SECURITY ADVISORIES

Disclosed Vulnerabilities

Public vulnerability disclosures from Bombadil Systems. Each advisory includes technical details, affected systems, severity assessment, and remediation guidance.

CNA SCOPE

- Format destruction
- Detection evasion
- File structure manipulation
- Format boundary exploitation
- Semantic execution techniques
- Bombadil Systems tools

October 2026 HIGH · 8.8 CVE-2026-118938

Fickling check_safety() Bypass via Dotted __qualname__ Resolution

A dotted __qualname__ in a pickle STACK_GLOBAL (for example optparse.Values.read_file) resolves to a class-method attribute path that fickling records as an import and never analyzes as a call. check_safety() returns LIKELY_SAFE while the method executes arbitrary code during fickling.load(). Confirmed still exploitable on the vendor master branch.

CVE record →

October 2026 HIGH · 8.8 CVE-2026-16760

Fickling check_safety() Bypass via Module-Origin Trust in likely_safe_imports

Through fickling 0.1.12, likely_safe_imports grants trust to any callable imported from a non-blocked standard library module, so pickle payloads reaching code execution (dataclasses._create_fn, mailcap.findmatch) or file write (readline) pass check_safety() as LIKELY_SAFE. The flaw is in the module-origin trust model, not any specific module list.

CVE record →

July 2026 HIGH · 8.8 CVE-2026-14534

Fickling check_safety() Bypass via Unlisted Standard Library Modules

Trail of Bits fickling versions through 0.1.10 do not include _posixsubprocess, site, or atexit in the UNSAFE_IMPORTS denylist. Pickle payloads invoking dangerous functions from these modules pass check_safety() as LIKELY_SAFE, enabling arbitrary code execution through fickling.load().

CVE record →

July 2026 HIGH · 8.8 CVE-2026-14535

Fickling MLAllowlist Rendered Inoperative by Shared Mutable State

In fickling versions through 0.1.11, the UnsafeImportsML pass unconditionally registers all import nodes in the shared AnalysisContext.reported_shortened_code set. MLAllowlist then receives already_reported=True for every import, skipping its allowlist check entirely. The pass becomes dead code, and check_safety() returns LIKELY_SAFE for malicious payloads.

CVE record →



© 2026 Bombadil Systems LLC · For authorized security testing only.