VectraOps

Knowledge Library

CHECKLIST

Ubuntu Security Hardening Checklist for Production Servers

A practical, verifiable Ubuntu security hardening checklist for production servers. Use it to validate access controls, patching, network exposure, logging, and recovery readiness before go-live.

Operating Systems - July 04, 2026

Purpose

Production Ubuntu servers fail most often because of small, preventable gaps: unnecessary services, overly broad access, missing patch discipline, weak logging, or a recovery process that no one has actually validated. This checklist is designed to catch those issues before a server is exposed to real traffic.

Use this checklist to verify whether a production Ubuntu host is hardened enough for your environment, capture evidence for audit or change review, and decide whether the server is ready to move forward. It is intentionally operational: every item is something you can confirm, review, validate, document, assign, or test.

How to use this checklist

Read each phase in order and treat it like a change-control review. For every checked item, capture evidence, name the owner, and decide whether the control is passing, partially implemented, or failing.

A practical rule: if you cannot produce evidence quickly, the control is not yet production-ready. When in doubt, test the failure case as well as the happy path.

1) Baseline and scope confirmation

This phase defines what the server is supposed to do and what must be protected. Without a clear baseline, hardening can break required services or leave unknown exposures in place.

Checklist items

- ? Confirm the Ubuntu release, kernel version, and support status installed on the server.
- ? Review the server role, environment, and required network-facing services before changing any security setting.
- ? Validate that only approved packages are installed for the server role.
- ? Document the administrative access method, bastion path, and approved emergency access process.
- ? Confirm whether the server is physical, virtual, cloud-based, or container-hosting, because hardening dependencies can differ.
- ? Assign an owner for patching, firewall changes, logging, and recovery validation.

Evidence to capture

- lsb_release -a or /etc/os-release output
- uname -r output
- Package inventory for the host
- Server role definition and approved service list
- Access matrix showing approved users, groups, and escalation path

Acceptance criteria

- The operating system version is known and supported for production use.

- The server role is documented and matches the installed service set.
- Ownership for security operations is explicit and current.

Owner and review cadence

- Owner: platform or infrastructure engineer
- Review cadence: at build time, then whenever the server role changes

Common mistakes

- Hardening a host before the service inventory is complete
- Assuming every server in a fleet has the same role or exposure
- Leaving emergency access undefined until an outage occurs

2) Patch, kernel, and package hygiene

Unpatched packages and unnecessary software are common sources of avoidable risk. This phase verifies that the host can receive updates predictably and that it only contains what it needs.

Checklist items

- ? Confirm automatic security updates are enabled or document the manual patch process with a defined SLA.
- ? Review installed packages and remove software that is not required for the server role.
- ? Validate that the system is current on security updates before production cutover.
- ? Test that the server can be patched and rebooted within the maintenance window without breaking the application.
- ? Document the rollback plan if a security update causes service instability.

- ? Verify that kernel updates are monitored and planned for, including reboot coordination where required.

If you need a repeatable way to audit package patch status across servers, pair this checklist with a Bash Script to Automate Ubuntu Security Patch Audits workflow and use the output as evidence during review.

Evidence to capture

- apt security update status or patch audit output
- Package list before and after cleanup
- Change record for update testing
- Maintenance window approval and rollback notes

Acceptance criteria

- Security updates are applied within the defined maintenance SLA.
- Unused packages and services are removed or explicitly justified.
- Reboot impact has been tested or formally assessed.

Owner and review cadence

- Owner: system engineer or platform engineer
- Review cadence: daily or weekly patch review, plus every release window

Common mistakes

- Treating package cleanup as optional because the service is already running
- Applying updates without reboot planning for kernel-dependent fixes

- Failing to verify that unattended updates are actually configured as intended

3) Access control and privilege management

A hardened server is only as secure as its admin model. This phase checks that access is minimal, traceable, and recoverable if an account is compromised.

Checklist items

- ? Confirm that direct root login is disabled or tightly controlled and documented.
- ? Review sudo access and ensure only approved users and groups have elevated privileges.
- ? Validate that SSH key-based authentication is used for administrative access where possible.
- ? Disable password-based admin access if your operational model allows it and the change has been tested.
- ? Review local accounts and remove stale, shared, or unused administrative users.
- ? Assign named ownership for privileged account review and offboarding.
- ? Test that emergency access works without bypassing audit controls.

Evidence to capture

- /etc/ssh/sshd_config or managed SSH configuration state
- sudoers review notes or configuration management output
- Approved admin account list
- Authentication test result from a non-production session

Acceptance criteria

- Only approved administrators can gain elevated access.
- Access changes are logged and attributable to named users.
- Emergency access exists, but it is restricted and reviewable.

Owner and review cadence

- Owner: security engineer or Linux operations lead
- Review cadence: monthly and after any personnel change

Common mistakes

- Keeping shared admin accounts because they are convenient
- Leaving password authentication enabled without a clear reason
- Forgetting to remove temporary access after a change window

4) Network exposure and firewall control

Every open port should have a reason, an owner, and a validation path. This phase verifies that the host exposes only the services it must provide and that the firewall rules match the intended design.

Checklist items

- ? Review listening services and confirm each exposed port is required for the server role.
- ? Validate that the host firewall permits only approved inbound traffic.
- ? Confirm that outbound restrictions exist where your environment requires them.
- ? Test remote access from approved source networks before production use.
- ? Document any exceptions, temporary openings, or upstream security group dependencies.

- ? Reconcile the host firewall with any cloud, virtual network, or perimeter firewall controls so they do not conflict.

If you need a safe, repeatable workflow for host firewall setup and validation, use How to Configure UFW Firewall Rules on Ubuntu Server as the implementation reference and verify the allowed services against this checklist.

Evidence to capture

- Listening socket inventory from ss -tulpn or equivalent
- Firewall ruleset export
- Remote connectivity test results from approved IP ranges
- Exception register for temporary openings

Acceptance criteria

- No unexpected listening services remain exposed.
- Only approved sources and destinations can reach required services.
- Network tests succeed for intended traffic and fail for disallowed traffic.

Owner and review cadence

- Owner: network/security engineer or platform engineer
- Review cadence: at build time and after any firewall or service change

Common mistakes

- Allowing a port because an application team ?might need it later?
- Testing connectivity only from the server itself, not from the real source network

- Forgetting that upstream controls can hide a misconfigured host firewall

5) Authentication, SSH, and session hardening

SSH is usually the primary administrative entry point, so weak SSH settings can undo other controls. This phase checks the session path end to end.

Checklist items

- ? Confirm that the SSH service uses approved protocols and ciphers for your baseline.
- ? Review SSH timeout, idle session, and login banner settings where required by policy.
- ? Validate that root login over SSH is disabled or explicitly governed.
- ? Test that administrative logins require the intended authentication factors.
- ? Document whether MFA is enforced through your access layer, bastion, or identity platform.
- ? Review whether X11 forwarding, agent forwarding, and other convenience features are disabled when not required.

Evidence to capture

- SSH configuration review output
- Successful and failed login test results
- Policy reference for session settings and MFA requirements

Acceptance criteria

- The SSH configuration matches the approved hardening baseline.
- Authentication behavior aligns with the documented administrative access model.

- Convenience features are not enabled unless they are required and justified.

Owner and review cadence

- Owner: security engineer
- Review cadence: at build time and after SSH-related changes

Common mistakes

- Changing SSH settings without a fallback console path
- Enforcing a stronger policy before testing access from the real admin workflow
- Leaving forwarding features on because they were present in the default config

6) Logging, audit, and time synchronization

If you cannot reconstruct who did what and when, you cannot reliably investigate incidents. This phase ensures the host produces trustworthy logs and that timestamps can be correlated.

Checklist items

- ? Confirm that system logs are being collected centrally or retained locally for the required period.
- ? Validate that authentication, sudo, and service logs are available for review.
- ? Review audit coverage for sensitive actions, privilege changes, and administrative access where required.
- ? Test that the host time is synchronized with a trusted time source.
- ? Document log ownership, retention, and alerting responsibility.
- ? Verify that log rotation and disk usage are monitored so logging does not fail silently.

Evidence to capture

- Log forwarding configuration or collector receipt
- journalctl or log review output
- Time sync status
- Audit policy or audit rule set
- Retention and alerting documentation

Acceptance criteria

- Critical security and administrative events are recorded and available for review.
- Timestamps are aligned closely enough for incident correlation.
- Log storage and forwarding are monitored for failure.

Owner and review cadence

- Owner: security operations or platform operations
- Review cadence: continuous monitoring with monthly validation

Common mistakes

- Forwarding logs but never checking whether the collector is receiving them
- Ignoring time sync because the server ?looks correct? locally
- Keeping logs too short for incident investigation or compliance needs

7) Service minimization and application boundary control

A hardened server should run only the services required for its production function. This phase focuses on reducing the attack surface created by daemons, scheduled jobs, and local application behavior.

Checklist items

- ? Review enabled services and disable anything not required by the server role.
- ? Validate that scheduled jobs, timers, and background processes are documented and owned.
- ? Confirm that application user accounts run with the minimum required privileges.
- ? Test that services fail safely when a dependency is missing or unavailable.
- ? Document which ports, files, and directories each application component needs.
- ? Review temporary files, caches, and upload locations for access control and cleanup behavior.

Evidence to capture

- Service inventory from systemctl
- Job and timer inventory
- Application run-as user mapping
- Dependency and failure test results

Acceptance criteria

- Only required services remain enabled.

- Application processes do not run with unnecessary privilege.
- Service failure behavior is known and acceptable.

Owner and review cadence

- Owner: application owner with platform review
- Review cadence: at deployment and after application changes

Common mistakes

- Leaving support daemons enabled because disabling them seems harmless
- Running a service as root when a dedicated user would work
- Not documenting scheduled jobs, then being surprised by production activity

8) File permissions, secrets, and sensitive data handling

Hardening is incomplete if secrets, keys, and sensitive configuration are readable by the wrong users. This phase checks local data exposure and permission drift.

Checklist items

- ? Review ownership and permissions on configuration files, keys, certificates, and application secrets.
- ? Validate that sensitive files are not world-readable or group-readable without justification.
- ? Confirm that SSH keys, API tokens, and service credentials are stored and rotated according to policy.

- ? Document where secrets live, who can read them, and how they are rotated.
- ? Test backup access controls for sensitive configuration and private keys.
- ? Remove leftover installer artifacts, staging files, and old credentials from the host.

Evidence to capture

- Permissions listing for sensitive paths
- Secret inventory or vault reference
- Rotation records or change tickets
- Backup access review results

Acceptance criteria

- Sensitive files are restricted to approved users and groups only.
- Secret storage and rotation follow the documented standard.
- Backups do not widen access to protected material.

Owner and review cadence

- Owner: application owner and security engineer
- Review cadence: at deployment and during credential rotation events

Common mistakes

- Fixing one file permission while missing related private keys or config directories
- Backing up secrets without reviewing who can restore them
- Leaving deployment artifacts with credentials in home directories or temp paths

9) Backup, recovery, and restore validation

A server is not production-ready if you cannot restore it to a known good state. This phase checks both backup availability and actual restore confidence.

Checklist items

- ? Confirm that the server is included in the backup scope for system state, configuration, and required application data.
- ? Review backup frequency, retention, and encryption requirements.
- ? Test a restore of at least one critical configuration item or dataset.
- ? Validate that the restore target has the same or compatible access controls.
- ? Document recovery point objective and recovery time objective expectations.
- ? Assign ownership for restore testing and incident-time recovery.

Evidence to capture

- Backup job status and retention policy
- Restore test record
- Recovery objective documentation
- Encryption and access control evidence for backup storage

Acceptance criteria

- Backups exist for all required data and configuration.
- At least one restore has been validated recently.
- Recovery expectations are explicit and realistic.

Owner and review cadence

- Owner: infrastructure or backup operations
- Review cadence: per backup cycle with quarterly restore validation

Common mistakes

- Assuming backups work because the job reports success
- Backing up data but not configuration, keys, or service definitions
- Never testing restore access until after a failure

10) Readiness scoring and go-live decision

Use a simple score to decide whether the host is ready for production. This is not a substitute for judgment, but it helps identify whether the remaining risk is acceptable.

Scoring method

For each checklist item:

- Pass = 2 points
- Partial = 1 point
- Fail = 0 points

Score the phases together, then calculate the percentage of points achieved.

Interpretation

- 90?100%: Ready for production, with normal operational monitoring
- 75?89%: Conditionally ready; document exceptions and complete remediation within a defined window
- Below 75%: Not ready for production; block go-live until critical gaps are fixed

Pass/fail decision rules

- Treat any failed item in access control, firewall exposure, logging, or backup restore validation as a production blocker unless a formal risk acceptance exists.
- Treat undocumented exceptions as failures.
- Treat untested controls as partial, not pass.

Evidence to capture

- Scoring sheet or review record
- Exception approvals
- Risk acceptance documentation for unresolved items

Acceptance criteria

- The score meets the agreed production threshold.
- Any remaining gaps are explicitly accepted, time-bound, and owned.
- No critical control is left unverified.

Owner and review cadence

- Owner: change owner or service owner
- Review cadence: at every production build or major security review

Common mistakes that weaken hardening reviews

These issues appear repeatedly in production environments and are worth checking explicitly:

- Relying on defaults instead of an approved baseline
- Changing one control without validating dependent controls
- Collecting evidence after the fact instead of during the review
- Accepting verbal approval for firewall or privilege exceptions
- Skipping restore tests because the backup dashboard shows green
- Letting temporary access or temporary rules remain in place after the change window

Practical acceptance summary

A production Ubuntu server is hardening-ready when the baseline is known, only required services are installed and exposed, administrative access is tightly controlled, logs are usable, backups can be restored, and all exceptions are documented with owners and deadlines.

If the server cannot be patched predictably, cannot be reached only from approved sources, or cannot be restored with confidence, it is not ready for production. A hardened system is not one with the most settings enabled; it is one whose security controls are verified, owned, and operationally maintainable.

Use this guidance together with Windows 11 BitLocker and Secure Boot hardening to connect the workflow with related operational context already available on the site.