CentralHost

GUÍA DE HARDENING

Cómo blindar tu servidor cPanel

18 pasos ordenados · lista de verificación · ejemplos reales

Guía práctica para proveedores de hosting y administradores de sistemas: blinda tus servidores cPanel, sea uno o una flota.

www.centralhost.sh

Edición 2026

Contenido

Introducción

Un servidor cPanel recién instalado no es un servidor seguro.

Las configuraciones por defecto priorizan la compatibilidad sobre la seguridad: el puerto SSH está completamente abierto, las reglas de firewall son permisivas, hay puertos de panel en texto plano a la escucha, y el WAF está apagado o usa un conjunto de reglas mínimo. En un solo servidor eso es una tarde de limpieza manejable. En una flota de 20, 50 o 200 — donde se aprovisionan servidores nuevos constantemente, las configuraciones se desvían y los miembros del equipo entran y salen — mantener el blindaje deja de ser una tarea y se convierte en una disciplina operativa.

Esta guía es esa disciplina puesta por escrito: **18 pasos concretos, ordenados por impacto**, desde la configuración de SSH a nivel del sistema operativo hasta el firewall, el WAF, el hardening del panel, la autenticación de correo, los backups y la auditoría. Cada paso es práctico, está probado en producción y está diseñado para no romper los sitios de los clientes. Al final encontrarás una **lista de verificación para copiar y pegar** que ejecutar contra cualquier servidor antes de darlo por listo para producción, además de un **apéndice con ejemplos reales** — del tipo de incidentes por los que realmente te llaman.

Una nota sobre de dónde viene esto. Construimos CentralHost, una plataforma para gestionar flotas de cPanel, así que la verás referenciada donde responde a una pregunta que plantea una sección: visibilidad de toda la flota, un mapa de acceso SSH, un asistente de IA que investiga incidentes. Léelas como la respuesta a *"cómo hago esto en 200 servidores"*. Todo lo demás se sostiene por sí solo — el hardening es idéntico tanto si usas CentralHost como si no.

Y una opinión por adelantado, porque le da forma a varios de los pasos: **lo más eficaz que puedes hacer por SSH no es blindarlo — es dejar de exponerlo.** Cerrar el puerto 22 a la internet pública por completo (Pasos 4 y 5) elimina toda una categoría de ataque en lugar de mitigarla. La guía vuelve una y otra vez a esa idea.

Parte 1 — Mantén cPanel actualizado

Paso 1: Habilita las actualizaciones automáticas en el tier STABLE

Lo más impactante que puedes hacer antes de tocar cualquier otra cosa es asegurar que el servidor se mantenga parcheado automáticamente. cPanel gestiona su propio ciclo de vida de actualizaciones — el sistema operativo, el propio panel, los componentes de EasyApache y los paquetes incluidos como PHP y MySQL pasan todos por el mecanismo de actualización de cPanel. Pelearse con eso usando herramientas externas causa más problemas de los que resuelve.

En **WHM → Update Preferences**, configura:

- **Update Tier:** STABLE — no EDGE ni CURRENT en producción. STABLE recibe actualizaciones después de que han sido validadas en la flota de cPanel; los pocos días de retraso valen la estabilidad.
- **Automatic Updates:** Enabled — cPanel aplicará las actualizaciones según su propia planificación, sin intervención manual.

Eso es todo en cuanto a mantenimiento a nivel de sistema operativo. cPanel se encarga del resto. Programa una ventana de reinicio después de las actualizaciones de kernel — los servicios de cPanel se reinician limpiamente en el arranque y el proceso es sencillo.

Parte 2 — SSH

Paso 2: Genera claves SSH y pruébalas antes de restringir el login de root

Antes de configurar `PermitRootLogin prohibit-password`, debes verificar que la autenticación por clave realmente funciona. Dejarte fuera del root en un servidor de producción porque la clave nunca se probó es uno de los incidentes más evitables de esta guía.

Sigue este orden estrictamente: genera la clave → instálala en el servidor → verifica el login → luego restringe la contraseña.

Genera tu par de claves SSH

Linux / macOS

Abre una terminal en tu máquina local:

```
ssh-keygen -t ed25519 -C "your-name@yourcompany"
```

Cuando te pida una ubicación de archivo, pulsa Enter para aceptar la predeterminada (`~/.ssh/id_ed25519`). Establece una passphrase — esto protege la clave si tu máquina se ve comprometida.

Esto crea dos archivos:

- `~/.ssh/id_ed25519` — tu clave privada (nunca la compartas)
- `~/.ssh/id_ed25519.pub` — tu clave pública (esta va en el servidor)

Windows

En Windows 10/11, OpenSSH viene integrado. Abre PowerShell y ejecuta:

```
ssh-keygen -t ed25519 -C "your-name@yourcompany"
```

Las claves se guardan en `C:\Users\YourName\.ssh\`. Si usas PuTTY en lugar de OpenSSH, usa PuTTYgen para generar una clave ed25519 y exporta la clave pública en formato OpenSSH.

Instala la clave pública en el servidor

Linux / macOS — usa `ssh-copy-id`:

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub root@your-server-ip
```

Este comando añade tu clave pública a `/root/.ssh/authorized_keys` en el servidor.

Windows (PowerShell) — `ssh-copy-id` no está disponible de forma nativa, así que hazlo manualmente:

```
# Lee tu clave pública
$pubkey = Get-Content "$env:USERPROFILE\.ssh\id_ed25519.pub"

# Añádela al authorized_keys del servidor (te pedirá la contraseña de root)
ssh root@your-server-ip "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '$pubkey' >>
~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys"
```

Verifica el login por clave — antes de cambiar nada en sshd_config

Abre una **nueva ventana de terminal** (mantén tu sesión actual abierta) y prueba el login por clave de forma explícita:

```
# Linux / macOS
ssh -i ~/.ssh/id_ed25519 root@your-server-ip

# Windows PowerShell
ssh -i $env:USERPROFILE\.ssh\id_ed25519 root@your-server-ip
```

Deberías iniciar sesión sin que te pidan una contraseña (solo la passphrase de la clave si configuraste una). **No continúes al siguiente paso hasta que esto funcione.**

Ahora endurece sshd_config

Solo después de confirmar que el login por clave funciona, edita `/etc/ssh/sshd_config`:

```
# Login de root permitido, pero solo vía clave SSH — nunca por contraseña
PermitRootLogin prohibit-password

# Deshabilita la autenticación por contraseña — solo claves
# No se rompe el acceso de ningún cliente: los clientes usan FTP/FTPS, no SSH (ver Paso
10)
PasswordAuthentication no

# Deshabilita las contraseñas vacías
PermitEmptyPasswords no

# Limita los intentos de login por conexión
MaxAuthTries 3

# Desconecta las sesiones inactivas tras 5 minutos (300s × 1)
ClientAliveInterval 300
ClientAliveCountMax 1

# Deshabilita el reenvío de X11
X11Forwarding no
```

Reinicia SSH — de nuevo, manteniendo tu sesión actual abierta mientras pruebas:

```
systemctl restart sshd
```

Abre una nueva terminal y confirma que todavía puedes iniciar sesión como root con tu clave. Solo entonces cierra la sesión anterior.

Sobre la autenticación por contraseña: en un servidor cPanel donde los clientes usan FTP/FTPS para las transferencias de archivos, SSH solo lleva tráfico administrativo de tu equipo. No hay ninguna razón para mantener `PasswordAuthentication yes`. Ponlo en `no` — no se rompe el acceso de ningún cliente, y los ataques de fuerza bruta contra contraseñas SSH dejan de ser una preocupación por completo. Combinado con `PermitRootLogin prohibit-password`, la configuración de SSH acepta únicamente claves, para todos.

> **En una flota:** la vista de postura de seguridad de CentralHost saca a la luz los servidores donde `PermitRootLogin` no está restringido a claves, en toda tu flota — de modo que la desviación de configuración es visible sin tener que entrar en cada servidor individualmente.

Paso 3: Una clave por operador, nunca compartida

Cada persona que necesite acceso root por SSH debería tener su propio par de claves. Nunca compartas una clave privada entre miembros del equipo.

Mantén un `/root/.ssh/authorized_keys` limpio en cada servidor — solo claves que estén activas actualmente. Audítalo con regularidad y elimina las claves que pertenezcan a antiguos empleados, contratistas que han terminado su trabajo y sistemas dados de baja.

```
# Revisa las claves autorizadas actuales en root
cat /root/.ssh/authorized_keys

# Busca claves añadidas bajo cuentas de hosting
find /home -name authorized_keys 2>/dev/null
```

Para acceso temporal (contratistas, soporte externo), usa el acceso SSH delegado de CentralHost — acotado, limitado en el tiempo, auditado y revocable sin tocar `authorized_keys` en el servidor.

Paso 4: Reduce o elimina la exposición del puerto 22

El puerto 22 abierto a la internet pública es una fuente constante de ruido de fuerza bruta. Hay dos enfoques prácticos:

Opción A — Cierra el puerto 22 por completo (recomendado si usas CentralHost)

CentralHost te da dos formas de acceder a tus servidores sin un puerto 22 abierto:

- **Terminal SSH en el navegador.** Una sesión SSH completa directamente en tu navegador, autenticada a través de CentralHost. No requiere cliente SSH local, funciona desde cualquier sitio.
- **Bastión SSH.** CentralHost provee un host bastión — un único punto de entrada blindado a través del cual se enrutan todas las conexiones SSH a tu flota. En lugar de conectarte directamente a cada servidor, tu cliente SSH se conecta al bastión, que actúa de proxy de la sesión hacia el servidor

destino sobre el canal cifrado del agente. Desde la perspectiva del operador se comporta como una conexión SSH normal; desde la perspectiva del servidor, no llega nunca tráfico entrante al puerto 22 desde internet.

Ambos métodos te permiten cerrar el puerto 22 a la internet pública vía PXF — con una diferencia: si usas el bastión, añade la IP del bastión de CentralHost como una regla de permiso explícita en PXF antes de cerrar el puerto 22. El terminal web no tiene ese requisito, ya que no se conecta de forma entrante al puerto SSH de tu servidor directamente.

Las transferencias de archivos de los clientes usan FTP/FTPS en el puerto 21 (ver Paso 10), así que cerrar el puerto 22 no tiene impacto en los clientes de hosting.

Opción B — Restringe el puerto 22 a IPs conocidas

Si cerrar SSH por completo no es factible de inmediato, restringe el puerto 22 a la IP de tu oficina o al endpoint de tu VPN vía reglas del firewall PXF y bloquea todo lo demás. Esto elimina la exposición a toda internet mientras preservas el acceso directo desde ubicaciones de confianza.

En la práctica, este enfoque requiere o bien una IP estática en cada ubicación desde la que trabaje tu equipo, o bien una VPN con una IP de salida fija. Sin una de esas, la lista de permitidos se vuelve difícil de mantener — las IPs dinámicas cambian, los miembros del equipo trabajan en remoto, y cada excepción abre otro agujero en la regla. Esto es precisamente por lo que el modelo de bastión gana terreno: en lugar de gestionar una lista cambiante de IPs de confianza, pones en la lista de permitidos una única IP fija — el bastión — y accedes a toda tu flota a través de ella desde cualquier sitio.

Cualquiera de las dos opciones es una mejora sustancial respecto a un puerto 22 completamente abierto. Si ninguna es factible de inmediato, la protección contra fuerza bruta de PXF (Paso 5) combinada con `PermitRootLogin prohibit-password` y `PasswordAuthentication no` elimina los vectores de ataque significativos mientras planificas la migración.

Parte 3 — Firewall

Paso 5: Instala y configura PXF

Durante más de una década la respuesta por defecto aquí fue **CSF (ConfigServer Security & Firewall)**. Esa era se acabó: CSF/LFD ya no se mantiene activamente y ha llegado al fin de su vida útil, sin más desarrollo por parte de ConfigServer. Ejecutar un firewall abandonado como tu primera línea no es una postura de seguridad — sin parches, envejece hasta convertirse en un pasivo. Si todavía tienes CSF en tus servidores, planifica una migración para salir de él.

PXF es un firewall gratuito y mantenido activamente, construido específicamente para servidores cPanel — un reemplazo directo de CSF en ese rol. Se integra de forma nativa con WHM, la línea de comandos y CentralHost, y admite tanto `iptables` como `nftables`, cubriendo todo el rango de distribuciones actuales compatibles con cPanel.

Instala PXF siguiendo las instrucciones en pyxsoft.com. Una vez instalado, está disponible en WHM como plugin.

PXF cierra todo por defecto y abre automáticamente los puertos que un servidor cPanel necesita — HTTP, HTTPS, SMTP, IMAP, POP3, FTP y los puertos del panel cPanel/WHM. No necesitas configurar una lista de puertos para abrir o cerrar manualmente. Los valores por defecto conscientes de cPanel se aplican en la instalación.

Las únicas decisiones que necesitas tomar tras instalar PXF son las excepciones:

Puerto SSH

PXF detecta el puerto SSH configurado en el servidor y lo mantiene abierto por defecto. Para restringirlo o cerrarlo, necesitas configurar explícitamente una excepción después de la instalación. Dependiendo de cómo accede tu equipo a los servidores:

- **Terminal web de CentralHost:** el terminal web funciona sobre una conexión saliente desde el agente — no se ve afectado por que el puerto SSH esté abierto o cerrado. Puedes cerrarlo con seguridad.
- **Bastión de CentralHost:** restringe el puerto SSH solo a la IP del bastión. Tu equipo se conecta al bastión desde cualquier sitio y el bastión se conecta de forma entrante a tu servidor — solo esa única IP necesita estar permitida.
- **SSH directo desde IPs fijas:** restringe el puerto SSH a la IP de tu oficina o VPN.

Puertos de panel en texto plano (2082, 2086, 2095)

PXF los abre por defecto por compatibilidad. Si fuerzas HTTPS en cPanel, WHM y Webmail (Paso 11), ciérralos en PXF — no hay razón para mantener abierto el acceso al panel en texto plano.

Habilita la protección contra fuerza bruta en los ajustes de PXF. PXF monitoriza los intentos de login fallidos contra SSH y bloquea automáticamente las IPs ofensoras. Esto solo es relevante si el puerto SSH está abierto — si lo has cerrado o restringido a una única IP de confianza, la fuerza bruta contra SSH ya es un no-problema a nivel de red. La protección contra fuerza bruta para cPanel, WHM, FTP y correo la gestiona por separado cPHulk (Paso 12).

Como PXF se integra con CentralHost, las reglas anteriores no tienen que configurarse servidor por servidor: gestionas puertos, excepciones y bloqueos de IP en toda la flota desde un solo panel, sin entrar en cada WHM ni abrir una terminal.

Paso 6: Bloquea IPs y redes maliciosas conocidas

PXF admite listas de bloqueo de IP. Habilita las actualizaciones automáticas de las listas de bloqueo para mantener bloqueadas las redes maliciosas conocidas sin intervención manual.

Considera también bloquear el tráfico de regiones a las que no das servicio, si tu base de clientes está geográficamente limitada. La función de bloqueo por país de PXF maneja esto a nivel de firewall.

Parte 4 — Web Application Firewall

Paso 7: Reemplaza ModSecurity por pxShield

ModSecurity es el motor WAF por defecto en los servidores cPanel. Funciona, pero tiene inconvenientes operativos bien conocidos: se ejecuta dentro del pipeline de procesamiento de peticiones de Apache, los falsos positivos son opacos y difíciles de afinar, y la gestión de reglas requiere acceso directo a los archivos de configuración. La opción comercial dominante, **Imunify360**, no escapa de esto — su WAF es una capa de conjunto de reglas y reputación en la nube que se ejecuta *sobre el mismo motor de ModSecurity*, así que hereda el modelo de ejecución dentro de Apache (su componente separado Proactive Defense es protección en tiempo de ejecución a nivel de PHP, no el WAF). Tanto si usas ModSecurity a secas como Imunify360, el filtrado sigue ocurriendo dentro de Apache, consumiendo procesos worker.

pxShield es un WAF construido específicamente para entornos de hosting cPanel. La diferencia arquitectónica clave no son las reglas sino *dónde se ejecuta*: **pxShield se sitúa como un reverse proxy delante de Apache**, inspeccionando y filtrando el tráfico antes de que llegue siquiera a tu servidor web.

Ventajas prácticas:

- **Gestión de falsos positivos con UI completa.** Cuando se bloquea una petición legítima, puedes revisar, poner en whitelist y afinar reglas directamente desde WHM o CentralHost — sin necesidad de editar archivos de configuración.
- **Visibilidad completa en WHM y CentralHost.** Las peticiones bloqueadas, las reglas activas, los patrones de tráfico y las alertas son visibles en ambas interfaces. Desde CentralHost obtienes una vista de toda la flota a través de todos los servidores.
- **Mejor rendimiento.** Filtrar antes de Apache significa que las peticiones rechazadas nunca consumen procesos worker de Apache.
- **Conjunto de reglas consistente.** Las reglas y el afinado se gestionan de forma centralizada y se aplican uniformemente.
- **Visibilidad de cara al cliente.** pxShield añade un plugin a la cuenta cPanel de cada cliente que muestra los bloqueos activos y la actividad del WAF en su sitio. Los clientes pueden ver que su proveedor de hosting tiene medidas de seguridad activas — una señal de confianza tangible que la mayoría de los proveedores de hosting compartido no ofrecen.

Instala pxShield siguiendo las instrucciones en pyxsoft.com. Tras la instalación, habilítalo por dominio o globalmente, luego revisa el conjunto de reglas por defecto y afina cualquier falso positivo antes de ponerlo en modo de bloqueo completo.

> **A lo largo de la flota:** la pregunta difícil de responder a escala no es "¿está afinado el WAF?" — es "¿cuáles de mis servidores no tienen ninguna cobertura de WAF en absoluto?" El panel de postura de seguridad de CentralHost responde exactamente eso, listando el estado de pxShield, los conteos de peticiones bloqueadas y las alertas activas por servidor, de modo que un servidor que se coló por el aprovisionamiento sin un WAF deja de ser invisible.

Parte 5 — Específico de cPanel y WHM

Paso 8: Restringe el acceso a WHM por IP

WHM (puerto 2087) no debería ser alcanzable desde direcciones IP arbitrarias. El método más fiable en un servidor cPanel es bloquear el puerto 2087 a nivel de firewall vía PXF: permite solo tus IPs conocidas, deniega todo lo demás. Este enfoque sobrevive a las actualizaciones de cPanel y no depende de ninguna capa de configuración específica de cPanel.

En PXF, añade una regla para restringir el puerto 2087 (y el 2086 si todavía no lo has cerrado) a tu rango de IP de administración. Por ejemplo, si tu equipo accede a WHM desde una única IP de oficina:

```
# Restringe WHM (2087) a tu IP de administración, luego aplica el cambio.
# Acotar allow-port a un --cidr abre el puerto solo para ese origen.
pxf ports allow-port --port 2087 --proto tcp --cidr YOUR.ADMIN.IP/32
pxf apply
```

Como alternativa, usa **WHM → Security Center → Host Access Control** para establecer reglas de TCP wrapper — pero ten en cuenta que los TCP Wrappers (`/etc/hosts.allow` / `/etc/hosts.deny`) están obsoletos desde RHEL 8 y solo funcionan para servicios compilados explícitamente con soporte de libwrap, lo cual excluye a la mayoría de los servicios modernos. Las reglas de firewall de PXF son el método correcto y fiable aquí.

Si tu equipo accede a WHM desde IPs dinámicas, la alternativa práctica es una VPN: enruta todo el acceso a WHM a través de un endpoint de VPN fijo, luego pon ese endpoint en la whitelist de PXF.

Paso 9: Habilita la autenticación de dos factores en WHM

Exige 2FA para todas las cuentas de WHM. Ve a **WHM → Two-Factor Authentication** y fuérzala para root y todas las cuentas de reseller.

Esto es especialmente importante si WHM es accesible por la internet pública.

Paso 10: Deshabilita los servicios de cPanel que no se usen

En **WHM → Service Manager**, deshabilita los servicios que tus clientes no usan:

- **Anonymous FTP** — deshabilítalo por completo. No hay ningún caso de uso legítimo para FTP anónimo en un servidor de hosting compartido.
- **FTP** — mantenlo habilitado para el acceso a archivos de los clientes. FTP con TLS (FTPS) es el método estándar de transferencia de archivos de los clientes en servidores cPanel. La elección de FTPS sobre SFTP aquí es **operativa, no una clasificación de seguridad** — ambos están cifrados. FTPS usa un puerto dedicado (21), lo que mantiene la transferencia de archivos de los clientes desacoplada de SSH para que puedas restringir o cerrar el puerto 22 sin afectar a los clientes; además es compatible de forma universal con herramientas como FileZilla y encaja de forma natural con los usuarios FTP por cuenta de cPanel. Asegúrate de que Pure-FTPd (el daemon FTP

por defecto de cPanel) está configurado para requerir conexiones TLS (FTPS sin TLS es texto plano — esa es la parte que realmente importa para la seguridad).
- **Cpsrvd** puerto en texto plano (2082) — deshabilítalo si fuerzas HTTPS en el panel.

Mantener el acceso a archivos de los clientes en FTP/FTPS en lugar de SFTP también simplifica el hardening de SSH: el puerto SSH lleva solo tráfico administrativo (tu equipo, automatización), no conexiones de clientes, lo que hace mucho más fácil restringirlo por IP o cerrarlo por completo.

Paso 11: Fuerza HTTPS en cPanel, WHM y Webmail

En **WHM → Tweak Settings**, habilita:

- Always redirect to SSL
- Require SSL for cPanel, WHM, and Webmail

Asegúrate también de que el hostname de tu servidor tiene un certificado SSL válido. El AutoSSL de cPanel maneja esto automáticamente — verifica que está habilitado y en marcha en **WHM → Manage AutoSSL**.

Paso 12: Configura la protección contra fuerza bruta de cPHulk

cPHulk es la capa de protección contra fuerza bruta integrada de cPanel para el propio panel — monitoriza los intentos de login fallidos a WHM, cPanel, Webmail, FTP y SSH y bloquea las IPs que exceden los umbrales configurados. Habilítalo y configúralo en **WHM → Security Center → cPHulk Brute Force Protection**.

Ajustes recomendados:

- **Number of failures that trigger a two-week brute force IP block:** 15
- **Number of failures that trigger a two-week brute force account block:** 15
- **Number of failures per IP address before an IP is reported:** 25
- **Look-back timeframe (minutes):** 15

Un paso crítico antes de habilitarlo: **pon tus propias IPs de administración en la whitelist** bajo la pestaña Whitelist. Si te equivocas al teclear tu propia contraseña tres veces desde la IP de tu oficina, cPHulk te bloqueará el acceso a WHM. Añade la IP de tu oficina y cualquier otra IP de administración fija a la whitelist antes de activar la protección contra fuerza bruta. La protección contra fuerza bruta de PXF opera en la capa de red antes que cPHulk, así que ambas funcionan juntas de forma efectiva — PXF descarta los ataques repetidos en el firewall, cPHulk maneja los fallos a nivel de aplicación que se cuelan.

Parte 6 — Seguridad del correo

Paso 13: Aplica SPF, DKIM y DMARC para todos los dominios

La autenticación de correo es tanto un requisito de seguridad como de entregabilidad. En los servidores cPanel, DKIM y SPF se gestionan a nivel de dominio — cPanel genera las claves y publica los registros DNS automáticamente cuando se añade un dominio, siempre que el servidor sea también el servidor DNS autoritativo para ese dominio.

Habilita DKIM y SPF globalmente para que estén activos para todas las cuentas nuevas por defecto: ve a **WHM** → **Email** → **Email Authentication** y asegúrate de que tanto DKIM como SPF están habilitados. Esto controla el valor por defecto para las cuentas recién creadas.

Audita los dominios existentes en masa vía **WHM** → **Email** → **Email Deliverability**. Esta vista muestra cada dominio del servidor con el estado de sus registros SPF, DKIM y DMARC, y ofrece una reparación con un clic para los dominios donde los registros faltan o son incorrectos. En un servidor con muchas cuentas, esta es la forma práctica de arreglar las brechas sin tocar cada cuenta individualmente.

Para los dominios donde el DNS está alojado externamente (no en el servidor cPanel), cPanel no puede publicar los registros automáticamente — la zona necesita actualizarse manualmente en el proveedor de DNS. La página Email Deliverability muestra los registros exactos que hay que añadir.

DMARC no lo configura automáticamente cPanel. Añade un registro DMARC por dominio empezando en modo de monitorización:

```
_dmarc.yourdomain.com  TXT  "v=DMARC1; p=none; rua=mailto:dmarc-reports@yourdomain.com"
```

Pasa a `p=quarantine` y luego a `p=reject` después de revisar los informes agregados y confirmar que no falta ninguna fuente de correo legítima en SPF/DKIM.

Auditar esto un servidor a la vez vía Email Deliverability está bien para un puñado de dominios; a lo largo de una flota no escala. El panel de operaciones de correo de CentralHost resume el estado de SPF/DKIM/DMARC y la presencia en listas negras por dominio en todos los servidores, de modo que los dominios con brechas de autenticación afloran como una lista para ir resolviendo en lugar de ser algo que descubres cuando el correo empieza a rebotar.

Paso 14: Establece límites de envío de correo por cuenta

Las cuentas cPanel comprometidas son una fuente común de spam. Establece límites horarios de envío de correo por cuenta para limitar el alcance del daño:

En **WHM** → **Tweak Settings** → **Mail**:

- Máximo de correos por hora por dominio: 200 (ajústalo según tu base de clientes)
- Porcentaje máximo de mensajes fallidos o diferidos: 20%

Cuando una cuenta alcanza el límite, cPanel suspende el envío de correo para esa cuenta automáticamente. Verás el pico en la monitorización de CentralHost antes de que se convierta en un problema de entregabilidad para tus otros clientes.

Parte 7 — Backups

Paso 15: Configura destinos de backup remotos

Los backups locales en el mismo servidor no son backups — son copias que desaparecen con el servidor. Configura destinos de backup remotos:

En **WHM → Backup Configuration**:

- Habilita los backups
- Establece un destino remoto: almacenamiento compatible con S3, SFTP o un servidor de backup dedicado
- Retén como mínimo: backups diarios durante 7 días, semanales durante 4 semanas

Prueba los procedimientos de restauración periódicamente. Un backup que nunca has restaurado es un backup en el que no puedes confiar.

> **El modo de fallo que esto previene:** los backups fallan en silencio, y te enteras el día en que un cliente pide una restauración. CentralHost le da la vuelta a eso — su panel de backups muestra el estado en toda la flota (qué servidores tuvieron éxito, cuáles fallaron, qué cuentas no tienen ninguna cobertura) y un backup fallido se convierte en una alerta *antes* de convertirse en un incidente.

Parte 8 — Monitorización y auditoría

Paso 16: Conoce qué se está ejecutando en tus servidores

cPanel no tiene un interruptor dedicado de "monitorización de procesos" en WHM. Lo que sí tiene es útil si sabes dónde mirar:

WHM → Server Status → Service Status muestra de un vistazo la salud de todos los servicios gestionados por cPanel — httpd, exim, dovecot, mysql, named, ftpd y otros. Comprueba esto primero cuando algo se comporte de forma inesperada.

WHM → Server Status → CPU/Memory/MySQL Usage da una instantánea actual de los consumidores de recursos. Para métricas históricas, cPanel incluye integración con Munin si se habilitó durante la instalación.

Para visibilidad a nivel de proceso que va más allá de lo que WHM saca a la luz, los logs del sistema relevantes en un servidor cPanel son:

```
# Intentos fallidos de SSH y su
/var/log/secure

# Logs de correo de Exim — spam, fallos de entrega, abuso de relay
/var/log/exim_mainlog
/var/log/exim_rejectlog

# Eventos de servicio de cPanel
/usr/local/cpanel/logs/error_log

# Acceso y errores de Apache por dominio (bajo /home/user/logs/)
# Más el log principal de errores de Apache:
/etc/apache2/logs/error_log
```

Para visibilidad de toda la flota — detectar un servidor con un proceso inusual, un pico en la profundidad de la cola de exim, o un cron job añadido por una cuenta comprometida — aquí es donde importa la monitorización centralizada:

Aquí es donde CentralHost deja de ser un tablero y empieza a hacer el trabajo. Su **Asistente de Operaciones con IA** investiga anomalías directamente en cualquier servidor de tu flota — preguntas en lenguaje natural (*"¿Por qué web-03 está consumiendo el 90% de memoria?"*, *"¿Hay algo inusual en los logs de exim de este servidor?"*) y lee los logs, correlaciona las métricas y explica lo que encontró. No cambia nada en el servidor a menos que tú lo apruebes. El apéndice completo al final de esta guía recorre investigaciones reales.

Paso 17: Audita el acceso SSH con regularidad

Revisa periódicamente quién tiene acceso SSH a cada servidor. Comprueba:

```
# Lista los usuarios con acceso a shell
grep -v '/sbin/nologin\|/bin/false' /etc/passwd

# Comprueba authorized_keys de root y todos los usuarios
find /root /home -name authorized_keys -exec cat {} \;

# Logins SSH exitosos recientes
last -n 50

# Intentos SSH fallidos recientes
grep "Failed password" /var/log/secure | tail -50
```

Elimina cualquier clave autorizada que pertenezca a antiguos empleados, contratistas o sistemas que ya no estén en uso.

Ejecutar esos comandos a mano en cada servidor es exactamente la tarea pesada que no se hace de forma consistente. El panel de postura de seguridad de CentralHost mantiene un mapa de acceso SSH permanente para toda la flota — quién tiene acceso a shell en cada servidor, qué claves están autorizadas, y qué usuarios pueden escalar a root — de modo que la auditoría anterior es algo que lees, no algo que tienes que ejecutar.

Paso 18: Habilita el logging de auditoría para cambios críticos

`auditd` provee logging a nivel de kernel de los cambios de archivos y las llamadas al sistema. En servidores cPanel vale la pena habilitarlo, pero el conjunto de reglas debe ser conservador: una regla general que registre cada comando ejecutado por root (`-S execve -F euid=0`) genera un volumen enorme de eventos en un servidor activo — Apache, Exim, MySQL y los propios daemons de cPanel ejecutan procesos como root o con privilegios elevados constantemente. El log de auditoría se llena rápido y se vuelve imposible de revisar.

Instala y habilita auditd:

```
dnf install audit -y
systemctl enable --now auditd
```

Usa reglas dirigidas que capturen cambios significativos sin inundar el log. Crea `/etc/audit/rules.d/cpanel-hardening.rules`:

```
# Rastrea cambios en los archivos de usuario y autenticación
-w /etc/passwd -p wa -k passwd_changes
-w /etc/shadow -p wa -k shadow_changes
-w /etc/sudoers -p wa -k sudoers_changes
-w /etc/sudoers.d/ -p wa -k sudoers_changes

# Rastrea cambios en la configuración de SSH
-w /etc/ssh/sshd_config -p wa -k sshd_config

# Rastrea cambios en la configuración de cPanel
-w /etc/cpanel/ -p wa -k cpanel_config
-w /usr/local/cpanel/etc/ -p wa -k cpanel_config

# Rastrea modificaciones de cron (vector de persistencia común tras un compromiso)
-w /var/spool/cron/ -p wa -k cron_changes
-w /etc/cron.d/ -p wa -k cron_changes

# Rastrea cambios en PXF y las reglas de firewall
-w /etc/pxf/ -p wa -k firewall_changes
```

Carga las nuevas reglas:

```
augenrules --load
```

Estas reglas registran los cambios de configuración, no cada comando — que es lo que importa para la auditoría. Si necesitas logging a nivel de comando para usuarios específicos (p. ej. resellers con acceso a shell), acótalo por UID en lugar de habilitarlo globalmente.

> **Atribución, no solo logging:** `auditd` registra qué cambió en el host; no puede decirte *quién* lo hizo una vez que varias personas comparten root. Cada acción tomada *a través de* CentralHost — comandos en el terminal SSH, acciones del asistente de IA, sesiones de acceso delegado — se registra con atribución completa: quién la ejecutó, cuándo, en qué servidor y qué devolvió el servidor. Las dos capas se complementan.

Lista de verificación del hardening

Usa esto como referencia rápida antes de dar por listo un servidor para producción:

Actualizaciones de cPanel

- [ ] Update tier configurado en STABLE en WHM → Update Preferences
- [ ] Actualizaciones automáticas habilitadas

SSH

- [ ] Autenticación de root por contraseña deshabilitada — `PermitRootLogin prohibit-password`
- [ ] Autenticación por contraseña deshabilitada — `PasswordAuthentication no`
- [ ] `PermitEmptyPasswords no`
- [ ] Puerto 22 cerrado o restringido a IPs conocidas vía PXF
- [ ] MaxAuthTries configurado en 3
- [ ] Timeout de inactividad configurado
- [ ] authorized_keys auditado — una clave por operador, sin claves compartidas ni obsoletas

Firewall

- [ ] PXF instalado y configurado
- [ ] Puertos innecesarios cerrados
- [ ] Protección contra fuerza bruta habilitada
- [ ] Listas de bloqueo de IPs maliciosas conocidas activas

WAF

- [ ] pxShield instalado y en modo de bloqueo
- [ ] Falsos positivos revisados y afinados
- [ ] Cobertura del WAF verificada en todos los dominios alojados

cPanel / WHM

- [ ] Acceso a WHM restringido por IP
- [ ] Autenticación de dos factores forzada en WHM
- [ ] Anonymous FTP deshabilitado
- [ ] FTP/FTPS habilitado para los clientes — TLS requerido en Pure-FTPd
- [ ] Puertos de panel en texto plano (2082, 2086, 2095) deshabilitados
- [ ] HTTPS forzado en todas las interfaces del panel
- [ ] Protección contra fuerza bruta de cPHulk habilitada

Correo

- [ ] SPF habilitado para todos los dominios
- [ ] Firma DKIM habilitada globalmente
- [ ] Registros DMARC publicados

- [ ] Límites de envío por cuenta configurados

Backups

- [ ] Destino de backup remoto configurado
- [ ] Política de retención de backups establecida
- [ ] Procedimiento de restauración probado

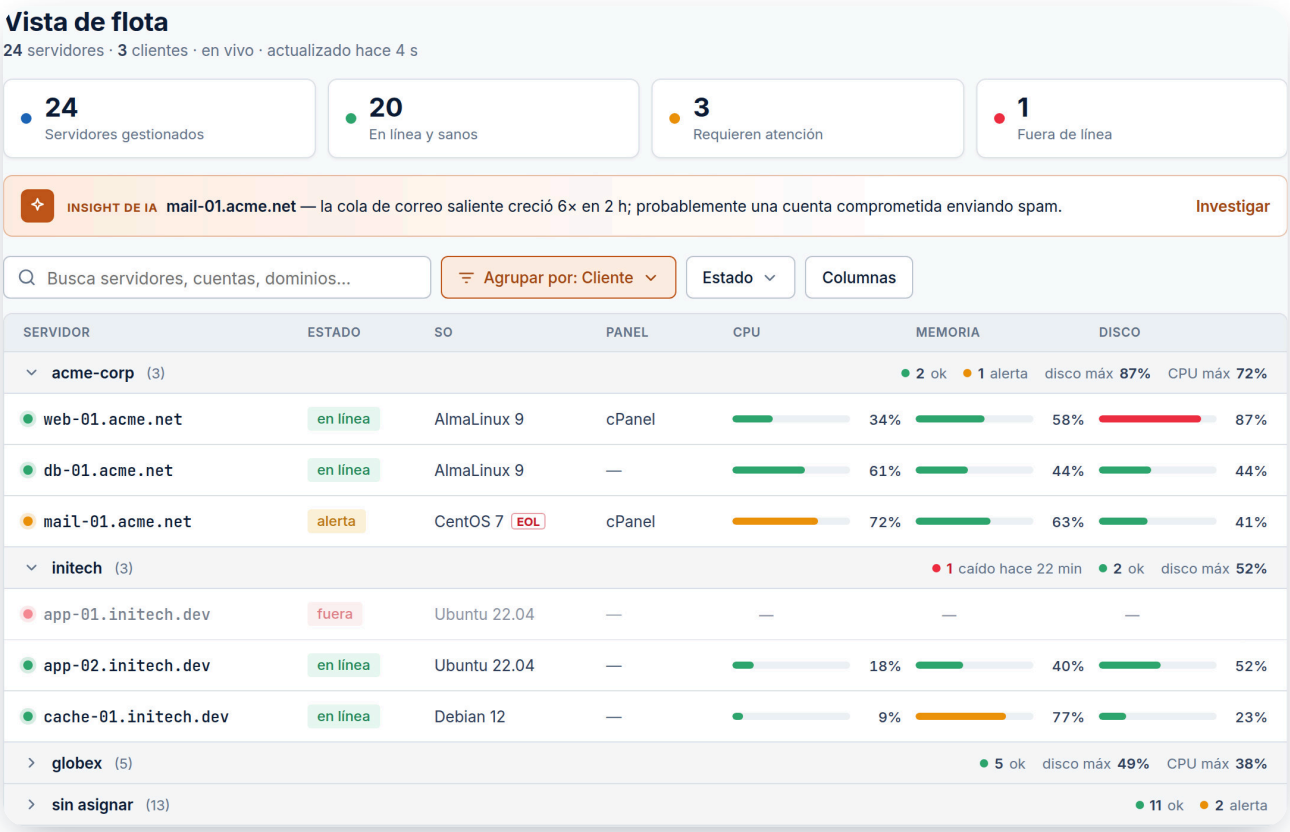
Monitorización y auditoría

- [ ] Monitorización de procesos y logins activa
- [ ] authorized_keys de SSH auditadas
- [ ] auditd instalado y reglas configuradas

Gestionar esto a lo largo de una flota

Recorrer esta lista de verificación una vez en un solo servidor es sencillo. El reto operativo es mantenerla de forma consistente en decenas o cientos de servidores a lo largo del tiempo — a medida que las configuraciones se desvían, se añaden servidores nuevos y los miembros del equipo entran y salen.

CentralHost centraliza la postura de seguridad de toda tu flota: configuraciones SSH débiles, falta de cobertura de WAF, backups fallidos, puertos expuestos, brechas de autenticación de correo y procesos sospechosos afloran todos como hallazgos priorizados — sin entrar en cada servidor. Cuando algo necesita arreglarse, el Asistente de Operaciones con IA investiga y propone la remediación. Tú apruebas el cambio.



La vista de flota: estado por servidor agrupado por cliente, y un Insight de IA que ya señala el spam saliente — todo sin entrar en cada máquina.

El primer servidor es gratis, sin tarjeta de crédito: centralhost.sh

---

Última actualización: junio de 2026

Apéndice I: IA aplicada a la administración de sistemas

Pregúntale a tu servidor.

Durante décadas, diagnosticar un servidor ha significado lo mismo: entrar por SSH, abrir cinco o seis logs, encadenar `grep`, `tail` y `awk`, y reconstruir a mano qué pasó. Funciona, pero no escala — y a las 3 de la madrugada, con un cliente esperando, es lento y propenso a errores.

CentralHost le da la vuelta: un asistente de IA con acceso a cada servidor de tu flota, al que le preguntas en lenguaje natural. Lee los logs, correlaciona las métricas, inspecciona los procesos y te explica qué encontró — con las líneas de evidencia concretas, no una respuesta vaga.

Dos cosas antes de los ejemplos:

- **Llega a tus servidores por el canal saliente del agente**, no por un puerto abierto. Funciona aunque hayas cerrado el puerto 22 (Paso 4) — de hecho, esa es la idea.
- **Opera en dos modos, y tú mandas.** *Investiga* en solo lectura: lee, correlaciona y propone, sin cambiar nada. Cuando le pides que *actúe*, cada acción requiere tu aprobación explícita — una acción, una aprobación; nunca en lote, nunca en silencio.

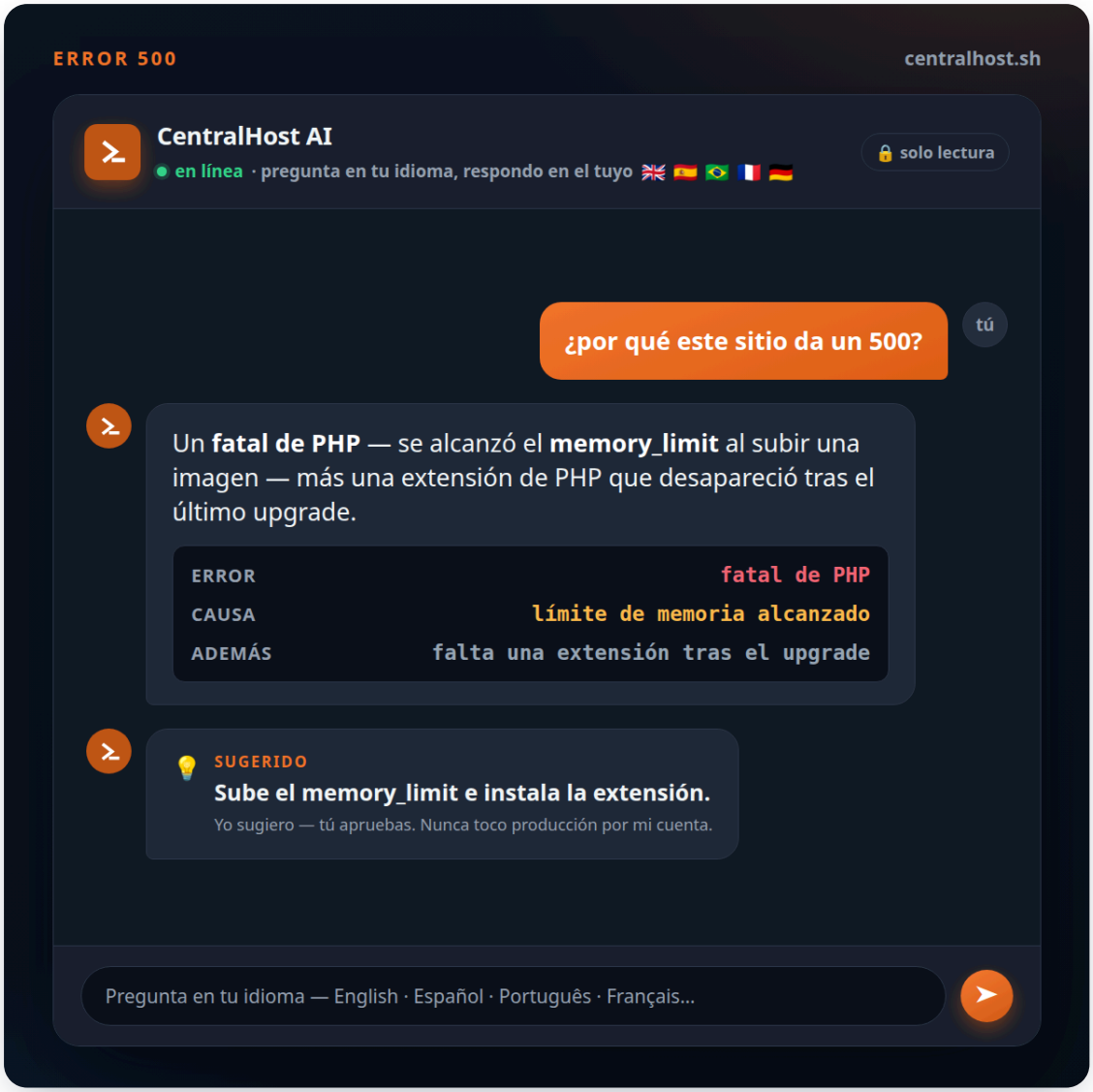
A continuación, tres investigaciones reales tal como se ven en el producto.

Investigar: pregunta qué va mal

Un buzón comprometido enviando spam. Tu IP acaba en una lista negra y no sabes quién la quemó. Encontrarlo a mano son 15-20 minutos cavando en `exim_mainlog`; el asistente identifica el buzón exacto en segundos y propone la contención:

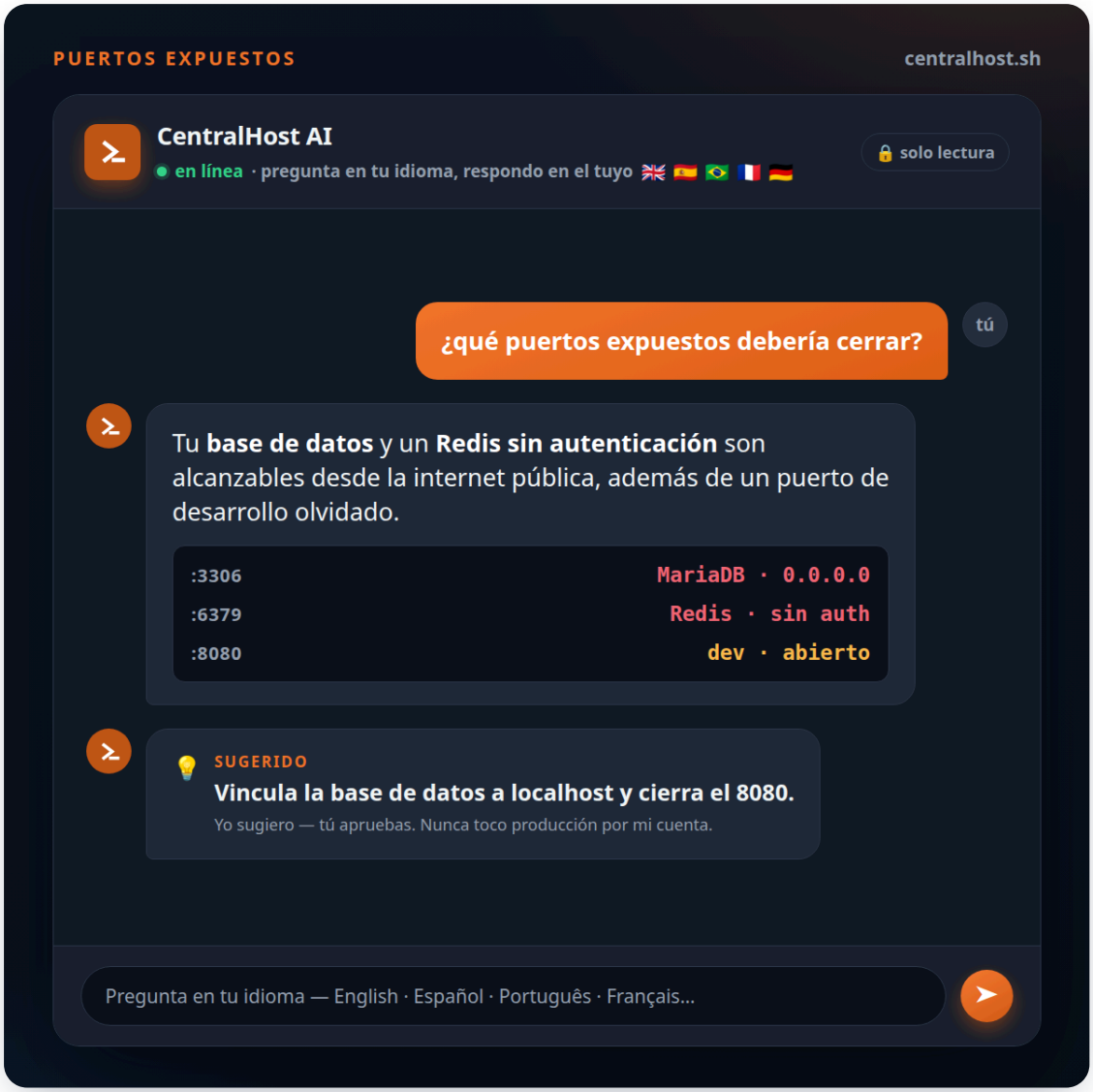
Rastrea el spam saliente hasta el buzón comprometido y propone suspender el correo saliente de la cuenta y forzar el cambio de contraseña — y tú apruebas.

Un sitio que devuelve un 500. En vez de adivinar entre Apache, PHP-FPM y media docena de logs, lo preguntas en una frase:



Correlaciona el fatal de PHP con el límite de memoria y con una extensión que desapareció tras el último upgrade.

Una auditoría de exposición. Justo lo que predican los Pasos 4–8 de esta guía, pero respondido al instante:



Detecta la base de datos y el Redis sin autenticación abiertos a internet, y propone cerrarlos.

Actuar: pídele que haga algo

Hasta aquí el asistente solo ha leído. Cuando le pides que *actúe*, ejecuta comandos en el servidor — pero te muestra exactamente qué va a hacer, espera tu confirmación, y solo entonces ejecuta. Una acción, una aprobación; nunca en lote, nunca en silencio. Algunos ejemplos:

- *"Suspende el correo saliente de la cuenta* `clientxyz` *mientras investigo."*
- *"Mata el proceso que se ejecuta desde /tmp y elimina el archivo."*
- *"Bloquea esta IP en PXF en todos los servidores de mi flota."*
- *"Reinicia PHP-FPM en este servidor."*
- *"Añade un registro DMARC para este dominio."*
- *"Sube* `max_children` *de PHP-FPM de 10 a 20 en este servidor."*

Apéndice II: Terminal web y acceso externo

Un terminal SSH completo, sin exponer el puerto 22

CentralHost incluye un terminal SSH en el navegador que te da una sesión de shell completa en cualquier servidor de tu flota — desde cualquier dispositivo, sin un cliente SSH local, y sin el puerto 22 abierto a internet. La autenticación pasa por CentralHost; el propio servidor nunca recibe una conexión entrante desde la internet pública.

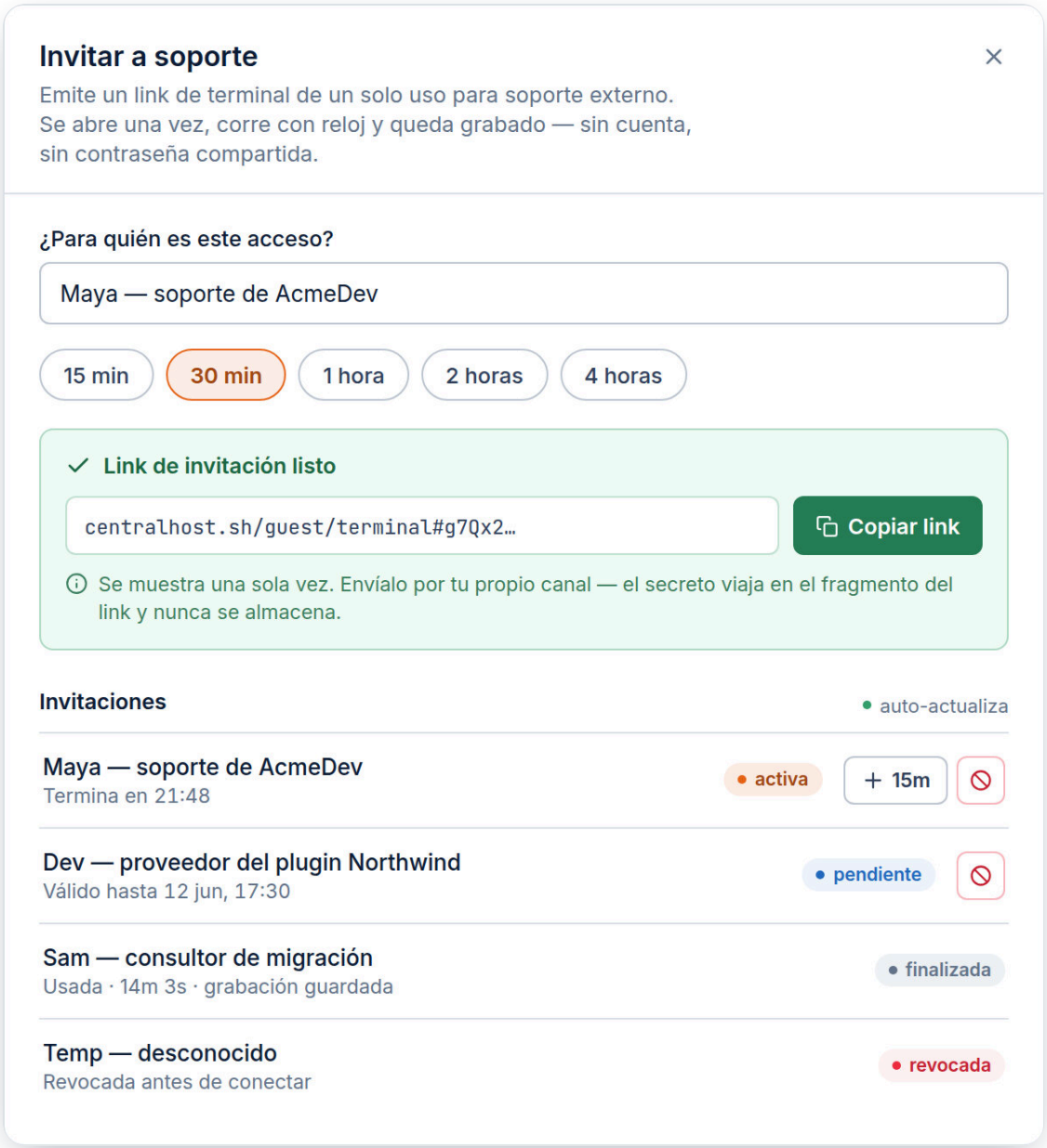
Para las operaciones del día a día esto significa que tu equipo puede trabajar desde cualquier sitio sin requisitos de VPN ni excepciones de firewall. Para la seguridad significa que la superficie de ataque se reduce a un único punto de entrada autenticado en lugar de un puerto abierto en cada servidor.

Invitar a soporte externo

Cuando un incidente requiere ayuda de fuera — un contratista, un especialista o el equipo de soporte de un proveedor — el enfoque estándar es compartir credenciales de root o abrir temporalmente el acceso SSH. Ambos crean riesgo: las credenciales se reutilizan, el acceso no siempre se revoca, y no hay registro de lo que se hizo.

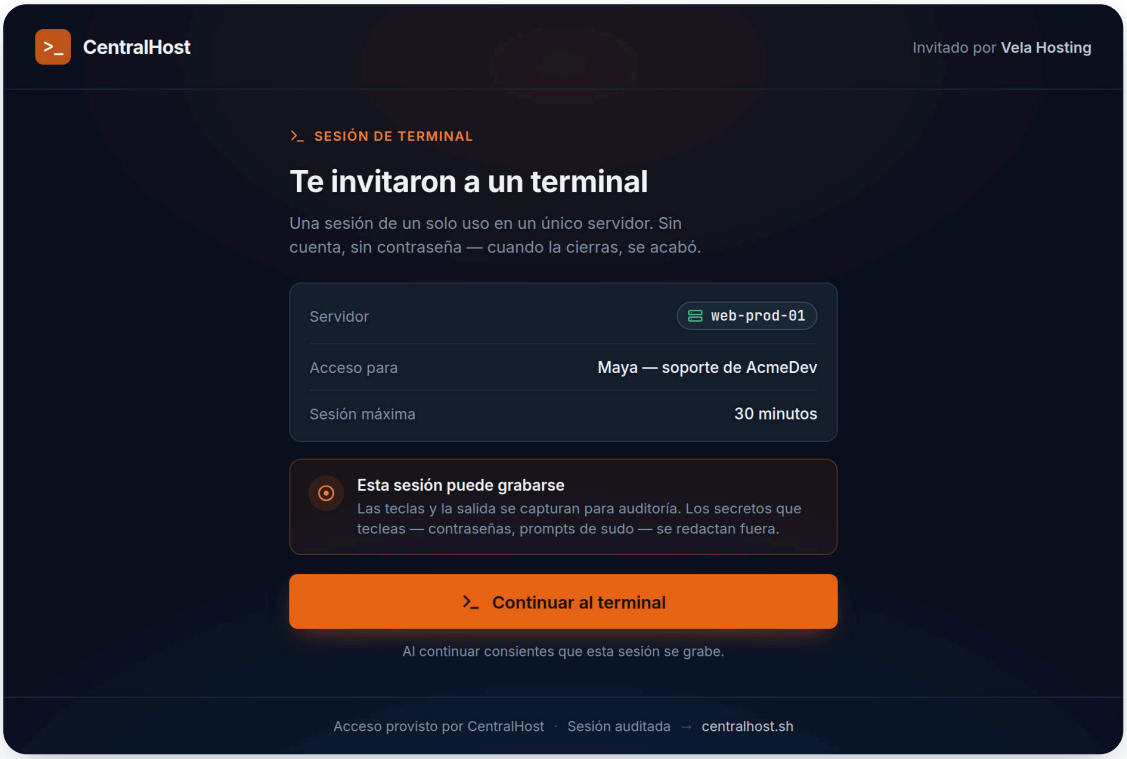
CentralHost maneja esto de forma diferente. Envías un enlace de invitación a la parte externa. Obtienen acceso a una sesión de terminal web acotada al servidor específico que elijas — nada más en tu flota es visible o alcanzable. No se comparten credenciales. No hay que cambiar ninguna regla de firewall.

El enlace de invitación es de un solo uso y la sesión de terminal tiene un límite de tiempo máximo. Una vez que la sesión expira, el acceso desaparece automáticamente — no hay nada que revocar ni nada que olvidarse de limpiar.



Eliges la duración, copias el link de un solo uso y lo envías por tu propio canal — sin compartir credenciales y con la sesión grabada.

Cuando la parte externa abre el enlace, no aterriza en tu panel ni en un formulario de login. Ve una única pantalla de consentimiento que indica quién la invitó, a qué servidor está a punto de acceder, cuánto dura la sesión y que queda grabada — seguida de un solo botón para continuar. Sin cuenta, sin contraseña, y nada más de tu flota a su alcance.



Lo que ve el operador invitado al abrir el enlace: una página sin adornos que nombra al anfitrión, el servidor de destino y el límite de tiempo, con el aviso de grabación y un solo botón para entrar — sin login, sin barra lateral, sin contraseña.

Cada sesión queda grabada

Cada sesión de terminal web — ya sea ejecutada por tu propio equipo o por una parte externa invitada — se graba por completo. La grabación captura:

- Cada comando tecleado y su salida exacta
- La línea temporal completa de la sesión con marcas de tiempo
- La identidad del operador que ejecutó la sesión

Las grabaciones se almacenan y son revisables a posteriori. Si un contratista hace un cambio que causa problemas tres días después, puedes reproducir la sesión y ver exactamente qué se hizo, en qué orden, y qué respondió el servidor. Si un cliente disputa que se cambió una configuración, el registro está ahí.

Para los proveedores de hosting con requisitos de cumplimiento — auditorías de GDPR, procesos de ISO 27001, o simplemente responsabilidad interna — este es el tipo de trazabilidad que normalmente solo está disponible con costosas soluciones dedicadas de PAM (Privileged Access Management). En CentralHost viene incorporado.