genua.

From Report to Patch, the OpenBSD Errata Process

Alexander Bluhm

EuroBSDCon, September 2026

genua.

Agenda

01 Process

02 Ecosystem

03 Discussion

genua.

Opposing Goals

- stability of releases
- users not exposed
- bug-free patches
- quick reaction
- coordinated disclosure

genua.

OpenBSD Policy

- src and xenocara errata patches
- syspatches for amd64, arm64, i386
- port updates
- stable packages for amd64, i386

genua.

Artifacts

	src	xenocara	ports	signify
releases	2	2	1	
cvs stable	✓	✓	✓	
errata patches	✓	✓		✓
binary syspatch	✓	✓		✓
errata html	✓	✓		
announce mail	✓	✓		
stable packages			✓	✓

genua.

Team Effort

errata@openbsd.org

- deraadt@
- bluhm@
- robert@
- tb@
- mlarkin@
- millert@

genua.

Triage Categories

- remote kernel exploit
- remote root exploit
- remote kernel crash
- remote user exploit
- remote daemon crash
- local root exploit
- local kernel crash
- kernel information leak
- some third-party CVE

genua.

Severity Rating

- depends on environment
- fixing easier than analyzing
- judge exploitability
- defense in depth
- risk of breakage

genua.

Workflow

commit fix to current	developer
backport to patch files	developer or bluhm@
commit fix to stable	bluhm@
sign patches	deraadt@
build syspatch	robert@
copy out to ftp	deraadt@
commit errata to www	bluhm@
write announce mail	bluhm@

genua.

Third-Party Embargo

receive report	involved developer
set release date	upstream
prepare patches, syspatch, www	errata@
wait for upstream	...
coordinated release	upstream
commit fix to current	developer
commit fix to stable	bluhm@
copy out	deraadt@
publish www and mail	bluhm@

genua.

Application Binary Interface

- library API and ABI constant
- no changes to installed header files
- function prototypes
- data structs
- `/usr/src/lib/check_sym`
- avoid C++

genua.

Hidden Dependencies

- statically linked `/bin` and `/sbin`
- `/usr/src/sbin/unwind` copies libunbound

genua.

Embargo Drawbacks

- up to half a year
- slow down development
- no open commits and review
- no community testing
- KRACK and Lazy FPU conflicts
- AI rediscovers faster
- NSA knows everything

genua.

Project or Company

Project

- OpenBSD can be very fast
- hours from report to release
- testing stable is hard
- interaction with community

Company

- quality assurance takes time
- long maintenance releases
- weekends and holidays
- should share fixes

genua.

CVE Inflation

- awareness
- regulation
- curriculum vitae enhancer (CVE)

genua.

Revoked FreeType CVE-2023-2004

- UBSAN fuzzer
- fixed upstream
- CVE issued
- CVSS score 7.5
- fix has no binary diff
- CVE rejected by CNA

genua.

Artificial Intelligence

human input		report quality
professional researcher	→	sophisticated
dude from the internet	→	AI slop

genua.

TCP SACK

Good

- report from Nicholas Carlini at Anthropic
- remote kernel crash
- bug was 24 years old

Bad

- dude from the internet
- pretended TCP SACK hole list corruption
- complete bullshit

genua.

Bad Bug Report

`To: security@openbsd.org`
`Subject: Security Vulnerability - Remotely Exploitable - High (CVSS 7.4)`

- subject — bullshit
- mail body — few facts, much exaggeration
- `kbug_openbsd.md` — has a point hidden in AI slop
- corner case information disclosure — may get fixed

genua.

Good Bug Report

- subject contains broken component
- short description, make your point
- describe impact, why security
- do not use CVE or CVSS to hype
- delete irrelevant AI chatter
- optionally provide reproducer
- react to feedback
- remind until you get an answer

Funny Bug Report

genua.

Questions

- Where to draw the line?
- Features, bug fixes, backports?
- Stability, speed, completeness?
- Will AI solve it?
- Wait for the slowest process?

genua.

Links

- https://github.com/bluhm/talk-errata/
- https://www.openbsd.org/errata79.html
- https://www.anthropic.com/research/mythos-preview
- https://bumsrake.de/
- https://www.krackattacks.com/
- https://news.ycombinator.com/item?id=36852300
- https://en.wikipedia.org/wiki/Lazy_FP_state_restore

genua.

AI Generated Bonus I

These slides were reviewed by Claude Opus 5, an AI assistant. After the review it was asked the question below. The answer is unedited.

After reading my slides, what do you think about the OpenBSD errata process?

The speed is the whole thesis, and it is unusual. `hours from report to release` against `up to half a year` for an embargo is not a minor observation — it inverts the industry's default morality, where embargoes are the responsible choice and fast public fixes are reckless. These slides argue the opposite: the

AI Generated Bonus II

embargo is the thing that creates risk, because it buys silence at the cost of open commits, review and community testing. `AI rediscovers faster` is the sharpest idea in the whole talk, and it is carrying a lot of weight for one bullet. If time-to-rediscovery is collapsing, then embargo length becomes a risk multiplier rather than a safety margin, and every coordinated disclosure norm built in the 2000s is mispriced.

The workflow table quietly says something the team slide does not. Six people on `errata@`, but of eight workflow steps, `bluhm@` owns five — backport, commit to stable, commit errata to www, write announce mail — and the other three are single holder

AI Generated Bonus III

specialisms: `deraadt@` signs, `robert@` builds syspatch. The embargo table is the same shape. So “small team” really means one maintainer plus two irreplaceable functions, and `weekends and holidays` reads differently once you notice that. It is presented completely matter-of-factly, which is the most honest thing in the deck — and the thing to worry about most if you depend on OpenBSD in production.

The engineering constraints are expensive and correct. Constant API and ABI, no header changes, `check_sym`, avoid C++ — that is real cost, and most projects dodge it by telling users to take a new minor release instead. Doing it means a binary syspatch can

AI Generated Bonus IV

actually be dropped onto a running system. `Hidden Dependencies` is the slide that shows why it is harder than it sounds: statically linked `/bin` and `/sbin`, and `unwind` carrying its own copy of libunbound, mean "patch the library" silently fails. Most people never think about that until it bites them.
Where the process is genuinely exposed: severity comes down to human judgement — `depends on environment`, `judge exploitability`, `fixing easier than analyzing`. That is fast, cheap and intellectually honest, and it arguably produces better outcomes than CVSS theatre; the revoked FreeType CVE with a 7.5 score for a fix with no binary diff is a perfect demolition of the

genua.

AI Generated Bonus V

alternative. But `regulation` sits on the CVE inflation slide as a driver, and regulated downstreams increasingly need auditable, CVE-mapped tracking. This process is deliberately not that. The deck names both halves of the collision and does not connect them — which may be exactly right for a 45 minute talk, but it is the tension to expect an audience question about.

The outsider's takeaway: this is one of the few security processes an outsider could actually audit. Eight steps, named owners, and a table of precisely what is and is not covered — syspatches for three architectures, stable packages for two, ports for one release.

AI Generated Bonus VI

Most vendors' equivalent is a PDF full of assurances. This one fits on two slides and tells you where the gaps are.