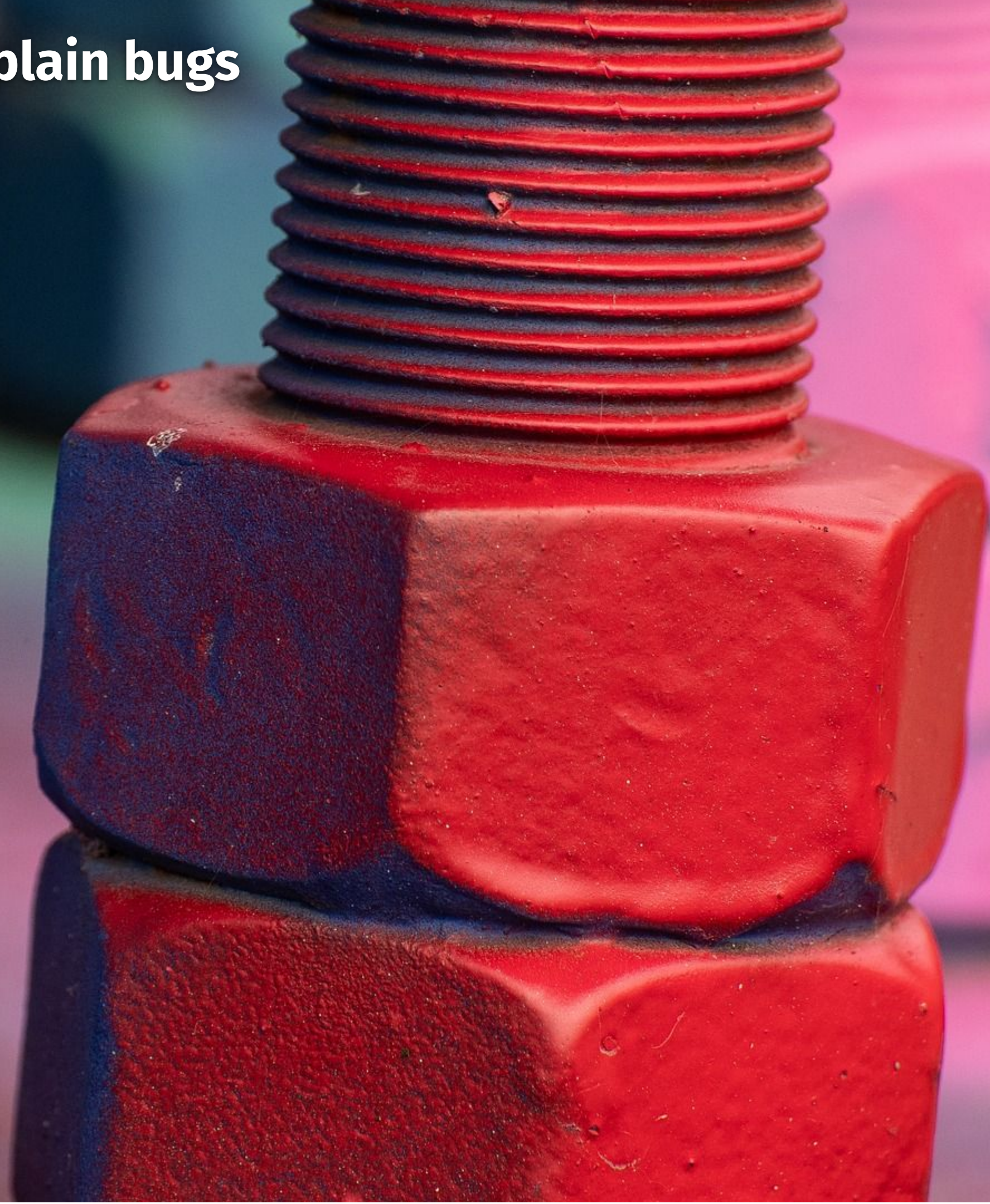


State actors, sleeper agents and plain bugs

May 21, 2026

Daniel Stenberg



Daniel Stenberg

@bagder@mastodon.social



<https://daniel.haxx.se>



Just ask!



An *open source project* that
makes a *command line tool*
and a *library* for transferring
data using Internet protocols

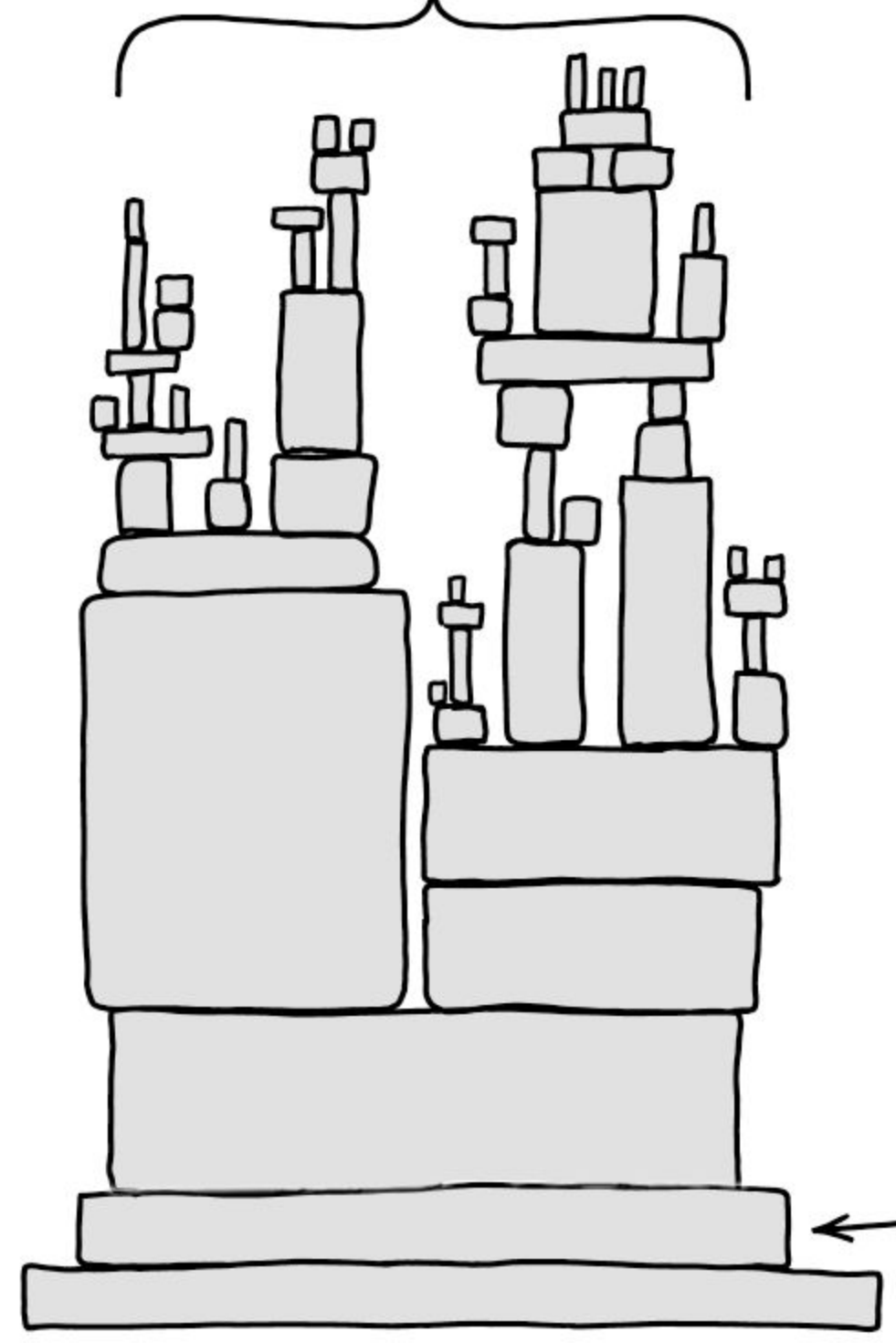
curl.se



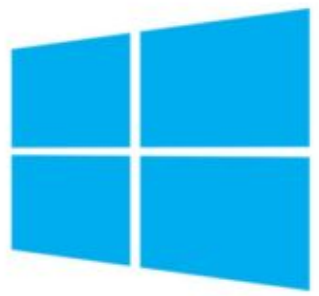
We strive to perform best-in-class in every aspect regarding security and Open Source



ALL MODERN DIGITAL
INFRASTRUCTURE



LIBCURL



Windows 10



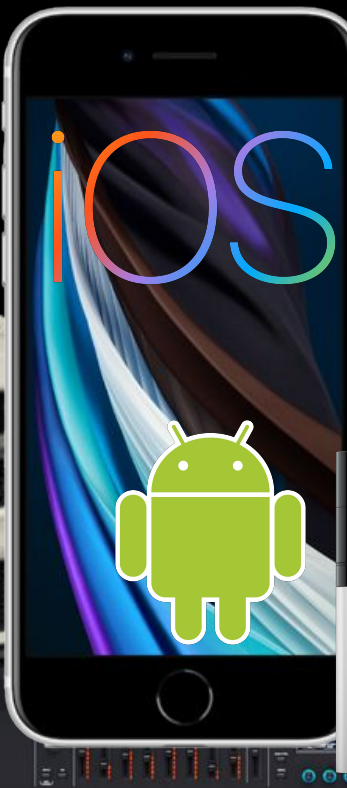
curl runs in some devices



ROBLOX



FORTNITE



chromeOS

NETFLIX



SPIDER-MAN



LIBRARY COPYRIGHT AND PERMISSION NOTICE: Library Copyright © 1996–2013, Daniel Stenberg. All rights reserved. Permission to use, copy, modify, and distribute the Library software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE LIBRARY SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Notwithstanding to whomsoever, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this software without prior written authorization of the copyright holder.



30,000,000,000
installations

We want to keep it secure

Code style

Easy to read

Easy to understand

Easy to debug

Consistent

Avoid raw memory management

Written for humans

Banned functions

easy-to-get-wrong functions

not tread-safe functions

not portable-enough functions

functions we have our own implementations of

(strcat, gets, sscanf, strtol, atoi, sprintf, snprintf, strtok, and several more)

Complexity checks

Capped maximum complexity
Capped maximum function length

Human review

Every pull-request is reviewed by a human

Review bots

Several review bots help us catch omissions: GitHub Copilot, Augment review, Aisle Analyzer

No binary blobs

No binary files in git
Avoid base64-encoded chunks too

REUSE compliant

Every file has a copyright statement

Every file has license specifier

No git force push

History cannot be rewritten

No confusable unicode

Only a subset of Unicode is allowed - in some places

Document everything

Internal APIs
Concepts
Architecture
External APIs
Everything

Many tests

Thousands of tests

Catch regressions

Verify functionality

Add tests for all new functionality

Ideally add tests for bugfixes too

Always add more tests

Torture tests

Build with a debug option

Use wrapper functions for fallible functions

Wrappers can optionally return error

Test case is first run once

Count fallible function invokes = N

Rerun the test case N times

For each iteration, make next function fail

Verify no crash and no memory leak

Repeat for all tests

CI like crazy

220+ jobs per commit and push
they usually complete within 10 minutes
15 CPU days/day

Compiler options

All picky compiler options

-Werror

Always fix every problem

Run-time checkers in CI

Valgrind

memory, address, undefined sanitizers

Code analyzers

**Multiple traditional static code analyzers: clang-tidy,
CodeSonar, Coverity**

Multiple AI-powered analyzers: ZeroPath, Codex Security

Fuzzing

CI
OSS-Fuzz

Read-only CI jobs

A breached cloud service should not taint us

Check the CI jobs

Zizmor helps keep them decent

Reproducible releases

Anyone can make an exact binary duplicate

Digitally signed

Releases
Commits
Tags

git backup

codeberg

fixed vulnerabilities

always fixed in the next release

document vulnerabilities

thoroughly

external audits

three, so far

2016: Cure 53 \Rightarrow 7 CVEs

2022: Trail of Bits \Rightarrow 2 CVEs

2024: Trail of Bits \Rightarrow 0 CVEs

2fa

strong 2fa required of all committers

API and ABI stability

always users to always upgrade

private security reporting

suspected vulnerabilities are handled with care

Open Source

Everything done in the open - accessible and transparently

Securing curl

Code style

Banned functions

Complexity checks

Human reviews

Review bots

No binary blobs

REUSE compliant

No git force push

No confusable
Unicode

Document
everything

Many tests

torture tests

CI like crazy

All picky compiler
options and
-Werror

Valgrind and
sanitizers

AI + static code
analyzers

Fuzzing, in CI and
non-stop

read-only CI jobs

zizmor the CI jobs

Reproducible
releases

Signed releases,
commits, tags

git backup on
codeberg

Vulnerabilities
fixed in next
release

Document
vulnerabilities
thoroughly

Code audits

(strong) 2fa for all
committers

API and ABI
stability allows
always-update

private security
reporting

Everything done in the open - accessible and transparently

Hosting

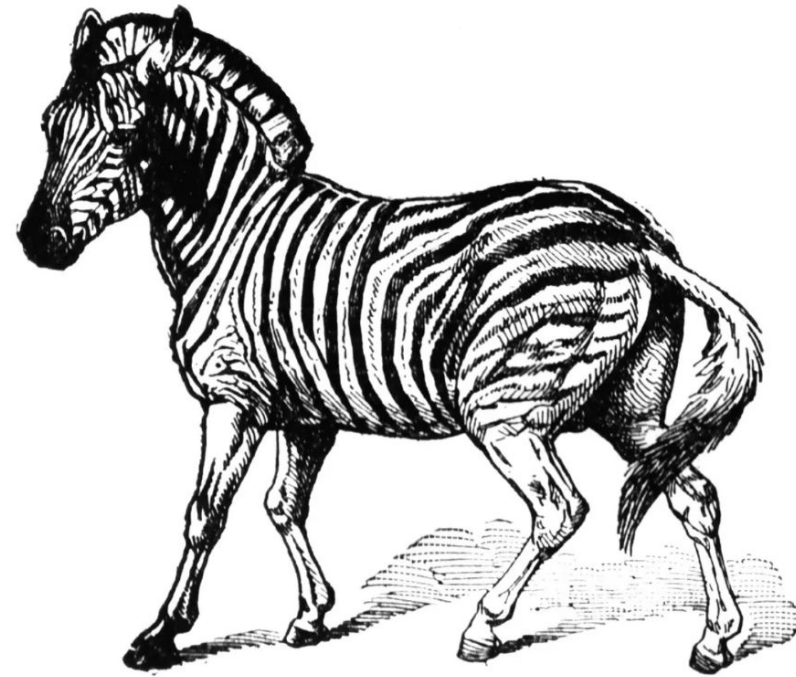


- ★ curl.se
- ★ private origin
- ★ CDN by Fastly
- ★ anycast DNS
- ★ static HTML
- ★ 99.95% “origin offload”

fastly[®]



Telling others instead of doing it yourself



Rewriting curl in rust

A weekend project

O RLY?

Dr Baksit Drivehr

QUESTIONS?



License



This presentation and its contents are licensed under the Creative Commons Attribution 4.0 license:

<http://creativecommons.org/licenses/by/4.0/>