Incomplete fix for GHSA-9h8m-3fm2-qjrq: BSD kenv command not using absolute path enables PATH hijacking

High **dashpole** published **GHSA-hfvc-g4fc-pqhx** 6 hours ago

Package

go.opentelemetry.io/otel/sdk (Go)

Affected versions

>= v1.15.0, <= 1.42.0

Patched versions

1.43.0

Description

Summary

The fix for GHSA-9h8m-3fm2-qjrq (CVE-2026-24051) changed the Darwin `ioreg` command to use an absolute path but left the BSD `kenv` command using a bare name, allowing the same PATH hijacking attack on BSD and Solaris platforms.

Root Cause

`sdk/resource/host_id.go` line 42:

```
if result, err := r.execCommand("kenv", "-q", "smbios.system.uuid"); err == nil {
```

Compare with the fixed Darwin path at line 58:

```
result, err := r.execCommand("/usr/sbin/ioreg", "-rd1", "-c",
"IOPlatformExpertDevice")
```

The `execCommand` helper at `sdk/resource/host_id_exec.go` uses `exec.Command(name, arg...)` which searches `$PATH` when the command name contains no path separator.

Affected platforms (per build tag in `host_id_bsd.go:4`): DragonFly BSD, FreeBSD, NetBSD, OpenBSD, Solaris.

The `kenv` path is reached when `/etc/hostid` does not exist (line 38-40), which is common on FreeBSD systems.

Attack

1. Attacker has local access to a system running a Go application that imports `go.opentelemetry.io/otel/sdk`
2. Attacker places a malicious `kenv` binary earlier in `$PATH`
3. Application initializes OpenTelemetry resource detection at startup
4. `hostIDReaderBSD.read()` calls `exec.Command("kenv", ...)` which resolves to the malicious binary
5. Arbitrary code executes in the context of the application

Same attack vector and impact as CVE-2026-24051.

Suggested Fix

Use the absolute path:

```
if result, err := r.execCommand("/bin/kenv", "-q", "smbios.system.uuid"); err == n
{
```

On FreeBSD, `kenv` is located at `/bin/kenv`.

Severity

High

CVE ID

CVE-2026-39883

Weaknesses

▶ CWE-426

Credits

 **kodareef5** Reporter

 **dmathieu** Remediation developer