Introduce GrimAC alerts before automated punishments

GrimAC checks player behaviour using server-side movement simulation. Introduce it as an observed change: review alerts and legitimate gameplay before turning a violation into an automatic ban. A single flag is evidence to investigate, not a complete moderation decision.

Check the supported environment

Read the current supported-platform and installation documentation for the exact Grim release. Record Minecraft, Java, proxy, Via translation and other packet-plugin versions. Support ranges change, and an older installation article should not decide whether a newly released server version is supported.

Include Bedrock bridging and special movement plugins in your plan. A launch pad, vehicle, teleport or custom ability changes the behaviour players produce. Check the project's integration guidance instead of giving all affected players a permanent anticheat bypass.

Install and inspect the generated rules

1. Stop the server and take a completed backup of its current plugin files and configuration.
2. Download a supported official build and follow its dependency instructions. Keep only one intended Grim JAR in the loaded folder.
3. Start once, inspect Console for startup failures, and review the generated configuration and punishment rules before reopening unrestricted access.
4. Use the project's current rule syntax to keep initial monitoring free of automatic ban or kick actions. Check the actual command actions rather than assuming a configuration section named alerts is harmless.

5. Start the supervised test with an ordinary player and a separate staff observer.

Give staff specific observation permissions

The documented `/grim alerts` command toggles violation notifications for an authorised observer. `/grim profile TestPlayer` shows diagnostic information about the named player. Grant the corresponding `grim.alerts` and `grim.profile` permissions only to staff who need them. Replace the example player name with your own test account.

Commands in Console omit the leading slash. Keep administrative reload, exemption and debug-upload capabilities separate. Some diagnostics contain player information; review them before sharing an upload or screenshot publicly.

Test normal gameplay, including difficult conditions

Walk, sprint, swim, climb, use vehicles and test your server's teleports or movement abilities. Repeat representative actions using supported translated clients and any bridged clients covered by the project's guidance. Note the time, client type, ping and action when an alert occurs.

Look for repeatable patterns rather than raising thresholds until every alert disappears. Investigate lag and related plugin changes separately. A temporary, narrowly scoped workaround should have an owner and review date; broad permanent exemptions make later test results misleading.

Enable enforcement only after review

Introduce one documented punishment rule at a time, with a clear staff policy and appeal route. Keep evidence of the behaviour and the configuration that produced it. After an update, repeat the legitimate-player tests before assuming earlier thresholds remain appropriate.

See packet dependency setup and TPS and performance diagnosis. Further reading: commands, punishment rules and permissions.

Looking for another plugin? Browse the Minecraft plugin guide directory.

Revision #2
Created 2026-11-09 08:57:17 UTC by CloudExa Hosting Admin
Updated 2026-11-09 08:57:18 UTC by CloudExa Hosting Admin