

# Diagnosing Common Crash Causes

## Mod or Plugin Issues

- **Symptoms:** The crash report references a specific mod or plugin, often with an error such as `NullPointerException`, `ClassNotFoundException`, or `NoClassDefFoundError`.
- **Steps to Diagnose:**
  1. Open the crash report and look for references to a mod or plugin in the stack trace.
  2. Ensure that all mods/plugins are compatible with your Minecraft version and up-to-date.
  3. If a recently added mod or plugin caused the crash, remove it and restart the server.

## Example Error:

```
Caused by: java.lang.NullPointerException: null
    at com.example.mod.ModClass.someMethod(ModClass.java:42)
```

In this case, you may need to update or disable **ModClass**.

## 2. World Corruption

- **Symptoms:** Crashes related to world generation or chunk loading are often caused by corrupted world files.
- **Steps to Diagnose:**
  1. Open the crash report and check for keywords like “World Generation” or “Chunk Loading”.
  2. Backup the world folder and test with a clean, new world to see if the crash persists.
  3. Use tools like **WorldEdit** or **RegionFixer** to repair corrupted chunks.

## Example Error:

```
at net.minecraft.world.gen.ChunkGenerator.func_225535_a_(ChunkGenerator.java:122)
```

This points to an issue during world generation or chunk loading, likely due to a corrupted world file.

## 3. Out of Memory (OOM) Issues

- **Symptoms:** Crashes caused by memory limitations will show a `java.lang.OutOfMemoryError` in the crash report.
- **Steps to Diagnose:**
  1. Look for the **OutOfMemoryError** in the crash report.
  2. Increase the amount of RAM allocated to your server in your **startup parameters**.
    - Edit your start script to allocate more memory, for example:

```
java -Xmx4G -Xms2G -jar paper.jar
```

3. If the issue persists, reduce the number of plugins or mods, optimize server settings, or upgrade the server.

## Example Error:

```
java.lang.OutOfMemoryError: Java heap space
```

This indicates your server ran out of allocated memory and needs more resources to run properly.

## 4. Plugin or Mod Conflicts

- **Symptoms:** Crashes involving conflicting plugins or mods will show multiple references to conflicting classes or methods in the stack trace.
- **Steps to Diagnose:**
  1. Check the crash report for any repeated references to the same classes or methods across different plugins/mods.
  2. Disable or remove half of your installed plugins or mods to narrow down the source of the conflict.
  3. Update or replace conflicting plugins.

## Example Error:

```
at org.bukkit.plugin.java.JavaPluginLoader.loadPlugin(JavaPluginLoader.java:78)
```

This points to a potential conflict during plugin loading. It may be necessary to update or disable one of the conflicting plugins.

---

Revision #3

Created 3 January 2025 04:30:35 by Ridgeline Servers

Updated 4 January 2025 04:48:48 by Ridgeline Servers