Autonomous Incident Report

Generated: 2026-02-12T21:19:35.731Z
Incident ID: 29d8a5a5-e95a-49ee-a63b-5cd41dd4ce20

Root Cause Analysis

Diagnosis: The error 'SeededDemoFailure: deterministic bug for AIA demo' is intentionally thrown inside the POST /trigger handler when the payload.action is 'cause_error'. This is a deliberate error to simulate a failure scenario, but the code does not properly handle the thrown error after logging it, causing the uncaught exception and crash.
Confidence: 95%
File: apps/sample-app/src/index.ts:0-0

Generated Patch

```
--- a/apps/sample-app/src/index.ts
+++ b/apps/sample-app/src/index.ts
@@ -35,11 +35,15 @@
     if (req.method === "POST" && url.pathname === "/trigger") {
       return req
         .json()
-        .then((payload: any) => {
-          if (payload.action === "cause_crash") {
-            throw new ReferenceError("config_is_undefined is not defined");
-            return new Response("Crashed", { status: 500 });
-          }
-
-          if (payload.action === "cause_cpu") {
-            const start = Date.now();
-            while (Date.now() - start < 3000) {}
-            throw new Error("Request Timeout - CPU Limit Exceeded");
-          }
-
-          if (payload?.action === "cause_error") {
-            console.log("Triggering intentional failure scenario...");
-            const error = new Error(
-              "SeededDemoFailure: deterministic bug for AIA demo",
-            );
-            console.log(error.stack);
-            throw error;
-          }
-
-          const response: TriggerResponse = { status: "ok" };
-          return new Response(JSON.stringify(response), {
-            headers: { "Content-Type": "application/json" },
-          });
-        })
-        .catch(async (err) => {
-          console.log("Unhandled error processing request:");
-          console.log(err);
-
-          await tracer.recordError(err, {
-            method: req.method,
-            route: url.pathname,
-            statusCode: 500,
-          });
+        .then((payload: any) => {
+          if (payload.action === "cause_crash") {
+            throw new ReferenceError("config_is_undefined is not defined");
+          }
+
+          if (payload.action === "cause_cpu") {
+            const start = Date.now();
+            while (Date.now() - start < 3000) {}
+            throw new Error("Request Timeout - CPU Limit Exceeded");
+          }
+
+          if (payload?.action === "cause_error") {
+            console.log("Triggering intentional failure scenario...");
+            const error = new Error(
+              "SeededDemoFailure: deterministic bug for AIA demo",
+            );
+            console.log(error.stack);
+            throw error;
+          }
+
+          const response: TriggerResponse = { status: "ok" };
+          return new Response(JSON.stringify(response), {
+            headers: { "Content-Type": "application/json" },
+          });
+        })
+        .catch(async (err) => {
+          console.log("Unhandled error processing request:");
+          console.log(err);
+
+          await tracer.recordError(err, {
+            method: req.method,
+            route: url.pathname,
+            statusCode: 500,
+          });
+
+          // Return a proper error response to avoid uncaught exceptions
+          return new Response(JSON.stringify({ error: err.message }),
{ status: 500, headers: { "Content-Type": "application/json" } });
         });
```