CVE Processing Workflow

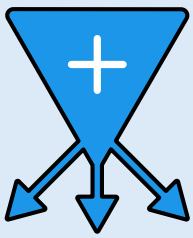
For Embedded Projects

CHEAT SHEET BY **Embedded** Expertise

PREPARE

- Remove the unnecessary features and packages from the codebase
- Enable the CVE reporting tools as part of the build process
- Run a full build to check the stripped down code base and generate a CVE report.

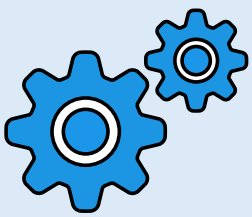
TRIAGE



For each CVE, identify the component and context

- What component is affected
- What version is the CVE known to impact
- Is the issue exploitable in the specific project context?
- How critical the issue is (score)

PROCESS



For each CVE, perform **one** of the following actions:

- Whitelist if irrelevant
- Update the package to the most suitable version (not latest)
- Search or develop a patch and apply
- Apply mitigation: firewall, SELinux, AppArmor, etc
- Leave and accept risk

If code changes, run a full regression test suite

DOCUMENT

- All CVE handling decisions must be justified and put under version control.
- See justification template on the right.

```
YAML

cve_id: CVE-2024-12245
package: OpenFeature
version: 1.1.1x
decision: Not applicable
rationale: Affected feature is
disabled in config
reviewed: albert@emb-exp.com
date: 2025-01-22
```

Read the full article on https://emb-exp.com

Need assistance triaging and fixing your CVEs? We're here to help, contact us now at contact@emb-exp.com