Guía 04: Buenas Prácticas de Seguridad y DevOps en LAMP

Objetivo: Consejos de nivel Senior para que tu servidor no sea hackeado en la primera semana y puedas dormir tranquilo.

---

1. Seguridad Básica del Servidor

No uses ROOT

Nunca trabajes como `root` para tareas diarias. Usa un usuario con permisos `sudo`.

```
# Crear usuario
adduser nuevo_usuario
# Darle poderes de sudo
usermod -aG sudo nuevo_usuario
```

Firewall (UFW)

Cierra todo lo que no uses.

```
sudo ufw allow OpenSSH
sudo ufw allow 'Apache Full'
sudo ufw enable
```

SSH Seguro

Edita `/etc/ssh/sshd_config`: - `PermitRootLogin no` (Prohíbe entrar como root directo). - `PasswordAuthentication no` (Usa llaves SSH, es mucho más seguro).

---

2. Seguridad en CMS y Archivos

Permisos Estrictos

El error #1 es poner todo en `777` "para que funcione". **JAMÁS lo hagas**. - Directorios: `755` - Archivos: `644` - `wp-config.php` (o equivalentes con claves): `600` o `640`.

Ocultar Errores en Producción

Que PHP no muestre errores en pantalla (da pistas a hackers). En `php.ini`:

```
display_errors = Off
log_errors = On
error_log = /var/log/php_errors.log
```

---

3. Backups (La regla de oro)

Si no hay backup, el dato no existe.

Backup de Base de Datos (Automatizable)

Crea un script simple `backup_db.sh`:

```
#!/bin/bash
FECHA=$(date +%F)
mysqldump -u usuario_cms -p'tu_password' nombre_db > /home/usuario/backups/db_$FECHA.sql
```

Ponlo en el `cron` (`crontab -e`) para que corra cada noche a las 3 AM: `0 3 * * * /bin/bash /home/usuario/scripts/backup_db.sh`

Backup de Archivos

```
tar -czf /home/usuario/backups/sitio_$FECHA.tar.gz /var/www/mi-sitio
```

---

4. HTTPS (SSL/TLS)

Hoy en día es obligatorio. Usa **Certbot** (Let’s Encrypt) para certificados gratis y automáticos.

```
sudo apt install certbot python3-certbot-apache
sudo certbot --apache
```

- Elige tu dominio.
- Selecciona “Redirect” para forzar HTTPS.
- Certbot configura la renovación automática solo.

---

5. Logs y Rotación

Los logs llenan el disco. Linux usa `logrotate` por defecto, pero revísalo. - Comprueba `/var/log/apache2/` regularmente. - Si ves archivos de 10GB, algo va mal (ataque o error en bucle). - Usa `tail -f` para ver qué pasa en tiempo real.