GammaCSOC-CERT
By Gamma Ingenieros

GAMMA INGENIEROS

Boletín de Ciberseguridad Semanal

Edición nº1926

BOLETÍN DE CIBERSEGURIDAD SEMANAL

Bienvenido a su boletín semanal de ciberseguridad generado por Gamma Ingenieros. A través de esta publicación tenemos el propósito de compartir contenidos relevantes para nuestras operaciones y aliados de negocio.

Aquí encontrará información sobre vulnerabilidades, alertas tempranas, noticias del sector, incidentes de ciberseguridad, tecnologías y servicios relacionados, entre otros.

VISTA RÁPIDA

	CRÍTICO	URGENTE	IMPORTANTE
VULNERABILIDADES	3	0	0
MALWARE	0	1	0
NOTICIAS DE CIBERSEGURIDAD	0	0	1

VULNERABILIDADES

CVE-2026-26083 – FORTINET FORTISANDBOX (EJECUCIÓN REMOTA DE CÓDIGO CRÍTICA - RCE)

Fortinet ha publicado una advertencia de seguridad crítica que afecta a sus soluciones de análisis de amenazas FortiSandbox. La falla reside en la interfaz de usuario web (Web UI) y permite a un atacante omitir los controles de autorización para interactuar directamente con el sistema.

Resumen técnico:

- Identificador principal: CVE-2026-26083.
- Severidad: 9.1 (Crítica) según la escala CVSS v3.
- Causa raíz: Una vulnerabilidad de falta de autorización (CWE-862) en el componente de la interfaz gráfica (GUI).
- Mecanismo de falla: Un atacante remoto no autenticado puede enviar peticiones HTTP especialmente diseñadas a la interfaz web del dispositivo para forzar la ejecución de acciones no permitidas.
- Estado de explotación: No se ha reportado explotación activa en la naturaleza al momento de esta publicación, aunque dispositivos de Fortinet suelen ser objetivos prioritarios para actores de amenazas.
- Versiones afectadas: FortiSandbox versiones 5.0.0 a 5.0.1 y 4.4.0 a 4.4.8; así como todas las versiones de FortiSandbox Cloud y PaaS (específicamente ramas 23.x y 24.x).

Impacto potencial:

- Ejecución de código no autorizado: La vulnerabilidad permite que un actor externo ejecute código con los privilegios del servidor web sin necesidad de credenciales válidas.
- Ejecución de comandos del sistema: Posibilidad de ejecutar comandos arbitrarios en el sistema operativo subyacente, comprometiendo la integridad total del appliance.
- Compromiso de datos confidenciales: Acceso a los archivos, muestras de malware y reportes de análisis almacenados dentro de la plataforma de sandboxing.
- Pivoteo en la red: Al comprometer una herramienta de seguridad central, el atacante puede utilizarla como punto de apoyo para realizar desplazamientos laterales dentro de la infraestructura corporativa.

Recomendaciones de mitigación:

1. Actualización inmediata de software: Migrar a las versiones corregidas de FortiSandbox (5.0.2 o 4.4.9 en adelante) según la rama de software que se encuentre en uso.
2. Restricción de acceso a la GUI: Limitar el acceso a la interfaz web de administración únicamente a redes internas de confianza o mediante el uso de VPNs, evitando su exposición directa a Internet.
3. Migración en entornos Cloud/PaaS: Para usuarios de versiones en la nube o como servicio, coordinar la migración a los releases remediados siguiendo las guías oficiales de Fortinet.
4. Monitoreo de tráfico administrativo: Implementar reglas de detección en sistemas perimetrales (IPS/WAF) para identificar peticiones HTTP inusuales o malformadas dirigidas a los puertos de administración de FortiSandbox.

Prioridad: Crítica.

Ampliar información:

- https://unaaldia.hispasec.com/2026/05/fortinet-corrige-dos-fallos-criticos-que-permiten-ejecucion-remota-de-codigo.html
- https://csirt.telconet.net/comunicacion/boletines-servicios/fortinet-corrige-fallas-de-seguridad-que-podrian-comprometer-fortisandbox-fortiap-fortianalyzer-fortimanager-y-fortios/
- https://securityaffairs.com/192047/security/critical-fortinet-vulnerabilities-fixed-in-fortisandbox-and-fortiauthenticator.html
- https://www.securityweek.com/fortinet-ivanti-patch-critical-vulnerabilities/amp/
- https://www.scworld.com/brief/fortinet-addresses-critical-vulnerabilities-in-fortisandbox-and-fortiauthenticator

CVE-2026-43284 / CVE-2026-43500 – KERNEL DE LINUX (ESCALADA LOCAL DE PRIVILEGIOS - DIRTY FRAG)

Se han identificado dos vulnerabilidades críticas denominadas conjuntamente como "Dirty Frag", las cuales permiten la escalada local de privilegios (LPE) hasta obtener acceso de superusuario (root). Este fallo aprovecha errores de lógica determinista en la gestión de fragmentos de memoria y el caché de páginas del kernel, afectando a la mayoría de las distribuciones principales.

Resumen técnico:

- Identificador principal: CVE-2026-43284 (módulo ESP) y CVE-2026-43500 (módulo RxRPC).
- Severidad: 8.8 (Alta) según el estándar CVSS v3.1.
- Causa raíz: Gestión incorrecta de fragmentos de páginas compartidas (CWE-787) y fallos en el mecanismo de "descompartición" de paquetes en protocolos de red específicos.
- Mecanismo de falla: El sistema permite adjuntar páginas de una tubería (pipe) directamente a un búfer de socket (skb), lo que permite a un usuario malintencionado corromper el contenido del caché de páginas del kernel de forma determinista, sin depender de condiciones de carrera.
- Estado de explotación: Se ha confirmado la existencia de exploits funcionales y reportes de explotación limitada en entornos reales para obtener acceso root tras un compromiso inicial.
- Versiones afectadas: Múltiples distribuciones incluyendo Ubuntu (todas las versiones), RHEL (10.1), Fedora (44), CentOS Stream, AlmaLinux y openSUSE Tumbleweed. Los módulos vulnerables datan de código introducido desde 2017.

Impacto potencial:

- Control total del sistema: Un usuario local con pocos privilegios puede elevar sus permisos a root de manera altamente confiable, obteniendo acceso total a la máquina host.
- Escape de contenedores: En entornos de virtualización y contenedores (como Docker o Kubernetes), la vulnerabilidad puede facilitar el escape desde el contenedor hacia el sistema operativo anfitrión.
- Persistencia y evasión: Al comprometer el kernel, el atacante puede deshabilitar herramientas de seguridad, manipular logs y establecer puertas traseras que persistan incluso tras intentos de limpieza superficiales.
- Ineficacia de mitigaciones previas: El ataque es capaz de evadir parches o bloqueos previos diseñados para vulnerabilidades similares como "Copy Fail", lo que lo hace especialmente peligroso para sistemas que se consideraban protegidos.

Recomendaciones de mitigación:

1. Actualización de Kernel: Aplicar de forma inmediata los parches distribuidos por los proveedores de cada distribución (ej. líneas principales de Linux f4c50a4034e6 y aa54b1d27fe0).
2. Deshabilitación de módulos afectados: En sistemas donde no sea posible parchear de inmediato, se recomienda bloquear la carga de los módulos esp4, esp6 y rxrpc mediante la creación de un archivo en /etc/modprobe.d/dirtyfrag.conf con instrucciones de instalación falsas.
3. Limpieza de caché de páginas: Tras aplicar mitigaciones o parches, ejecutar el comando echo 3 > /proc/sys/vm/drop_caches para eliminar posibles residuos de memoria corrupta que puedan persistir en el caché.
4. Restricción de acceso local: Limitar el acceso a terminales y shells locales únicamente a personal administrativo indispensable, reduciendo así la superficie de ataque para la ejecución del exploit inicial.

Prioridad: Crítica.

Ampliar información:

- https://ecosistemastartup.com/linux-kernel-7-0-5-4-versiones-con-parche-parcial-dirty-frag/
- https://www.microsoft.com/en-us/security/blog/2026/05/08/active-attack-dirty-frag-linux-vulnerability-expands-post-compromise-risk/
- https://thehackernews.com/2026/05/linux-kernel-dirty-frag-lpe-exploit.html
- https://security.berkeley.edu/news/dirty-frag-universal-local-privilege-escalation-linux
- https://ubuntu.com/blog/dirty-frag-linux-vulnerability-fixes-available
- https://www.incibe.es/incibe-cert/alerta-temprana/avisos/escalada-local-de-privilegios-en-el-kernel-de-linux-dirty-frag

CVE-2026-7482 – OLLAMA ("BLEEDING LLAMA" - FUGA DE MEMORIA CRÍTICA)

Se ha descubierto una vulnerabilidad crítica denominada "Bleeding Llama" en la plataforma de inteligencia artificial de código abierto Ollama. El fallo permite a atacantes remotos no autenticados extraer información altamente sensible directamente desde la memoria del proceso, lo que pone en riesgo la confidencialidad de los datos manejados por modelos de lenguaje locales (LLMs).

Resumen técnico:

- Identificador principal: CVE-2026-7482.
  Severidad: 9.1 (Crítica) según la escala CVSS v3.1.
- Causa raíz: Una falla de lectura fuera de límites en el montículo (heap out-of-bounds read) dentro del cargador de modelos en formato GGUF.
- Mecanismo de falla: Al procesar un archivo GGUF malicioso con dimensiones de tensores alteradas a través del endpoint /api/create, el servidor realiza lecturas de memoria más allá del búfer asignado, capturando datos almacenados en el heap.
- Estado de explotación: Vulnerabilidad pública con prueba de concepto (PoC) disponible. Se estima que miles de servidores Ollama expuestos a Internet son vulnerables debido a la falta de autenticación por defecto.
- Versiones afectadas: Todas las versiones de Ollama anteriores a la v0.17.1.

Impacto potencial:

- Filtración de credenciales y secretos: Robo de variables de entorno que pueden contener claves de API (OpenAI, Anthropic), credenciales de bases de datos y tokens de acceso a servicios en la nube.
- Exposición de propiedad intelectual: Acceso a fragmentos de código propietario, contratos empresariales y documentos estratégicos que la IA esté procesando en ese momento.
- Compromiso de la privacidad del usuario: Extracción de prompts de sistema, historial de conversaciones activas y resultados de herramientas conectadas (como agentes de IA o Claude Code).
- Persistencia y escalada: El robo de secretos facilita el movimiento lateral del atacante hacia otros sistemas de la organización y la toma de control de cuentas administrativas.

Recomendaciones de mitigación:

1. Actualización mandatoria: Migrar inmediatamente a Ollama v0.17.1 o superior. Los usuarios de Docker deben realizar un pull de la última imagen oficial (docker pull ollama/ollama:latest).
2. Aislamiento de red: Configurar Ollama para que escuche exclusivamente en la interfaz local (127.0.0.1) o dentro de una VPN, evitando el uso de la configuración 0.0.0.0 que lo expone a Internet.
3. Implementación de Proxy de Autenticación: Dado que la API de Ollama no incluye autenticación nativa, se debe desplegar un proxy inverso (ej. Nginx o OAuth2 Proxy) que valide la identidad de quien accede a los endpoints /api.
4. Rotación preventiva de secretos: Si la instancia estuvo expuesta antes del parcheo, se deben rotar todas las claves de API, tokens y credenciales que hayan sido configuradas como variables de entorno en el servidor.

Prioridad: Crítica.

Ampliar información:

- https://www.indusface.com/blog/cve-2026-7482-bleeding-llama-vulnerability/
- https://thehackernews.com/2026/05/ollama-out-of-bounds-read-vulnerability.html
- https://www.akto.io/blog/bleeding-llama-300k-servers-at-risk-response-guide
- https://letsdatascience.com/news/ollama-vulnerability-exposes-remote-process-memory-caf67e65
- https://underc0de.org/foro/noticias-informaticas-120/bleeding-llama-vulnerabilidad-critica-afecta-a-ollama/
- https://hackersonlineclub.com/bleeding-llama-ollama-vulnerability/

MALWARE

PAMDOORA – BACKDOOR DE LINUX BASADO EN PAM (ROBO DE CREDENCIALES Y PERSISTENCIA)

Se ha detectado un nuevo kit de post-explotación para sistemas Linux denominado "PamDOORa", el cual está siendo comercializado en foros de cibercrimen de habla rusa. Este malware destaca por su capacidad de integrarse directamente en el sistema de módulos de autenticación enchufables (PAM), lo que le permite operar con privilegios de root y pasar desapercibido ante herramientas de monitoreo convencionales que operan a nivel de usuario.

Resumen técnico:

- Identificador: PamDOORa.
- Tipo de amenaza: Backdoor avanzado basado en la arquitectura PAM (x86_64).
- Objetivo principal: Establecer persistencia sigilosa en servidores comprometidos y recolectar credenciales SSH de usuarios legítimos.
- Vector de ataque: Herramienta de post-explotación (el atacante requiere obtener privilegios de root previos para el despliegue del módulo).
- Mecanismo de persistencia: Inyección de un módulo malicioso (pam_linux.so) en la pila de autenticación o abuso del módulo legítimo pam_exec para ejecutar scripts durante el inicio de sesión.
- Funcionalidades clave: Implementación de una "contraseña mágica" para acceso persistente, interceptación de contraseñas en texto plano y limpieza automatizada de registros de auditoría.

Impacto potencial:

- Robo masivo de credenciales: Captura nombres de usuario y contraseñas de todos los operarios y administradores que se autentiquen en el sistema afectado, almacenándolos localmente para su posterior exfiltración.
- Acceso persistente y oculto: Permite a los atacantes reingresar al servidor mediante una combinación específica de puerto TCP y contraseña secreta, evadiendo los métodos de autenticación estándar.
- Evasión de análisis forense: Capacidad de manipular y borrar entradas en archivos críticos de registro como lastlog, btmp, utmp y wtmp, eliminando cualquier rastro de conexiones sospechosas.
- Compromiso total de la infraestructura: Al actuar en la capa de autenticación, el atacante puede comprometer otros sistemas internos si los administradores utilizan credenciales compartidas o si el servidor actúa como un salto hacia otras redes.

Recomendaciones de mitigación:

1. Auditoría de configuraciones PAM: Verificar regularmente la integridad de los archivos de configuración en /etc/pam.d/, buscando inclusiones inusuales de módulos externos o el uso sospechoso de pam_exec.so.
2. Monitoreo de integridad de archivos (FIM): Implementar herramientas como Auditd o Tripwire para detectar la creación o modificación de archivos binarios (.so) en los directorios de librerías del sistema.
3. Endurecimiento del servicio SSH: Deshabilitar el acceso directo como usuario root a través de SSH y restringir el uso de sudo únicamente a cuentas estrictamente necesarias y debidamente monitoreadas.
4. Búsqueda de Indicadores de Compromiso (IoCs): Realizar escaneos proactivos en busca de archivos temporales en /tmp/ con nombres aleatorios y verificar la presencia de procesos o scripts no autorizados (ej. tn.sh) ejecutándose con privilegios elevados.

Prioridad: Urgente.

Ampliar información:

- https://foro3d.com/2026/mayo/pamdoora-nuevo-backdoor-linux-roba-credenciales-ssh-con-pam.html
- https://blog.elhacker.net/2026/05/nuevo-backdoor-pamdoora-para-linux.html
- https://www.cryptika.com/new-pamdoora-backdoor-attacking-linux-systems-to-steal-ssh-credentials/
- https://www.scworld.com/brief/new-pamdoora-linux-backdoor-sold-on-cybercrime-forum
- https://thehackernews.com/2026/05/new-linux-pamdoora-backdoor-uses-pam.html
- https://blog.segu-info.com.ar/2026/05/puertas-traseras-y-troyanos-para-linux.html

Recomendaciones generales sobre malware:

1. Actualizar el software y sistemas operativos regularmente, ya que los atacantes suelen explotar vulnerabilidades en software desactualizado.
2. Usar antivirus y herramientas antimalware confiables, manteniéndolos siempre al día.
3. Habilitar la autenticación multifactor (MFA) para proteger cuentas sensibles.
4. Evitar correos electrónicos y enlaces sospechosos, ya que el phishing es una de las principales vías de propagación de malware.
5. Realizar copias de seguridad regulares de archivos importantes para protegerse contra pérdidas en caso de ataques como el ransomware.
6. Limitar los privilegios de usuario, evitando el uso innecesario de cuentas con permisos de administrado

NOTICIAS DE CIBERSEGURIDAD

GOOGLE DETECTA EL PRIMER EXPLOIT ZERO-DAY DESARROLLADO COMPLETAMENTE POR INTELIGENCIA ARTIFICIAL

Investigadores del Google Threat Intelligence Group (GTIG) han confirmado el hallazgo del primer exploit "zero-day" creado íntegramente mediante modelos de lenguaje (LLM). Este descubrimiento marca un hito en la evolución de las amenazas digitales, evidenciando que la inteligencia artificial ha pasado de ser una herramienta de apoyo a una metodología sistemática para el desarrollo de ataques avanzados capaces de evadir defensas tradicionales.

Resumen técnico:

- Descubierto por: Google Threat Intelligence Group (GTIG).
- Tipo de amenaza: Script malicioso en Python (Exploit Zero-day).
- Objetivo principal: Vulnerar la autenticación de dos pasos (2FA) en una herramienta de administración web de código abierto.
- Causa raíz: Explotación de un fallo de lógica semántica y una suposición de confianza indebida en el código del sistema de autenticación.
- Indicios de IA: Presencia de "alucinaciones" (puntuaciones CVSS inventadas), estructuras de código excesivamente académicas, docstrings educativos y patrones de documentación demasiado ordenados, inusuales en desarrollos manuales de ciberdelincuentes.
- Frameworks relacionados: Se ha detectado el uso de sistemas multiagente como "Hexstrike" y "Strix" por grupos avanzados (vinculados a China y Corea del Norte) para automatizar fases de intrusión.

Impacto potencial:

- Bypass de capas críticas de seguridad: La capacidad de saltarse el 2FA permite el acceso a cuentas protegidas incluso si el atacante posee credenciales válidas, comprometiendo la integridad de la gestión web.
- Aceleración masiva del ciclo de vida del ataque: La IA reduce drásticamente el tiempo necesario para realizar reconocimientos detallados, identificar vulnerabilidades y generar payloads funcionales.
- Aumento en la sofisticación de la ofuscación: El uso de IA permite crear grandes volúmenes de código "señuelo" legítimo para camuflar funciones maliciosas, dificultando su detección por sistemas de análisis de comportamiento.
- Automatización de intrusiones a escala: El despliegue de sistemas de IA agéntica permite coordinar tareas de recolección de datos y preparación de ataques de forma prácticamente autónoma y constante.

Recomendaciones para mitigar el riesgo:

1. Refuerzo de la arquitectura MFA: Priorizar el uso de métodos de autenticación resistentes al phishing, como llaves de seguridad físicas (FIDO2) o certificados digitales, sobre métodos basados únicamente en lógica de software.
2. Implementación de auditorías de lógica semántica: Complementar el escaneo de vulnerabilidades tradicional con revisiones de código centradas en identificar suposiciones de confianza erróneas que la IA pueda detectar.
3. Actualización de sistemas de detección (EDR/XDR): Ajustar las herramientas de monitoreo para identificar patrones de ejecución de scripts y comportamientos de red asociados a herramientas de IA agéntica.
4. Adopción de "Zero Trust" en el desarrollo: Eliminar la confianza implícita en cualquier componente del sistema, asegurando que cada interacción y proceso de autenticación sea verificado de manera independiente.

Prioridad: Importante.

Ampliar Información:

- https://www.notimerica.com/ciencia-tecnologia/noticia-portaltic-google-identifica-primera-vulnerabilidad-dia-cero-desarrolla-ayuda-ia-20260511150116.html
- https://mobiletime.la/noticias/11/05/2026/ia-para-evadir-autenticacion-2fa/
- https://es.wired.com/articulos/google-alerta-sobre-hackers-que-ya-utilizan-ia-para-crear-exploits-de-dia-cero
- https://hipertextual.com/seguridad/google-descubre-el-primer-ciberataque-construido-con-inteligencia-artificial/
- https://www.infobae.com/tecno/2026/05/12/google-detecta-el-primer-ciberataque-desarrollado-completamente-con-inteligencia-artificial/