Photo by nate on Unsplash

Four Pillars of Software Security

Really: Four vendor practices that reduce risk from vulnerabilities

This material is based upon work funded and supported by the Department of Homeland Security under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

DM21-0473

How does one measure software security?

Some incomplete answers

- This is hard to do
- Many practices, processes, technologies
- Different ecosystems, software systems, context
- Practically and economically unlikely to produce vulnerability-free software

How about “reduce risk from vulnerabilities” instead?

Observe vendor practices about

- Response capability
- Transparency

I. CVD

Coordinated Vulnerability Disclosure

Have at least basic Coordinated Vulnerability Disclosure capability

- Vulnerability Disclosure Program (VDP)
- Bug bounty
- Intake alone won't cut it
  - Look for vulnerabilities yourself first
  - Sort out internal capability first
  - Then consider external VDP or bug bounty

II. SOTA

Secure Over-the-Air Updates

Really: Secure updates

- Radio, copper, fiber, USB flash drives
- Automatic
  - Depending on ecosystem
  - Very important for consumer/commodity software
- Centralized risk
  - Note recent (and past) “supply chain” incidents
  - Security benefit probably outweighs the risk?

III. EOL

End of Life

Really: Period during which vendor will provide security updates and support

- Inform users and potential users
- Before, at time of acquisition
- With sufficient notice

IV. SBOM

More on this tomorrow!
https://www.ntia.gov/sbom

You're already doing it

- Upstream copyleft/GPL
- To build software, tools know what goes in

The ability to produce *any* sort of SBOM indicates vendor knowledge of upstream supply chain

More, older software implies more vulnerabilities

Why?

Hypotheses: If practice followed, risk is reduced

Users/customers might look for it

- Or request it
- Or require it

Yes please also do other “to the left” SDL things

- Threat modeling
- Architecture
- Language choice
- Fuzz/negative testing

Review

I. CVD
II. SOTA
III. EOL
IV. SBOM