Guía Completa de Seguridad para VPS

Tabla de Contenidos

---

1. Fail2ban (Esencial)

Instalación y configuración:

```bash
# Instalar fail2ban
sudo apt update
sudo apt install fail2ban

# Habilitar servicio
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
```

Configuración avanzada `/etc/fail2ban/jail.local`:

ini

```ini
[DEFAULT]
# Tiempo de baneo (en segundos)
bantime = 3600
findtime = 600
maxretry = 3

# Notificaciones por email (opcional)
destemail = tu-email@ejemplo.com
sendername = Fail2Ban
action = %(action_mwl)s

# Whitelist de IPs confiables
ignoreip = 127.0.0.1/8 ::1 tu-ip-casa.com

[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 7200
findtime = 600

[apache-auth]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache2/error.log
maxretry = 5

[apache-noscript]
enabled = true
port = http,https
filter = apache-noscript
logpath = /var/log/apache2/error.log

[apache-overflows]
enabled = true
port = http,https
filter = apache-overflows
logpath = /var/log/apache2/error.log

[apache-badbots]
enabled = true
```

```
port = http,https
filter = apache-badbots
logpath = /var/log/apache2/access.log
maxretry = 2

[php-url-fopen]
enabled = true
port = http,https
filter = php-url-fopen
logpath = /var/log/apache2/access.log

[recidive]
enabled = true
filter = recidive
logpath = /var/log/fail2ban.log
action = iptables-allports[name=recidive]
bantime = 604800  # 1 semana
findtime = 86400  # 1 día
maxretry = 3
```

Reiniciar y verificar:

bash

```
# Reiniciar fail2ban
sudo systemctl restart fail2ban

# Ver estado
sudo fail2ban-client status

# Ver jaulas activas
sudo fail2ban-client status sshd

# Ver IPs baneadas
sudo fail2ban-client get sshd banip
```

---

2. Firewall UFW (Uncomplicated Firewall)

```bash
# Instalar UFW
sudo apt install ufw

# Configuración básica
sudo ufw default deny incoming
sudo ufw default allow outgoing

# Permitir SSH (IMPORTANTE: hazlo ANTES de habilitar)
sudo ufw allow ssh
# O puerto personalizado si cambiaste SSH
# sudo ufw allow 2222/tcp

# Permitir HTTP/HTTPS
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Permitir solo desde IP específica (opcional)
# sudo ufw allow from TU_IP_CASA to any port 22

# Habilitar firewall
sudo ufw enable

# Ver estado
sudo ufw status verbose

# Limitar intentos SSH (alternativa a fail2ban)
sudo ufw limit ssh
```

---

3. Cambiar Puerto SSH (Seguridad por oscuridad)

```bash
# Editar configuración SSH
sudo nano /etc/ssh/sshd_config
```

Cambia:

```bash
Port 2222  # O cualquier puerto > 1024
```

Luego:

```bash
# Actualizar firewall
sudo ufw allow 2222/tcp
sudo ufw delete allow 22/tcp

# Reiniciar SSH
sudo systemctl restart sshd
```

---

4. Configuración Avanzada SSH

```bash
# Editar /etc/ssh/sshd_config
sudo nano /etc/ssh/sshd_config
```

Configuración recomendada:

```bash
# Puerto personalizado
Port 2222

# Protocolo SSH 2 únicamente
Protocol 2

# Sin acceso root
PermitRootLogin no

# Solo autenticación por llave
PubkeyAuthentication yes
PasswordAuthentication no
ChallengeResponseAuthentication no
PermitEmptyPasswords no

# Desactivar forwarding innecesario
X11Forwarding no
AllowTcpForwarding no
AllowAgentForwarding no

# Usuarios permitidos (específico)
AllowUsers tu-usuario

# Timeouts de sesión
ClientAliveInterval 300
ClientAliveCountMax 2

# Intentos máximos de autenticación
MaxAuthTries 3
MaxSessions 2

# Configuración de login
LoginGraceTime 30
MaxStartups 2:50:10

# Sin variables de entorno
PermitUserEnvironment no

# Logs verbosos
LogLevel VERBOSE
```

---

5. Actualizaciones Automáticas de Seguridad

```bash
# Instalar unattended-upgrades
sudo apt install unattended-upgrades apt-listchanges

# Configurar
sudo dpkg-reconfigure -plow unattended-upgrades

# Editar configuración
sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
```

Configuración recomendada:

```bash
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    "${distro_id}ESMApps:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
};

Unattended-Upgrade::AutoFixInterruptedDpkg "true";
Unattended-Upgrade::MinimalSteps "true";
Unattended-Upgrade::Remove-Unused-Kernel-Packages "true";
Unattended-Upgrade::Remove-Unused-Dependencies "true";
Unattended-Upgrade::Automatic-Reboot "false";
Unattended-Upgrade::Automatic-Reboot-Time "03:00";

// Notificaciones por email
Unattended-Upgrade::Mail "tu-email@ejemplo.com";
Unattended-Upgrade::MailReport "on-change";
```

---

6. Deshabilitar Servicios Innecesarios

```bash
# Ver servicios activos
systemctl list-units --type=service --state=running

# Deshabilitar servicios innecesarios (ejemplos)
sudo systemctl disable bluetooth
sudo systemctl disable cups
sudo systemctl disable avahi-daemon
```

7. Monitoreo con Logwatch

```bash
# Instalar logwatch
sudo apt install logwatch

# Configurar
sudo nano /etc/logwatch/conf/logwatch.conf
```

Configuración:

```bash
MailTo = tu-email@ejemplo.com
MailFrom = servidor@tudominio.com
Range = yesterday
Detail = High
Service = All
Format = html
```

Ejecutar manualmente:

```bash
sudo logwatch --output mail
```

---

8. Rootkit Hunter (rkhunter)

```bash
# Instalar
sudo apt install rkhunter

# Actualizar base de datos
sudo rkhunter --update

# Escaneo inicial
sudo rkhunter --propupd

# Ejecutar escaneo
sudo rkhunter --check --skip-keypress

# Configurar escaneo automático
sudo nano /etc/default/rkhunter
```

Configurar:

```bash
CRON_DAILY_RUN="true"
CRON_DB_UPDATE="true"
APT_AUTOGEN="true"
```

---

9. ClamAV (Antivirus)

```bash
# Instalar ClamAV
sudo apt install clamav clamav-daemon

# Actualizar definiciones
sudo systemctl stop clamav-freshclam
sudo freshclam
sudo systemctl start clamav-freshclam

# Escaneo manual
sudo clamscan -r -i /home
```

---

10. Monitoreo de Integridad con AIDE

```bash
# Instalar AIDE
sudo apt install aide

# Inicializar base de datos
sudo aideinit

# Mover base de datos
sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db

# Verificar integridad
sudo aide --check
```

---

11. Hardening del Kernel (sysctl)

```bash
# Editar configuración
sudo nano /etc/sysctl.d/99-security.conf
```

Agregar:

bash

```
# Protección contra SYN flood
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_syn_retries = 2
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_max_syn_backlog = 4096

# Protección contra IP spoofing
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Deshabilitar redirecciones ICMP
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0

# No enviar redirecciones ICMP
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Deshabilitar source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# Ignorar pings broadcast
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Ignorar mensajes ICMP malintencionados
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Log de paquetes sospechosos
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Protección contra buffer overflow
kernel.exec-shield = 1
kernel.randomize_va_space = 2
```

Aplicar cambios:

```bash
sudo sysctl -p /etc/sysctl.d/99-security.conf
```

---

12. ModSecurity (WAF para Apache)

```bash
# Instalar ModSecurity
sudo apt install libapache2-mod-security2

# Habilitar módulo
sudo a2enmod security2

# Copiar configuración recomendada
sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf

# Editar configuración
sudo nano /etc/modsecurity/modsecurity.conf
```

Cambiar:

```bash
SecRuleEngine On
```

Instalar reglas OWASP:

```bash
cd /tmp
git clone https://github.com/coreruleset/coreruleset.git
sudo mv coreruleset /etc/modsecurity/
sudo cp /etc/modsecurity/coreruleset/crs-setup.conf.example /etc/modsecurity/coreruleset/crs-setup.conf

# Configurar Apache
sudo nano /etc/apache2/mods-enabled/security2.conf
```

Agregar:

```apache
IncludeOptional /etc/modsecurity/*.conf
IncludeOptional /etc/modsecurity/coreruleset/crs-setup.conf
IncludeOptional /etc/modsecurity/coreruleset/rules/*.conf
```

Reiniciar:

```bash
sudo systemctl restart apache2
```

---

13. Backups Automáticos

```bash
# Script de backup
sudo nano /usr/local/bin/backup-vps.sh
```

Script:

```bash
#!/bin/bash
BACKUP_DIR="/backups"
DATE=$(date +%Y%m%d_%H%M%S)
RETENTION_DAYS=30

mkdir -p $BACKUP_DIR

# Backup de archivos importantes
tar -czf $BACKUP_DIR/etc_$DATE.tar.gz /etc
tar -czf $BACKUP_DIR/www_$DATE.tar.gz /var/www
tar -czf $BACKUP_DIR/home_$DATE.tar.gz /home

# Backup de bases de datos
mysqldump --all-databases | gzip > $BACKUP_DIR/mysql_$DATE.sql.gz

# Limpiar backups antiguos
find $BACKUP_DIR -name "*.tar.gz" -mtime +$RETENTION_DAYS -delete
find $BACKUP_DIR -name "*.sql.gz" -mtime +$RETENTION_DAYS -delete

# Enviar a storage remoto (opcional)
# rsync -avz $BACKUP_DIR/ user@backup-server:/backups/
```

Hacer ejecutable y agendar:

```bash
sudo chmod +x /usr/local/bin/backup-vps.sh

# Agregar a crontab
sudo crontab -e
```

Agregar:

```bash
# Backup diario a las 2 AM
0 2 * * * /usr/local/bin/backup-vps.sh
```

---

14. Monitoreo con Netdata (Opcional)

```bash
# Instalación
bash <(curl -Ss https://my-netdata.io/kickstart.sh)

# Acceso: http://tu-ip:19999
# Configurar firewall
sudo ufw allow 19999/tcp
```

Soluciones para IPs Dinámicas

1. DNS Dinámico (DDNS) - Más Recomendada

Servicios DDNS gratuitos:

- **No-IP** (gratis con confirmación mensual)
- **DuckDNS** (gratis, simple)
- **Dynu** (gratis)
- **FreeDNS** (gratis)

Configuración con DuckDNS:

```bash
# Crear cuenta en https://www.duckdns.org
# Obtener tu token y crear un dominio: mi-casa.duckdns.org

# Instalar cliente
mkdir -p ~/duckdns
cd ~/duckdns
nano duck.sh
```

Script `duck.sh`:

```bash
#!/bin/bash
echo url="https://www.duckdns.org/update?domains=mi-casa&token=tu-token-aqui&ip=" | curl -k -o ~/duck
```

```bash
# Hacer ejecutable
chmod +x duck.sh

# Probar
./duck.sh

# Automatizar con cron (actualizar cada 5 minutos)
crontab -e
```

Agregar:

```bash
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
```

Configurar SSH para permitir solo tu DDNS:

```bash
# Crear script que actualice el firewall
sudo nano /usr/local/bin/update-ssh-access.sh
```

```bash
#!/bin/bash

DDNS_HOSTNAME="mi-casa.duckdns.org"
CURRENT_IP=$(dig +short $DDNS_HOSTNAME | tail -n1)

if [ ! -z "$CURRENT_IP" ]; then
    # Eliminar regla anterior si existe
    ufw delete allow from any to any port 22 2>/dev/null

    # Agregar nueva regla con IP actual
    ufw allow from $CURRENT_IP to any port 22

    echo "$(date): SSH access updated for IP: $CURRENT_IP" >> /var/log/ssh-access-update.log
fi
```

```bash
# Hacer ejecutable
sudo chmod +x /usr/local/bin/update-ssh-access.sh

# Ejecutar cada hora
sudo crontab -e
```

Agregar:

```bash
0 * * * * /usr/local/bin/update-ssh-access.sh
```

---

2. VPN (Wireguard) - Más Segura

Instalación Wireguard:

```bash
# Instalar Wireguard
sudo apt install wireguard

# Generar llaves del servidor
cd /etc/wireguard
sudo wg genkey | sudo tee privatekey | wg pubkey | sudo tee publickey

# Configurar servidor
sudo nano /etc/wireguard/wg0.conf
```

Configuración servidor:

```ini
[Interface]
Address = 10.0.0.1/24
ListenPort = 51820
PrivateKey = SERVIDOR_PRIVATE_KEY_AQUI

# Reglas de forwarding
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERAD
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUER

[Peer]
# Tu dispositivo
PublicKey = CLIENTE_PUBLIC_KEY_AQUI
AllowedIPs = 10.0.0.2/32
```

Configuración cliente (tu PC):

```ini
[Interface]
PrivateKey = CLIENTE_PRIVATE_KEY_AQUI
Address = 10.0.0.2/24
DNS = 1.1.1.1

[Peer]
PublicKey = SERVIDOR_PUBLIC_KEY_AQUI
Endpoint = tu-servidor.com:51820
AllowedIPs = 10.0.0.0/24
PersistentKeepalive = 25
```

Habilitar VPN:

```bash
# Habilitar forwarding
sudo nano /etc/sysctl.conf
```

Descomentar:

```bash
net.ipv4.ip_forward=1
```

```bash
# Aplicar
sudo sysctl -p

# Iniciar Wireguard
sudo systemctl enable wg-quick@wg0
sudo systemctl start wg-quick@wg0

# Firewall
sudo ufw allow 51820/udp
sudo ufw allow from 10.0.0.0/24 to any port 22
```

---

3. Port Knocking - Seguridad por Oscuridad

```bash
# Instalar knockd
sudo apt install knockd

# Configurar
sudo nano /etc/knockd.conf
```

Configuración:

```ini
[options]
  UseSyslog

[openSSH]
  sequence    = 7000,8000,9000
  seq_timeout = 5
  command     = /usr/sbin/ufw allow from %IP% to any port 22
  tcpflags    = syn

[closeSSH]
  sequence    = 9000,8000,7000
  seq_timeout = 5
  command     = /usr/sbin/ufw delete allow from %IP% to any port 22
  tcpflags    = syn
```

```bash
# Habilitar
sudo systemctl enable knockd
sudo systemctl start knockd
```

Usar desde cliente:

```bash
# Instalar knock
sudo apt install knockd

# Abrir puerto SSH
knock tu-servidor.com 7000 8000 9000

# Conectar
ssh usuario@tu-servidor.com

# Cerrar puerto
knock tu-servidor.com 9000 8000 7000
```

Wireguard VPN con Respaldo de Emergencia

Escenarios de Pérdida de Acceso con Wireguard

1. Error en Configuración (Más Común)

Si configuras mal y bloqueas SSH antes de probar VPN, no puedes entrar ni por SSH ni VPN si falla.

2. Problema con el Puerto UDP

- Algunos ISPs bloquean puertos UDP altos
- Firewall del router podría bloquear el puerto
- NAT podría fallar

3. Reinicio del Servidor

Si Wireguard no inicia automáticamente después de un reinicio.

4. Actualización del Kernel

- Módulos de Wireguard pueden no cargarse
- Incompatibilidad temporal

5. Cambio de IP del Servidor

Tu endpoint deja de funcionar y no puedes conectarte ni actualizar.

6. Error en Reglas de Firewall

Si accidentalmente bloqueas todo el tráfico.

Configuración Segura con Respaldos

Implementación de Acceso Multinivel:

bash

```
# /etc/ssh/sshd_config
# Configuración para acceso principal
Match Address 10.0.0.0/24
    Port 2222
    PermitRootLogin no
    PasswordAuthentication no
    PubkeyAuthentication yes

# Configuración para acceso de emergencia
Match Address *,!10.0.0.0/24
    Port 2223
    PermitRootLogin no
    PasswordAuthentication no
    PubkeyAuthentication yes
    AuthenticationMethods publickey,keyboard-interactive
```

Recomendación Final: Enfoque de "Cinturón y Tirantes"

Configuración Óptima:

plaintext

```
┌─────────────────────────────────────────────────────┐
│          NIVELES DE ACCESO                          │
├─────────────────────────────────────────────────────┤
│ 1. VPN Wireguard → SSH:2222 (Principal)             │
│    ✓ Máxima seguridad                               │
│    ✓ Uso diario                                     │
├─────────────────────────────────────────────────────┤
│ 2. SSH Directo → SSH:2223 (Emergencia)              │
│    ✓ Fail2ban agresivo                              │
│    ✓ Solo para emergencias                          │
├─────────────────────────────────────────────────────┤
│ 3. Consola Web del Proveedor (Último)               │
│    ✓ Siempre disponible                             │
│    ✓ Recuperación total                             │
└─────────────────────────────────────────────────────┘
```

Mejores Prácticas:

1. **NUNCA** cierres tu sesión SSH actual hasta probar los nuevos métodos
2. **SIEMPRE** mantén 2+ métodos de acceso
3. **PRUEBA** en una nueva terminal antes de confirmar cambios
4. **DOCUMENTA** todos los puertos y contraseñas
5. **GUARDA** las configuraciones en lugar seguro offline

Opciones de Implementación

Opción A: Conservador (Recomendado)

- SSH emergencia en puerto 2223 (siempre disponible)
- Wireguard como acceso principal
- Fail2ban ultra agresivo en emergencia

Opción B: Balanceado

- DDNS + Fail2ban agresivo
- Sin VPN
- Más simple, menos seguro pero sin riesgo de bloqueo

Opción C: Máxima Seguridad con Respaldo

- Wireguard + SSH emergencia + Port Knocking
- Múltiples capas
- Más complejo

Comandos de Verificación Útiles

```bash
# Ver estado de servicios de seguridad
systemctl status fail2ban
systemctl status ufw
systemctl status wg-quick@wg0

# Ver logs de seguridad
sudo tail -f /var/log/auth.log
sudo tail -f /var/log/fail2ban.log

# Ver conexiones activas
sudo ss -tulpn
sudo netstat -tulpn

# Ver reglas de firewall
sudo ufw status numbered
sudo iptables -L -n -v

# Ver IPs baneadas
sudo fail2ban-client status sshd

# Probar conectividad
ping -c 4 tu-servidor.com
nc -zv tu-servidor.com 22

# Ver procesos de red
sudo lsof -i -P -n
```

Documento generado: Septiembre 2025
Versión: 1.0
Última actualización: 30/09/2025