Boletín de alerta

Boletín Nro.: 86
Fecha de publicación: 07/08/2026
Tema: Alerta 2026-86 Escape de Máquina Virtual con RCE en host en el Kernel KVM de Linux («Zapscape»)
Traffic Light Protocol (TLP): White

Producto(s) afectado(s):

Linux Kernel — Subsistema KVM/x86 (Kernel-based Virtual Machine)

Producto	Versiones afectadas	Versión corregida
Linux Kernel (mainline)	5.9 hasta 7.1.5	7.1.6
Linux Kernel (stable 6.18.x)	6.18 hasta 6.18.41	6.18.42
Linux Kernel (stable 6.12.x)	6.12 hasta 6.12.100	6.12.101
Linux Kernel (stable 6.6.x LTS)	6.6 hasta 6.6.147	6.6.148
Red Hat / RHEL	Versiones con KVM afectadas	Actualización pendiente
Debian (bullseye, bookworm, trixie)	Todas las ramas actuales	Parche en distribución
CloudLinux 7h, 8, 9, 10	Afectadas	Parche publicado

Descripción

Se ha divulgado una vulnerabilidad de tipo *use-after-free* en el subsistema **KVM/x86** del kernel de Linux. La misma se ha identificado como CVE-2026-64561 con un score CVSSv3.1 7.0 — Importante, el investigador lo nombro Zapscape,

La falla permite que un atacante con privilegios de **root dentro de una máquina virtual guest (L1)**, en sistemas con **virtualización anidada habilitada**, escape del aislamiento de la VM y ejecute código arbitrario con privilegios de **root en el host físico**.

A diferencia de vulnerabilidades anteriores similares, Zapscape es explotable tanto en procesadores **Intel (Ice Lake-SP en adelante)** como **AMD**, ampliando significativamente la superficie de ataque.

El fallo reside en un **orden incorrecto de validación** en el manejo de fallos de página del guest dentro de la shadow MMU de KVM/x86. Cuando KVM intenta reutilizar una shadow page (página de sombra), no verifica correctamente si el *page root* ya fue marcado como inválido por el proceso de reclamación de memoria (make_mmu_pages_available()).

Si el reclaim invalida el root activo, KVM continúa intentando mapear memoria dentro de ese root inválido. Como consecuencia, las shadow pages hijas creadas durante el proceso heredan el estado inválido del padre, **violando el invariante interno de KVM** que establece que las páginas inválidas nunca deben estar en la lista de páginas MMU activas. Esto genera una condición de corrupción de memoria explotable para escape de VM.

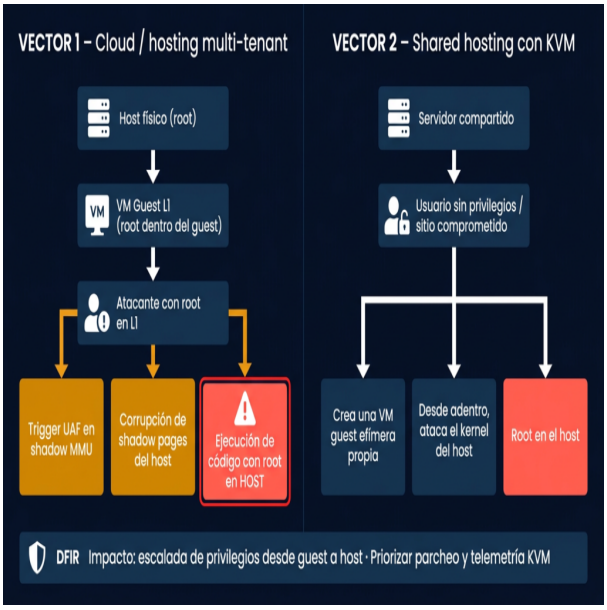

Figura 1 – Escalada de Privilegios en Virtualización: Vectores de Ataque KVM Guest-to-Host

Adicionalmente, se ha hecho publico el **código PoC (Proof of Concept) en GitHub** apenas días después del anuncio, haciendo el riesgo de explotación activa inmediatamente elevado.

El parche mueve la verificación de *stale root* para ejecutarse **después** de make_mmu_pages_available(). Si el reclaim invalida el root actual, KVM ahora reinicia el manejo del fault con RET_PF_RETRY en lugar de continuar mapeando bajo un root inválido.

Solución

Actualizar el kernel a las versiones corregidas:

Rama	Versión corregida
Mainline	7.1.6 o 7.2-rc5
Stable 6.18.x	6.18.42
Stable 6.12.x (LTS)	6.12.101
Stable 6.6.x (LTS)	6.6.148

Actualización del kernel en diversas distribuciones:

```
# Debian / Ubuntu
apt update && apt upgrade linux-image-$(uname -r)
```

```
# RHEL / CentOS / AlmaLinux / Rocky
dnf update kernel

# CloudLinux
yum update kernel

# Arch Linux
pacman -Syu linux

# Verificar versión post-actualización
uname -r
```

Mitigación

Mientras se planea la aplicación del parche del kernel, puede seguir la siguientes recomendaciones:

- **Deshabilitar virtualización anidada** en todos los hosts donde no sea estrictamente necesaria. Es la mitigación más efectiva ya que elimina el prerrequisito principal del ataque:

```
# QEMU/KVM — deshabilitar nested virt por VM
-cpu ${CPU},vmx=off,svm=off

# O a nivel de módulo del kernel
echo «options kvm_intel nested=0» >> /etc/modprobe.d/kvm.conf
echo «options kvm_amd nested=0»   >> /etc/modprobe.d/kvm.conf
modprobe -r kvm_intel && modprobe kvm_intel
```

- **Aplicar live patching** con herramientas como **KSplice** o **kpatch** como medida de primeros auxilios mientras se programa el reinicio para actualizar el kernel.
- **Auditar qué VMs tienen acceso a virtualización anidada** en el entorno y restringirlo exclusivamente a guests de confianza.
- **En entornos cloud multi-tenant:** revisar la configuración de cpu_flags expuestos a los guests y remover vmx/svm de los guests de terceros no confiables.
- **Monitorear** intentos de acceso inusuales al subsistema KVM desde dentro de VMs.

Información adicional:

- The Hacker News — Zapscape CVE-2026-64561: https://thehackernews.com/2026/08/new-zapscape-kvm-flaw-could-let.html
- NVD — CVE-2026-64561: https://nvd.nist.gov/vuln/detail/CVE-2026-64561
- Red Hat — CVE-2026-64561: https://access.redhat.com/security/cve/cve-2026-64561
- CloudLinux — Análisis y mitigación: https://blog.cloudlinux.com/zapscape-cve-2026-64561

- Debian Security Tracker: https://security-tracker.debian.org/tracker/CVE-2026-64561
- Hispasec — Análisis técnico en español: **https://unaaldia.hispasec.com**
- Commit de corrección upstream: 2abd5287f083 — linux/kernel/git/torvalds/linux

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=2abd5287f083