Linaro Connect
MADRID 2024 | MAY 14-17 2024

How to stay sane : Long Term Support, Government Mandates and Open Source

Tom Gall - Director Vertical Technologies

Syllabus

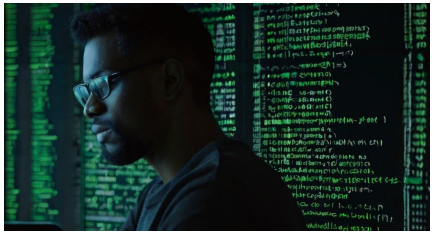
- LTS
  - Community Dynamics
  - LTS and Patch Strategies
- Government Mandates
- How to Stay Sane.
  - What we as the Open Source community can do to help
  - Useful methods for Solution Providers to stay ahead of the game

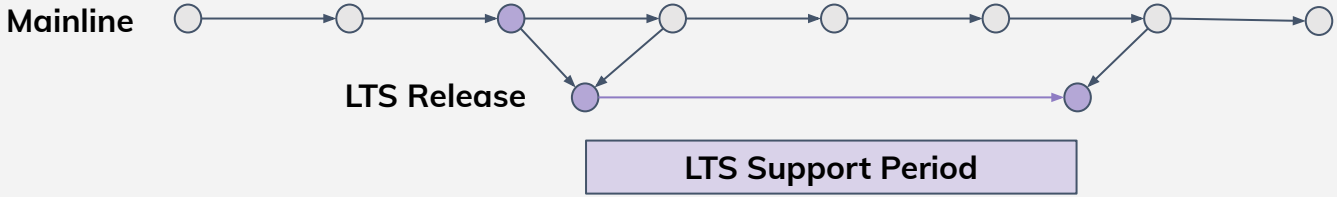
LTS

What is Long-Term Support (LTS)?

Long-Term Support (LTS) - Refers to an open source community supplying support for a released version(s) of software for an extended period of time.

- Fixes are added during the support period
- Users gain **stability** and **compatibility** over a period of time. 1 - 4 years is common
- Development continues separate from the release
- Most communities will not add new features in LTS





LTS has a resource cost, people devote time to author patches, review patches, machines for CI, triage test results, etc

LTS Policies

Essential LTS questions for users of LTS:

- How long is an LTS maintained?
- What is the policy for patches being accepted into an LTS?
- What does project hygiene look like?
  - Fixes clearly labeled? Security? Features?
  - Validation? Regression testing?

Example: U-Boot

Das U–Boot - a universal bootloader, arguably the most flexible and best open source boot loader

- No LTS
- Patches do not necessary call out they are fixes vs features
- Makes maintaining a long term version harder
- Shares a fair amount of code with Linux Kernel
  - CVEs that impact the Linux Kernel can overlap with U-Boot.
  - Likewise fixes for the kernel should in theory make it into U-Boot.

Example: The Linux Kernel

A gold standard in LTS

- LTS Started with 2.6.16 in 2006
- CVE Numbering Authority

Patch rules : See
https://www.kernel.org/doc/html/latest/process/stable-kernel-rules.html

- Must already exist in Linus' tree (upstream)
- Must be obviously correct and tested
- Must not be bigger than 100 lines
- Follow the Documentation/process/submitting-patches.rst rules
- Must fix something real - no trivial or "this could be a ..." scenarios
- Can be a new device id or quirk for hardware
- Does not add new functionality

LTS Policies

Essential LTS questions for users of LTS:

- How long is an LTS maintained?
- What is the policy for patches being accepted into an LTS?
- What does project hygiene look like?
  - Fixes clearly labeled? Security? Features?
  - Validation? Regression testing?

But wait it's **MORE THAN THAT** consider

- How long does it take for a patch to land?
- Out of order patches?
- Patch backporting - what happens when it gets hard?

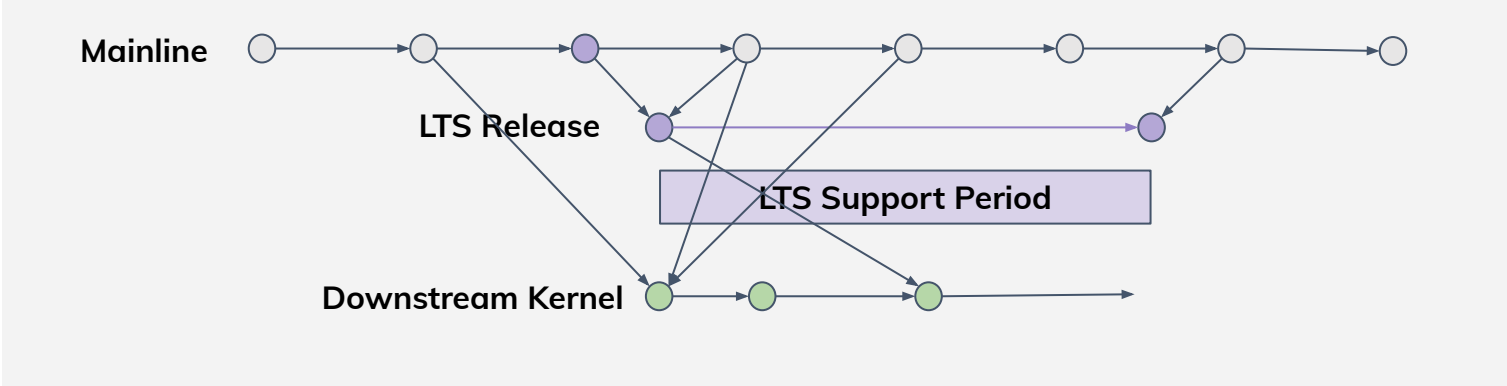
Patch strategies

Downstream Kernels might track the kernel.org Linux-stable LTS tree, taking all fixes, but not all do.

Many/Most/All downstream Kernels cherry pick fixes bringing them into their own kernels

Fixes that are integrated into mainline aren't automatically picked up.

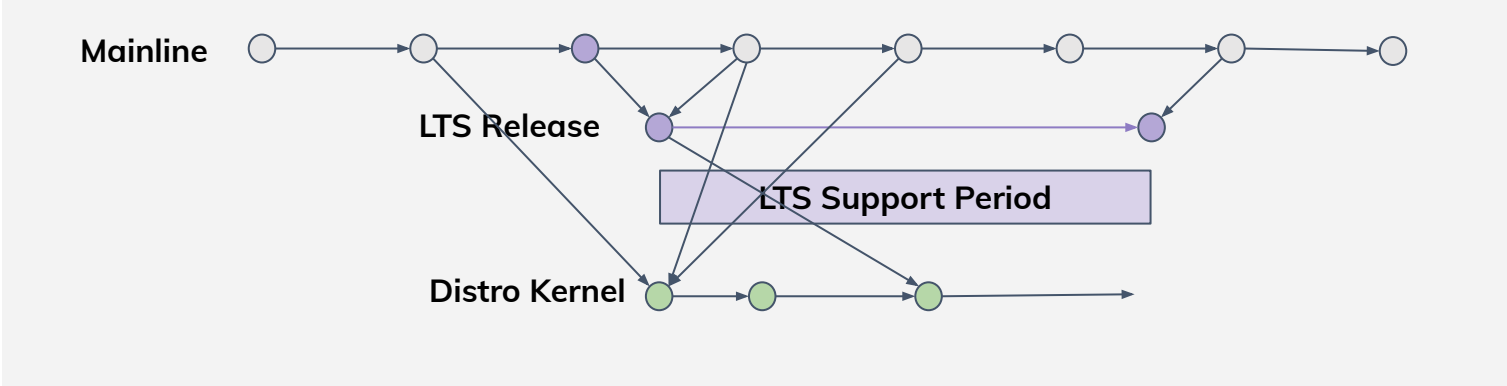
Distro/Product kernels can pick up code from vendors that isn't upstream.



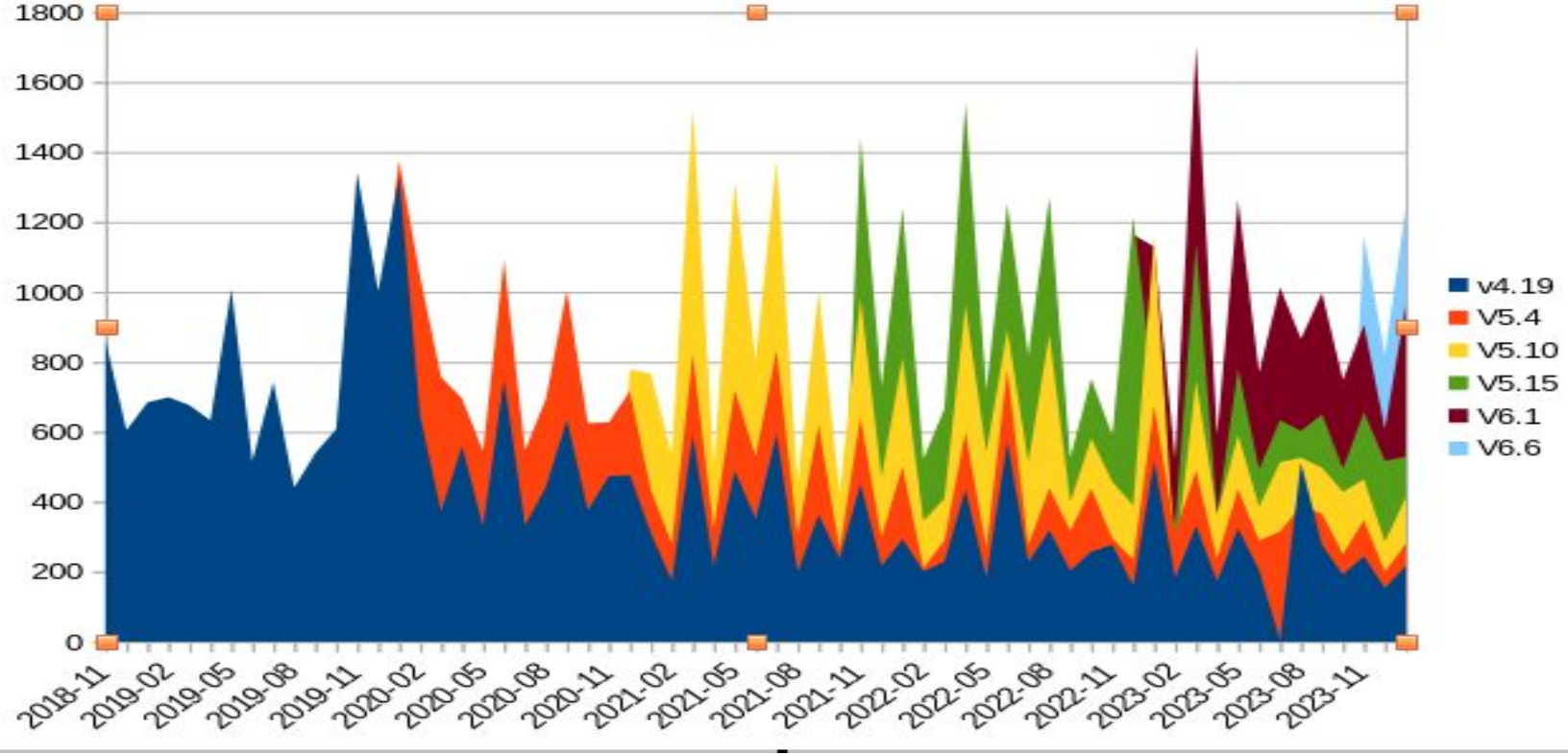
Patch strategies

Choices in patch strategies have impacts

- If all fixes must apply against mainline
  - Backports become harder the older a kernel is
  - No bug reports against old kernels, thus fixes against older kernels aren't upstreamed unless people have bandwidth to recreate and care about mainline
- On the positive side, if fixes must apply against mainline, it becomes the place to monitor for fixes bound for LTS
  - Mainline is crazy busy, CC: helps, AUTOSEL helps, data on missed fixes would be useful



Monthly LTS Kernel Fixes Over Time



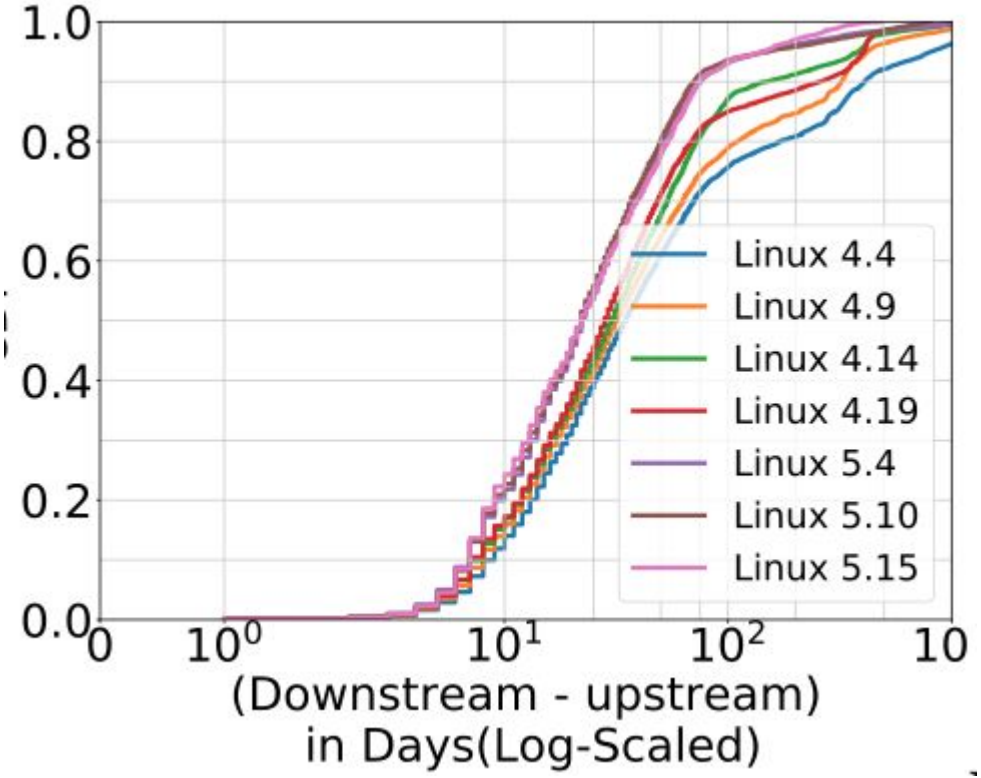
Patches

LTS Kernels, number of days for a patch to land. Mainline to LTS.

Notice for new kernels like 5.15, there is a slight uptick (fewer) days, thus patches landing quicker.

Note: 70-90% of fixes are in LTS within 100 days (10^2).

Note: Note 4.19, 5.10 and 5.15 haven't completed their 6 years of support.



Source : https://arxiv.org/pdf/2402.05212

Notable Paper

An Investigation of Patch Porting Practices of the Linux Kernel Ecosystem

- https://arxiv.org/pdf/2402.05212

Summary: Not all LTS kernels are maintained the same way, patch selection varies!

They examined 3 factors:

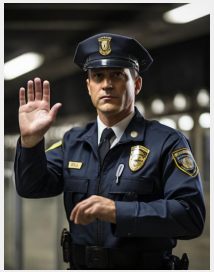
1. Patch delay: how long it takes a patch to port from upstream to downstream
2. Patch rate: the frequency of patch porting, i.e., number of ported patches per unit time
3. Bug inheritance ratio: the fraction of mainline bugs introduced in a downstream branch (in a given time period)

Observations

- Metrics and Data on LTS fixes would be useful
  - Monitor all fixes and their lifetime
  - Monitor delay for patches to travel from mainline to LTS kernels
  - Monitor patches that need porting - measure severity of rework
- Fixes against mainline aren't accompanied with new tests
  - Test using existing open test suites and "works for me" is the norm
  - https://lkft.linaro.org/tests/ is what Linaro uses
  - Would more tests based on past bugs help future regression testing?
- For LTS kernels not all patches that land in mainline make it into an LTS.
  - Please backport requests
  - CC: requests
  - AI selection via AUTOSEL <- black box
  - Greg, Shasha etc.

Government Mandates

Government Software Maintenance Obligations



Ecodesign Mandate - EU

Cyber Resiliency Act (CRA) - EU Draft

Cyber Resilience Review (CCR) - USA

US Presidential Policy Directive PPD-21 - USA

Executive Order on Cybersecurity - USA

Patch Act - USA

Securing Open Source Software Act - USA

National Institute of Standards and Technology Special Publication 800-160 Volume 2, Rev. 1

ISO 21434 - Cybersecurity Standard

There are common themes emerging from these proposed or enacted regulations.

- Encourage secure by default with limited attack surfaces
- Mandated security updates, even for devices no longer sold, must address known vulnerabilities
- Operating System, Security updates - updates available in a timely manner, device support for extended periods of time
- Process for vulnerability handling - SBOMs

Government Software Maintenance Obligations

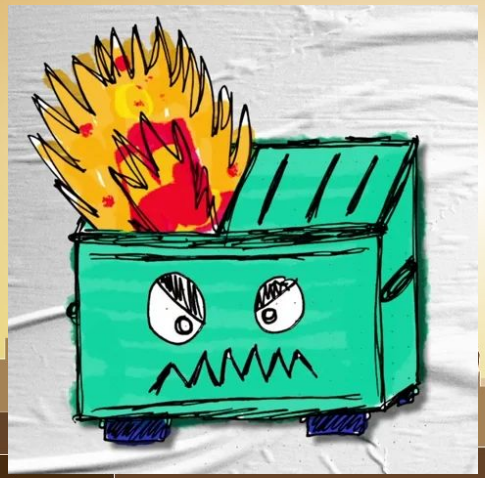
Example: Cyber Resiliency Act
(Draft EU legislation)

- Mandates security best practices in the tech industry across the supply chain
- Excuses non-profit open-source provided they are not engaged in “commercial activity”
- Onus is on manufacturers, makes them responsible for safe products through their lifespan. (5 years minimum)

Example: Ecosystem Mandate
(Impacts Handsets in the EU) pg12 Annex II

- 5 years of Operating System updates after product no longer sold into the market
- Security updates need to be available to users within 4 months of release of the OS/source code
- Functionality updates need to be available to users within 6 months of the release of the OS/source code

How to stay sane.



Stay sane out there...

- CRA, Ecodesign, ISO21434 etc
  - This is an internal to your company, for you to discussion, consult your legal team and decide on action. Don't panic.
- Software Trends
  - Software updates for the lifetime of a device addressing security and functionality is the norm
  - An update every 4 months that addresses known CVEs seems to be a good standard
- Open Source Software
  - Organized and clear communication on defects and fixes is important
  - CVEs, CNA is a process to be involved with
    - Users of your software will likely be critically depending on it (or something like it!)
  - More testing and more test suites!
- LTS
  - Software version lifetime - Uprev in field vs Extended LTS - it is a choice
    - Each approach has benefits and challenges
    - No matter the path, testing for regressions is key