Diagnose Arma 3 connection failures and mission loading errors

An Arma 3 timeout, a signature kick and a mission that returns to the lobby are different faults. Begin by identifying the last successful stage. Preserve the community's mission, profile and persistence data; a reinstall is not a useful first response to an unknown error.

1. Reproduce the failure once

Record the UTC time, client build, selected server and exact message. Note whether the server appears in the browser, whether the player reaches the lobby, and whether the mission begins. Ask one other player to report the same stages when practical. A problem affecting one client needs different evidence from a failure affecting everyone.

Keep the existing server output and locate its current RPT log using the configured profile and logging paths. Arma startup options can relocate this information. Avoid assuming that an old RPT in another directory belongs to the latest attempt.

2. If the process never becomes ready

Compare the active `-config`, mission selection and mod paths with the last working startup configuration. A file simply named `server.cfg` is not automatically the file loaded by the service. Look for the first missing dependency, configuration error or failed content load, rather than changing unrelated network settings.

Do not repeatedly launch additional instances. A process can fail because a required port is already occupied; a second start does not create a second allocation. Send the service identifier and first error to support if the panel status and running process appear inconsistent.

3. If the server is not reachable

Compare the supplied game address and game port with the current startup settings. Arma 3's reference incoming UDP set runs from the game port through game port plus four, with distinct query, Steam and BattlEye roles. Actual allocations may use different numbers. A TCP port-checking website does not validate an Arma UDP game connection.

Report which companion allocation is missing instead of opening all ports or changing the base port blindly. Browser visibility and playable connectivity are separate checks. Retain anti-cheat and authentication settings while investigating reachability.

4. If the lobby works but the mission fails

Compare the mission's required content with the server's and client's actual mod lists. Preserve the order and identifiers in a private incident note. A successful download does not prove that the expected build was loaded. Review the RPT for the first dependency or mission-load error at the recorded time.

For signature-related kicks, verify the intended approved mod files and corresponding server key configuration. Do not disable signature verification or BattlEye to conceal the mismatch. A vanilla comparison should run on an isolated test copy with separate persistence, not by removing production mods and overwriting the active mission.

5. Verify the complete path

After correcting one identified issue, start a new connection and test entry into the intended mission with an ordinary player account. Verify persistent progress if the mission framework stores it. Reaching the server browser or an empty lobby is not the final success condition.

Escalate with the service ID, stage of failure, exact client message, build/mod comparison and redacted RPT excerpt. Remove passwords, tokens and private player information. Keep the original configuration and last working mission available for a controlled rollback rather than deleting data during diagnosis.

Related CloudExa guides

- Protect game files before changing settings
- Follow the game connection checks

Official references

- Bohemia Interactive: dedicated server networking and troubleshooting
- Bohemia Interactive: configuration and server security
- Bohemia Interactive: active startup parameters

Revision #1
Created 2026-11-10 10:57:20 UTC by CloudExa Hosting Admin
Updated 2026-11-10 10:57:20 UTC by CloudExa Hosting Admin