VectraOps

Knowledge Library

CHECKLIST

Big Data Security Checklist for Hadoop Cluster Hardening

Use this practical Hadoop security checklist to verify identity, network, data, audit, and operations controls before production use. Each phase includes evidence, acceptance criteria, owners, and review cadence so you can assess readiness with confidence.

Programming - July 05, 2026

Purpose

Hadoop clusters fail in production for security reasons in predictable ways: overly broad access, exposed management ports, weak service identities, unencrypted data paths, and missing audit evidence. This checklist is designed to help you harden a Hadoop environment before it becomes a production liability. After using it, you should be able to decide whether the current controls are sufficient, verify the most important security dependencies, and capture the evidence needed for a production readiness review.

Use this checklist as a verification tool, not a policy document. It focuses on what you can confirm, document, test, and assign during a real hardening review.

How to use this checklist

Work through the phases in order. For each item, capture evidence, note the owner, and record whether the control is implemented, partially implemented, or missing. If a phase has unresolved failures, treat the cluster as not production-ready until the acceptance criteria are met.

A practical approach is to pair this checklist with your broader release gate, such as a Big Data Production Readiness Checklist review, so security findings are not isolated from data, scaling, and rollback readiness.

1. Identity, authentication, and administrative access

Purpose: Confirm that every human and service identity is authenticated, authorized, and traceable, with no shared admin shortcuts.

Checklist items

- ? Confirm that Kerberos or an equivalent centralized authentication mechanism is enabled for core Hadoop services.
- ? Review whether all service principals use unique identities and non-shared keytabs or secrets.
- ? Validate that administrator access requires named accounts, not generic shared logins.
- ? Document which groups or roles can administer HDFS, YARN, Hive, ZooKeeper, and cluster hosts.
- ? Assign least-privilege permissions for operators, developers, analysts, and automation accounts.
- ? Test that expired or revoked credentials are rejected across the cluster.
- ? Confirm that privileged access is time-bound or approved through a documented process.

Evidence to capture

Capture authentication configuration files, principal-to-service mappings, role definitions, and a sample access-denied event from a disabled or unauthorized identity. Save screenshots or command output that show account separation and the absence of shared administrative credentials.

Acceptance criteria

The cluster uses centralized authentication, service identities are unique, privileged access is role-based, and revoked credentials fail as expected. No critical service should rely on a shared secret that cannot be rotated independently.

Owner and review cadence

Owner: Platform security lead and Hadoop administrator Review cadence: At initial hardening, after every privilege change, and quarterly

Common mistakes

- Using one keytab for multiple daemons
- Leaving temporary admin accounts active after migration
- Granting broad filesystem permissions to automation jobs
- Forgetting to test authentication failure paths after rotation

2. Network exposure and cluster perimeter

Purpose: Reduce the attack surface by limiting where management, control, and data services can be reached.

Checklist items

- ? Confirm that all management interfaces are bound to trusted administrative networks only.

- ? Review firewall rules for NameNode, DataNode, ResourceManager, NodeManager, ZooKeeper, and query services.
- ? Validate that only required ports are exposed between cluster nodes and from client networks.
- ? Document which subnets, jump hosts, or bastions are allowed to reach administrative endpoints.
- ? Test that unauthorized source addresses are blocked at the network layer.
- ? Assign ownership for ingress, egress, and east-west traffic rules.
- ? Confirm that any public-facing access path is intentional, approved, and logged.

Evidence to capture

Capture firewall policy exports, routing or security group rules, port scans from allowed and disallowed networks, and the documented list of approved administrative sources. If your environment uses a service mesh, load balancer, or host-based firewall, include the exact enforcement point.

Acceptance criteria

Only required services are reachable from approved sources, management endpoints are not broadly exposed, and unauthorized connections are blocked consistently. Any externally reachable service must have an explicit business justification and compensating controls.

Owner and review cadence

Owner: Network security engineer and platform operations lead Review cadence: At hardening, after every network change, and monthly

Common mistakes

- Allowing broad inbound access for ?temporary testing? that never gets removed
- Exposing administrative UIs on shared application subnets
- Forgetting to validate east-west traffic between worker nodes
- Treating host firewalls and perimeter firewalls as interchangeable controls

3. Encryption in transit and at rest

Purpose: Ensure data cannot be read or modified by unauthorized parties while moving through the cluster or sitting on disk.

Checklist items

- ? Confirm that TLS is enabled for client-to-service and service-to-service traffic where supported.
- ? Review whether HDFS, YARN, Hive, and related services enforce encrypted connections for sensitive endpoints.
- ? Validate that certificates are issued, stored, and rotated through a controlled process.
- ? Document key ownership, rotation intervals, and renewal alerts.
- ? Test that plaintext fallback is disabled when encrypted transport is required.
- ? Assign responsibility for encryption of disks, volumes, object stores, backups, and snapshots.
- ? Confirm that encryption-at-rest settings apply to new data and not just existing volumes.

Evidence to capture

Capture TLS configuration, certificate chains, cipher policy or equivalent transport settings, and storage encryption documentation. Include proof that a sample client connection negotiates encryption and that plaintext access is rejected where required.

Acceptance criteria

Required data paths use encryption, certificates are managed through a repeatable process, and stored data is encrypted according to policy. Backups and snapshots must be included, not treated as exceptions.

Owner and review cadence

Owner: Security engineering and storage operations Review cadence: At deployment, during certificate rotation, and semiannually

Common mistakes

- Encrypting only client traffic while leaving internal service traffic unprotected
- Forgetting to include backup repositories and snapshot targets
- Allowing expired certificates to cause emergency bypasses
- Assuming encryption-at-rest is enabled because a volume type supports it

4. Authorization, file system controls, and data segregation

Purpose: Prevent users and jobs from accessing data outside their approved scope.

Checklist items

- ? Confirm that filesystem permissions are reviewed for top-level HDFS directories and shared service paths.

- ? Review whether service-specific ACLs or authorization plugins are enabled where needed.
- ? Validate that sensitive datasets are separated by project, tenant, or environment.
- ? Document who can read, write, create, delete, and administer each protected area.
- ? Test that a non-privileged user cannot list or read restricted directories.
- ? Assign explicit ownership for temporary directories, staging areas, and export locations.
- ? Confirm that default umask or creation permissions do not expose new files broadly.

Evidence to capture

Capture directory listings, ACL exports, role mappings, and failed access tests from a low-privilege account. Include sample evidence for restricted tables, queues, or storage buckets if the cluster integrates with external data stores.

Acceptance criteria

Protected data is segregated, permissions follow least privilege, and unauthorized read/write attempts fail. Shared directories do not become a back door around service-level controls.

Owner and review cadence

Owner: Data platform admin and dataset owners Review cadence: At onboarding of each new dataset, after permission changes, and monthly

Common mistakes

- Relying on directory structure alone without verifying ACL behavior
- Letting temporary data land in overly permissive shared folders

- Granting write access when read-only access would satisfy the use case
- Failing to review inherited permissions on newly created paths

5. Auditing, logging, and traceability

Purpose: Ensure security-relevant events are recorded, retained, and usable for investigations.

Checklist items

- ? Confirm that authentication, authorization, and privileged actions are logged for core services.
- ? Review log retention periods against operational and compliance requirements.
- ? Validate that logs are forwarded to a protected central system.
- ? Document which events are considered security-significant and must trigger alerts.
- ? Test that audit logs are immutable or access-restricted according to policy.
- ? Assign responsibility for log review, alert triage, and retention exceptions.
- ? Confirm that clock synchronization is enabled so logs can be correlated across nodes.

Evidence to capture

Capture sample audit records, log forwarding configuration, retention settings, and time synchronization status. Include a successful search for a failed login or denied access event across multiple nodes.

Acceptance criteria

Security-relevant actions are auditable end to end, logs are centrally available, and investigators can correlate events across the cluster. Retention must be long enough to support incident response and change review.

Owner and review cadence

Owner: Security operations and logging platform owner Review cadence: Continuous, with weekly review and quarterly validation

Common mistakes

- Logging locally without forwarding to a protected store
- Retaining logs for too short a period to support investigations
- Forgetting to synchronize time across nodes
- Alerting on every event without defining actionable security signals

6. Service hardening and host-level baseline

Purpose: Reduce the risk of host compromise becoming cluster compromise.

Checklist items

- ? Confirm that unnecessary services, daemons, and packages are removed or disabled on worker and master nodes.
- ? Review OS-level patching status for kernel, OpenSSH, Java runtime, and base libraries.
- ? Validate that remote root login is disabled and privilege escalation is controlled.
- ? Document host firewall settings, SSH access paths, and local administrative break-glass procedures.
- ? Test that default accounts, default passwords, and unused local users are removed.

- ? Assign baseline configuration ownership for each node class.
- ? Confirm that configuration drift detection is enabled or regularly checked.

Evidence to capture

Capture package inventory, patch status, disabled service lists, SSH configuration, local user review output, and drift reports if available. Include a sample check showing that a removed service does not restart after reboot.

Acceptance criteria

Hosts are minimally exposed, patched to the approved baseline, and controlled through repeatable configuration management. Any deviation from the baseline is documented and approved.

Owner and review cadence

Owner: Systems engineering and configuration management owner Review cadence: At build time, after patch windows, and monthly drift review

Common mistakes

- Treating cluster nodes as immutable without verifying they actually stay unchanged
- Leaving diagnostic services installed on every node
- Allowing SSH access to drift outside approved bastion workflows
- Skipping post-patch validation for services that depend on the Java runtime

7. Secure service configuration and metadata protection

Purpose: Protect catalogs, metastore data, queue definitions, and other security-sensitive configuration assets.

Checklist items

- ? Confirm that configuration files containing credentials are protected by restrictive file permissions.
- ? Review whether secrets are stored outside plain-text config where possible.
- ? Validate that metadata stores, metastores, and coordination services are access-controlled.
- ? Document backup protection for configuration, secrets, and metadata repositories.
- ? Test that a low-privilege user cannot read service configuration files with embedded secrets.
- ? Assign ownership for secret rotation and configuration change approval.
- ? Confirm that deprecated or test credentials are removed from all nodes.

Evidence to capture

Capture file permission listings, secret store references, metadata service access rules, and proof of a successful secret rotation. Include backup access controls and restoration approval records.

Acceptance criteria

Sensitive configuration is protected from unauthorized access, secrets have a controlled lifecycle, and backups are secured with the same rigor as live systems. Metadata services should not be treated as low-risk because they are administrative rather than user-facing.

Owner and review cadence

Owner: Service owner and security engineer Review cadence: At every secret rotation and after any configuration change

Common mistakes

- Embedding passwords in readable shell scripts or config templates
- Backing up secrets without protecting the backup repository
- Failing to remove test credentials after validation
- Assuming metadata systems are safe because they are not directly queried by end users

8. Job submission, workload isolation, and runtime controls

Purpose: Make sure submitted jobs cannot bypass security boundaries or consume resources in a way that masks malicious activity.

Checklist items

- ? Confirm that workload submission paths validate user identity and authorization.
- ? Review queue, pool, or capacity settings for tenant isolation.
- ? Validate that users cannot submit jobs with elevated filesystem or cluster privileges.
- ? Document limits for memory, CPU, runtime, and concurrency where they affect containment.
- ? Test that a restricted user cannot access another tenant?s data through job code or staging paths.
- ? Assign ownership for queue policy, submission gateways, and approved job profiles.
- ? Confirm that emergency overrides are logged and time-limited.

Evidence to capture

Capture queue settings, submission policy, failed authorization attempts for restricted workloads, and resource limit definitions. If the cluster supports multiple execution engines, include the controls that apply to each engine separately.

Acceptance criteria

Jobs are isolated by policy, unauthorized privilege escalation is blocked, and temporary overrides are both justified and visible. Resource controls must support security as well as availability.

Owner and review cadence

Owner: Data platform operations and workload governance lead Review cadence: At onboarding of new workload classes and after queue changes

Common mistakes

- Confusing performance tuning with security isolation
- Granting broad submission rights to avoid support tickets
- Letting emergency overrides remain active after the incident ends
- Assuming a workload manager enforces tenant separation without testing it

9. Backup, recovery, and secure deletion

Purpose: Ensure security controls survive recovery events and that retired data does not remain recoverable.

Checklist items

- ? Confirm that backups include security-critical configuration, metadata, and encryption material as required.
- ? Review who can access backup repositories and restore jobs.
- ? Validate that restore procedures preserve access controls and encryption expectations.
- ? Document retention, archival, and deletion rules for sensitive datasets.
- ? Test that deleted data is removed according to the storage layer?s capabilities and policy.
- ? Assign responsibility for backup verification and deletion approval.
- ? Confirm that restore testing includes security configuration, not only application data.

Evidence to capture

Capture backup inventory, access rules, restore test results, and deletion records. Include proof that restored systems preserve the intended permissions and encryption settings.

Acceptance criteria

You can recover securely, backups are protected from misuse, and deletion behavior matches policy and technical limits. Restore procedures must not silently weaken access controls.

Owner and review cadence

Owner: Backup administrator and security owner Review cadence: At each restore test, monthly for access review, and after retention changes

Common mistakes

- Restoring data without reapplying permissions checks
- Leaving backup repositories with broader access than production data

- Treating deletion as instantaneous when storage semantics do not support it
- Forgetting to test recovery of security metadata

10. Operational monitoring and continuous verification

Purpose: Detect drift, failed controls, and suspicious behavior before they become incidents.

Checklist items

- ? Confirm that security monitoring covers authentication failures, privilege changes, network anomalies, and service outages.
- ? Review whether configuration drift, certificate expiry, and key rotation deadlines generate alerts.
- ? Validate that security checks are included in routine cluster health reviews.
- ? Document escalation paths for critical findings and control failures.
- ? Test that alert thresholds produce actionable notifications rather than noise.
- ? Assign an owner for recurring control verification and exception tracking.
- ? Confirm that findings are tracked to closure with due dates and revalidation.

Evidence to capture

Capture alert definitions, sample notifications, exception logs, and evidence of a closed remediation item. If you have a periodic operational review, include the security checklist results as a standing agenda item.

Acceptance criteria

The cluster does not rely on a one-time hardening event. Security controls are monitored continuously, failures are visible, and every exception has an owner and due date.

Owner and review cadence

Owner: Security operations and platform reliability lead Review cadence: Continuous monitoring, weekly review, and monthly governance

Common mistakes

- Checking controls only at deployment time
- Generating alerts that no one owns
- Closing findings without re-testing the fixed control
- Ignoring certificate expiry until it causes an outage

Readiness scoring

Use a simple maturity score to decide whether the cluster is ready for production use. Score each phase as follows:

- 0 = Missing: No evidence, control not implemented, or unknown
- 1 = Partial: Control exists but is inconsistent, undocumented, or untested
- 2 = Implemented: Control is in place and evidenced, but not yet validated under failure conditions
- 3 = Verified: Control is implemented, tested, documented, and assigned for ongoing review

A practical production threshold is to require every phase to score at least 2, with identity, network exposure, encryption, and auditing scoring 3 before go-live. Any phase below 2 should block production use until remediated.

Scoring worksheet

- Identity, authentication, and administrative access: _ / 3
- Network exposure and cluster perimeter: _ / 3
- Encryption in transit and at rest: _ / 3
- Authorization, file system controls, and data segregation: _ / 3
- Auditing, logging, and traceability: _ / 3
- Service hardening and host-level baseline: _ / 3
- Secure service configuration and metadata protection: _ / 3
- Job submission, workload isolation, and runtime controls: _ / 3
- Backup, recovery, and secure deletion: _ / 3
- Operational monitoring and continuous verification: _ / 3

Interpretation:

- 0?12 total: Not ready; major control gaps remain
- 13?21 total: Limited readiness; restrict to non-production or tightly controlled pilot use
- 22?27 total: Near ready; close remaining gaps and re-test failed controls
- 28?30 total: Production-ready candidate, subject to final sign-off and exception review

Pass/fail decision rules

Treat the cluster as fail if any of the following is true:

- A management interface is exposed beyond approved administrative networks.
- A privileged service or admin account is shared, undocumented, or untraceable.
- Required encryption is disabled on a sensitive data path.
- Audit logs are missing, inaccessible, or not retained long enough for investigation.
- A low-privilege user can access protected data or metadata.
- Backups cannot be restored without weakening security controls.

Treat the cluster as pass only when all critical controls are verified, owners are assigned, exceptions are documented, and unresolved findings have explicit remediation dates.

Common mistakes during Hadoop hardening reviews

Many reviews stall because teams check for configuration presence instead of control behavior. A checkbox that says ?TLS enabled? is not enough unless you also confirm that plaintext is rejected where required. Likewise, a permissions review is incomplete if no one tests a non-privileged account against the protected path.

Another common issue is treating security as separate from platform readiness. Security defects often surface only when a cluster is scaled, patched, restored, or integrated with new jobs. If you are validating operational behavior alongside security, combine this checklist with a workload test such as Troubleshoot Apache Spark Shuffle Failures in Big Data Jobs when the cluster is expected to support Spark-based processing, because transport, permissions, and node health problems often appear together during load.

Final readiness note

A hardened Hadoop cluster is not one with the most controls; it is one with the controls you can prove, operate, and review repeatedly. Before production use, make sure every critical phase has evidence, every exception has an owner, and every control failure has a re-test plan. If you cannot demonstrate that on demand, the cluster is not yet hardened enough for production.

Use this guidance together with secure C# APIs with JWT to connect the workflow with related operational context already available on the site.