sobre binarios, scripts y tareas programadas.

También se recomienda reforzar el monitoreo de cambios asociados a privilegios elevados. Aunque las herramientas de integridad basadas únicamente en disco pueden no detectar todos los escenarios de abuso cuando la alteración ocurre en memoria, sí es posible mejorar la visibilidad mediante EDR/XDR para Linux, auditoría de procesos, monitoreo de sesiones, alertas sobre ejecución inesperada como root, cambios en cuentas privilegiadas, modificación de servicios, alteración de logs y ejecución anómala desde contenedores o pipelines. Microsoft advierte que la vulnerabilidad puede ser especialmente riesgosa en

entornos cloud, CI/CD y Kubernetes, donde la ejecución de código no confiable es más frecuente.

En infraestructuras cloud, se recomienda revisar imágenes base, plantillas de despliegue, autoscaling groups, snapshots, AMI, imágenes golden, nodos efímeros y sistemas creados automáticamente mediante infraestructura como código. La mitigación debe aplicarse no solo sobre los servidores activos, sino también sobre las imágenes desde las cuales se crean nuevos sistemas. De lo contrario, la organización puede corregir servidores existentes, pero seguir desplegando nuevas instancias vulnerables desde imágenes desactualizadas.

Las organizaciones también deben fortalecer su proceso de gestión de vulnerabilidades, incorporando esta CVE en los tableros de priorización, escaneo autenticado, validación de exposición, seguimiento de remediación y verificación posterior al parcheo. La priorización debe considerar criticidad del activo, exposición a usuarios no confiables, presencia de contenedores, ejecución de pipelines, acceso SSH, sensibilidad de la información procesada y función del sistema dentro de la operación.

Finalmente, se recomienda definir una estrategia de contingencia para sistemas que no puedan ser actualizados de inmediato. Esta estrategia debe incluir aislamiento temporal, restricción de accesos locales, reducción de usuarios habilitados, segmentación de red, bloqueo de cargas no confiables, monitoreo reforzado, revisión de procesos privilegiados y programación de una ventana de mantenimiento para aplicar el parche definitivo. En sistemas críticos o con sospecha de explotación, se debe considerar la rotación de credenciales, revisión forense y reconstrucción desde imágenes confiables..

Recomendaciones de respuesta ante compromiso

En caso de que se confirme o se sospeche un compromiso debido a la explotación de la vulnerabilidad CVE-2026-31431 "Copy Fail" en sistemas Linux, se deben seguir los siguientes

pasos para contener el incidente, mitigar el daño, preservar evidencia y restaurar la seguridad de los activos afectados:

1. Aislamiento inmediato de los sistemas comprometidos

Desconectar o aislar de la red los servidores, máquinas virtuales, nodos Kubernetes, hosts de contenedores o runners CI/CD donde se sospeche explotación de la vulnerabilidad. Esta medida busca evitar que el atacante mantenga acceso al sistema, ejecute nuevas acciones como root o utilice el host comprometido para realizar movimiento lateral hacia otros activos de la organización.

En entornos cloud o Kubernetes, se recomienda retirar temporalmente los nodos afectados del balanceo de carga, drenarlos del clúster y evitar que sigan ejecutando cargas de trabajo hasta completar la investigación.

2. Preservación de evidencia antes de realizar cambios mayores

Antes de reinstalar, reiniciar o eliminar archivos sospechosos, se debe preservar la mayor cantidad de evidencia posible. Esto incluye registros del sistema, procesos activos, conexiones de red, sesiones de usuario, contenedores en ejecución, tareas programadas, archivos modificados recientemente y eventos de autenticación.

La preservación de evidencia es fundamental para determinar si la vulnerabilidad fue explotada, identificar el vector inicial de acceso y establecer el alcance real del compromiso. En sistemas críticos, se recomienda realizar imágenes forenses del disco y capturas de memoria cuando sea viable.

3. Aplicación urgente de actualizaciones del kernel

Aplicar de forma prioritaria las actualizaciones de seguridad publicadas por el proveedor de la distribución Linux afectada. La remediación debe cubrir servidores físicos, máquinas virtuales, instancias cloud, nodos Kubernetes, hosts de contenedores, appliances Linux y runners CI/CD.

Después de instalar la actualización, se debe reiniciar el sistema para asegurar que el kernel corregido quede cargado en memoria. Posteriormente, se debe validar la versión activa del kernel y confirmar que el sistema ya no ejecuta una versión vulnerable.

4. Revisión del acceso privilegiado obtenido por el atacante

Verificar si el atacante logró obtener privilegios de root o ejecutar acciones administrativas no autorizadas. Para ello, se deben revisar eventos relacionados con cambios de usuario, ejecución de comandos privilegiados, creación de nuevas cuentas, modificación de grupos administrativos, alteración de permisos y uso anómalo de binarios privilegiados.

También se deben analizar shells administrativas inesperadas, procesos ejecutados como UID 0, accesos a archivos restringidos y modificaciones en servicios del sistema. Cualquier evidencia de escalamiento de privilegios debe tratarse como compromiso total del host.

5. Auditoría de registros y actividades sospechosas

Realizar una auditoría exhaustiva de los registros del sistema y de seguridad. Esta revisión debe incluir logs de autenticación, registros de sudo, sesiones SSH, eventos del kernel, logs de contenedores, registros de Kubernetes, eventos del EDR/XDR, actividad de usuarios locales y cambios recientes en archivos críticos.

Se debe prestar especial atención a:

Inicios de sesión inusuales.

- Ejecución de comandos administrativos por usuarios no autorizados.
- Creación o modificación de usuarios privilegiados.
- Cambios en servicios del sistema.
- Alteración o eliminación de logs.
- Desactivación de herramientas de seguridad.

- Actividad anómala desde contenedores o pipelines CI/CD.

6. Rotación de credenciales, secretos y llaves expuestas

Si existe sospecha de explotación exitosa, se deben rotar de inmediato las credenciales que pudieron quedar expuestas en el sistema comprometido. Esto incluye contraseñas de usuarios, llaves SSH, tokens de acceso, secretos de Kubernetes, credenciales cloud, variables de entorno, certificados, claves privadas y credenciales utilizadas por aplicaciones o pipelines de despliegue.

En entornos CI/CD, se recomienda revocar tokens de repositorios, credenciales de despliegue, llaves de acceso a registros de contenedores, secretos de automatización y cualquier credencial almacenada en runners o agentes comprometidos.

7. Análisis forense de la infraestructura afectada

Realizar un análisis forense completo para determinar el alcance del ataque. Este análisis debe establecer cómo se obtuvo el acceso inicial, si la vulnerabilidad fue explotada para escalar privilegios, qué acciones ejecutó el atacante como root y si existió movimiento lateral hacia otros sistemas.

El análisis debe incluir revisión de procesos, binarios sospechosos, mecanismos de persistencia, conexiones externas, modificaciones en archivos del sistema, cambios en configuraciones críticas, tareas programadas, servicios instalados y actividad en cuentas privilegiadas.

8. Búsqueda y eliminación de mecanismos de persistencia

Después de obtener privilegios elevados, un atacante puede intentar mantener acceso al sistema mediante puertas traseras, usuarios ocultos, llaves SSH no autorizadas, tareas programadas, servicios maliciosos, modificaciones en scripts de inicio o alteraciones en configuraciones del sistema.

Por esta razón, se debe revisar cuidadosamente:

- Archivos authorized_keys.
- Usuarios y grupos locales.
- Tareas cron.
- Servicios systemd.
- Scripts de inicio.
- Binarios modificados.
- Reglas de firewall.
- Configuraciones SSH.
- Agentes o procesos no reconocidos.

Si se confirma persistencia, se recomienda no limitar la respuesta a eliminar el archivo malicioso, sino considerar la reconstrucción completa del sistema.

9. Restauración o reconstrucción desde imágenes confiables

Cuando se confirme explotación exitosa o existan indicios de compromiso con privilegios de root, se recomienda reconstruir el sistema desde una imagen limpia y confiable. En estos casos, la limpieza manual puede ser insuficiente, ya que el atacante pudo haber modificado componentes críticos del sistema, ocultado persistencia o alterados registros.

La restauración debe realizarse únicamente desde copias de seguridad verificadas, libres de compromiso y actualizadas con los parches de seguridad correspondientes. Antes de reincorporar el sistema a producción, se debe validar que el kernel corregido esté activo, que las credenciales hayan sido rotadas y que los controles de seguridad estén operativos.

10. Contención específica en Kubernetes y contenedores

En entornos Kubernetes, se deben aislar los nodos sospechosos, drenar las cargas de trabajo, revisar pods ejecutados recientemente, analizar eventos del clúster y verificar si existió acceso no autorizado a secretos, cuentas de servicio, volúmenes o namespaces.

También se recomienda revisar imágenes de contenedores, registros de ejecución, permisos asignados a pods, uso de contenedores privilegiados, montajes hostPath, acceso a hostPID, hostIPC o hostNetwork, y cualquier carga de trabajo que haya podido servir como punto inicial de explotación.

Si el nodo fue comprometido, debe ser eliminado y reconstruido, no simplemente reiniciado.

11. Contención específica en plataformas CI/CD

Si la explotación pudo ocurrir desde un runner, agente de compilación o servidor de automatización, se debe suspender temporalmente la ejecución de pipelines en el host afectado. Posteriormente, se deben revisar los jobs ejecutados, artefactos generados, scripts recientes, repositorios involucrados y secretos disponibles durante la ejecución.

Se recomienda invalidar tokens de despliegue, revisar integridad de artefactos publicados, verificar imágenes de contenedores generadas durante el periodo de compromiso y confirmar que no se haya introducido código malicioso en repositorios, paquetes o ambientes productivos.

12. Verificación de movimiento lateral

Con privilegios de root, el atacante pudo haber recolectado credenciales o secretos para acceder a otros sistemas. Por esta razón, se debe realizar una búsqueda de actividad sospechosa en servidores relacionados, bases de datos, plataformas cloud, repositorios de código, servicios internos, controladores de dominio, sistemas de monitoreo y plataformas de administración.

Se deben correlacionar eventos de autenticación, direcciones IP, horarios, usuarios utilizados y acciones administrativas ejecutadas después del posible compromiso inicial.

13. Notificación y reporte del incidente

Notificar el incidente al equipo interno de respuesta ante incidentes, CSIRT organizacional, mesa de seguridad, oficial de seguridad de la información o autoridad competente, según los procedimientos internos de la entidad.

Si el incidente involucra exposición de datos personales, información sensible, credenciales institucionales o servicios esenciales, se debe evaluar el cumplimiento de las obligaciones legales y regulatorias aplicables, incluyendo la Ley 1581 de 2012 sobre protección de datos personales en Colombia, así como otros marcos normativos que correspondan según el sector.

14. Comunicación a usuarios, clientes o partes interesadas

Si se confirma que la explotación afectó información de usuarios, servicios institucionales, datos sensibles o disponibilidad de plataformas críticas, se debe preparar una comunicación clara y controlada para las partes interesadas.

La comunicación debe explicar el alcance conocido del incidente, las medidas adoptadas, las acciones recomendadas para los usuarios y los canales oficiales de atención. No se deben divulgar detalles técnicos que faciliten nuevas explotaciones o afecten la investigación en curso.

15. Revisión posterior y fortalecimiento de controles

Después de contener y erradicar el incidente, se debe realizar una revisión completa de las medidas de seguridad aplicadas sobre la infraestructura Linux. Esta revisión debe incluir gestión de parches, hardening de sistemas, control de accesos SSH, uso de MFA, monitoreo EDR/XDR, segmentación de red, políticas de contenedores, aislamiento de runners CI/CD y control de privilegios locales.

También se recomienda actualizar los procedimientos de respuesta ante incidentes, ajustar reglas de detección, mejorar los inventarios de activos Linux y reforzar los procesos de validación posterior al parcheo.

Conclusiones

La vulnerabilidad CVE-2026-31431 "Copy Fail" representa una amenaza relevante para organizaciones que operan infraestructura Linux, especialmente en ambientes cloud, Kubernetes, CI/CD y servidores multiusuario.

Aunque requiere ejecución local previa, su impacto es alto porque permite escalar privilegios a root y puede facilitar compromiso del host desde un contenedor o proceso con permisos limitados. La inclusión en el catálogo KEV de CISA confirma que debe tratarse como una vulnerabilidad prioritaria para remediación. NVD también registra que CVE-2026-31431 se encuentra en el catálogo de vulnerabilidades explotadas conocidas de CISA.

La acción recomendada es inventariar, parchear, restringir exposición local y endurecer contenedores. En sistemas donde no exista parche inmediato, se deben aplicar mitigaciones temporales como bloqueo de AF_ALG, aislamiento y control estricto de ejecución de código no confiable.

Fuentes:

- Microsoft Security Blog — análisis técnico, impacto, detecciones y mitigaciones de CVE-2026-31431.
- NVD — registro oficial de CVE-2026-31431 y puntaje CVSS del CNA.
https://www.microsoft.com/en-us/security/blog/2026/05/01/cve-2026-31431-copy-fail-vulnerability-enables-linux-root-privilege-escalation/

- CERT-EU — recomendaciones de mitigación temporal, afectación en distribuciones y endurecimiento de contenedores. https://cert.europa.eu/publications/security-advisories/2026-005/
- BleepingComputer — reporte de incorporación a CISA KEV y explotación activa. https://cert.europa.eu/publications/security-advisories/2026-005/
- Sysdig — versiones vulnerables, versiones corregidas y análisis de impacto. https://www.sysdig.com/blog/cve-2026-31431-copy-fail-linux-kernel-flaw-lets-local-users-gain-root-in-seconds

En caso de dudas, inquietudes o requerir asistencia adicional relacionada con esta alerta de seguridad, puede comunicarse directamente con el **CSIRT Salud** a través de las líneas telefónicas **(+57) 316 893 1490 - 318 155 3570** o mediante el correo electrónico **csirtsalud@minsalud.gov.co**. Nuestro equipo está disponible para brindar el acompañamiento necesario.