Server Hardening Checklist

The essential security checklist for every new VPS

by Amine | byte-guard.net

2026 Edition

This checklist covers the critical security steps every server needs.
Print it, check off each item, and sleep better knowing your VPS is locked down.

How to Use This Checklist

This checklist is designed to be printed and pinned next to your monitor. Every time you spin up a new VPS, work through each section in order. The commands are written for Ubuntu/Debian (apt-based). If you run RHEL/Rocky/Alma, swap apt for dnf and ufw for firewalld.

Each section includes the exact commands to run and a checkbox to mark completion. The entire process takes about 10 minutes on a fresh server.

For the full walkthrough with explanations, visit:
https://blog.byte-guard.net/harden-linux-vps-10-minutes/

1. System Updates

- [ ] Update package index and install all available updates
```
apt update && apt upgrade -y
```
- [ ] Verify no packages are held back
```
apt list --upgradable
```

TIP: Run this on every fresh server BEFORE doing anything else.

2. Non-Root User Setup

- [ ] Create a non-root user with sudo access
```
adduser <username> && usermod -aG sudo <username>
```
- [ ] Test sudo access from the new user
```
su - <username> && sudo whoami  # should print "root"
```
- [ ] Set a strong password (or generate one)
```
pwgen -s 32 1
```

TIP: Never use root for daily work. A compromised root = total control lost.

3. SSH Key Authentication

- [ ] Generate Ed25519 key on LOCAL machine (if not done)
```
ssh-keygen -t ed25519 -C "your_email@example.com"
```
- [ ] Copy public key to the server
```
ssh-copy-id <username>@<server-ip>
```
- [ ] Test key-based login from a NEW terminal
```
ssh <username>@<server-ip>
```

TIP: Ed25519 is faster and more secure than RSA. Use it as default.

4. SSH Configuration Hardening

- [ ] Edit SSH config

```
sudo vim /etc/ssh/sshd_config
```

- [ ] Set: PermitRootLogin no
- [ ] Set: PasswordAuthentication no
- [ ] Set: PubkeyAuthentication yes
- [ ] Set: ChallengeResponseAuthentication no
- [ ] Set: UsePAM no
- [ ] Reload SSH (keep current session open!)

```
sudo systemctl reload sshd
```

- [ ] Test login from a NEW terminal before closing old one

TIP: NEVER close your current session until you verify key login works in a new terminal.

5. UFW Firewall

- [ ] Install UFW

```
sudo apt install ufw -y
```

- [ ] Set default policies

```
sudo ufw default deny incoming && sudo ufw default allow outgoing
```

- [ ] Allow SSH

```
sudo ufw allow OpenSSH
```

- [ ] Allow HTTP/HTTPS (if running web services)

```
sudo ufw allow 80/tcp && sudo ufw allow 443/tcp
```

- [ ] Enable firewall

```
sudo ufw enable
```

- [ ] Verify rules

```
sudo ufw status verbose
```

TIP: Only open ports you actively use. Every open port is attack surface.

6. Fail2Ban

- [ ] Install and enable Fail2Ban

```
sudo apt install fail2ban -y && sudo systemctl enable --now fail2ban
```

- [ ] Verify SSH jail is active

```
sudo fail2ban-client status sshd
```

- [ ] Create local override for stricter settings

```
sudo vim /etc/fail2ban/jail.local
```

- [ ] Set: maxretry = 3, findtime = 10m, bantime = 1h
- [ ] Restart Fail2Ban

```
sudo systemctl restart fail2ban
```

7. Unattended Security Upgrades

- [ ] Install unattended-upgrades

```
sudo apt install unattended-upgrades -y
```

- [ ]

Enable automatic security updates

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

- [ ] Verify it is running

```
sudo systemctl status unattended-upgrades
```

8. Final Verification

- [ ] Verify root login and password auth are disabled

```
sudo sshd -T | grep -E "permitrootlogin|passwordauthentication"
```

- [ ] Verify firewall is active with correct rules

```
sudo ufw status
```

- [ ] Verify Fail2Ban SSH jail is running

```
sudo fail2ban-client status sshd
```

- [ ] Verify unattended upgrades are active

```
sudo systemctl is-active unattended-upgrades
```

TIP: Run this verification block on every server. Pin it to your workflow.

9. Bonus: Optional Hardening

- [ ] Change SSH port (reduces log noise)

```
sudo vim /etc/ssh/sshd_config  # set Port 2222
```

- [ ] Update UFW for new SSH port

```
sudo ufw delete allow OpenSSH && sudo ufw allow 2222/tcp
```

- [ ] Reload SSH

```
sudo systemctl reload sshd
```

- [ ] Add SSH config alias on local machine for convenience
- [ ] Set up SSH config (~/.ssh/config) for easy access

```
Host my-vps
  HostName <ip>
  User <username>
  Port 2222
  IdentityFile ~/.ssh/id_ed25519
```

10. Going Further

This checklist covers the essentials. For deeper hardening, check out these guides on byte-guard.net:

> SSH Hardening Guide

`https://blog.byte-guard.net/ssh-hardening-guide/`

> Docker Security Best Practices

`https://blog.byte-guard.net/docker-security-best-practices/`

> Fail2Ban Setup Guide

`https://blog.byte-guard.net/fail2ban-setup-guide/`

> Security Headers Check

`https://blog.byte-guard.net/check-security-headers/`

> OWASP Top 10 Explained

`https://blog.byte-guard.net/owasp-top-10-explained/`

Found this useful?

Subscribe for weekly security & self-hosting guides:

https://blog.byte-guard.net/#/portal/

Follow ByteGuard:

Twitter: @byte_guard_blog

Dev.to: dev.to/byte-guard

GitHub: github.com/byteguard-blog