

OpenWrt is reproducible!

How we got there and what we learned

Denver Gingerich

Linux Plumbers Conference 2026

Wednesday 7 October 2026

https://ossguy.com/talks/20261007_lpc/



why make OpenWrt reproducible?



why make OpenWrt reproducible?

one (of many) metric for code quality

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

demonstrates that this understanding exists

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

demonstrates that this understanding exists

security

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

demonstrates that this understanding exists

security

compliance

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

demonstrates that this understanding exists

security

compliance

for a community distro, benchmark to be proud of

why make OpenWrt reproducible?

one (of many) metric for code quality

requires a level of code understanding

demonstrates that this understanding exists

security

compliance

for a community distro, benchmark to be proud of
upstream-first approach means we solve at the root

why wasn't it reproducible?



why wasn't it reproducible?

timestamps

why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

kernel modules, esp. if used on few targets

why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

kernel modules, esp. if used on few targets

(support [hash-based module integrity checking](#)!)

why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

kernel modules, esp. if used on few targets

(support [hash-based module integrity checking!](#))

[build env - categories](#)

why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

kernel modules, esp. if used on few targets

(support **hash-based module integrity checking!**)

build env - categories

build env - examples



why wasn't it reproducible?

timestamps

vendor-/target-specific firmware image format quirks

kernel modules, esp. if used on few targets

(support **hash-based module integrity checking!**)

build env - categories

build env - examples

beyond the build env (Managing variance)

is OpenWrt reproducible?



is OpenWrt reproducible?

many years of opkg history

is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))



is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))

early work: [Reproducible OpenWrt](#)

is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))

early work: [Reproducible OpenWrt](#)

at last!

is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))

early work: [Reproducible OpenWrt](#)

at last!

but wait...

is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))

early work: [Reproducible OpenWrt](#)

at last!

but wait...

time to change everything

is OpenWrt reproducible?

many years of opkg history

>1 decade on reproducibility ([slides](#)) ([video](#))

early work: [Reproducible OpenWrt](#)

at last!

but wait...

time to change everything

for the better!

apk (Alpine Package Keeper)

apk (Alpine Package Keeper)

a very pleasant package format

apk (Alpine Package Keeper)

a very pleasant package format

from shell scripts to C

apk (Alpine Package Keeper)

a very pleasant package format

from shell scripts to C

main apk-tools fixes: [file ordering](#),
[SOURCE_DATE_EPOCH](#)

apk (Alpine Package Keeper)

a very pleasant package format

from shell scripts to C

main apk-tools fixes: [file ordering](#),
[SOURCE_DATE_EPOCH](#)

beyond Alpine: using v3 indexes already

apk (Alpine Package Keeper)

a very pleasant package format

from shell scripts to C

main apk-tools fixes: **file ordering**,
SOURCE_DATE_EPOCH

beyond Alpine: using v3 indexes already

let's not forget **the tool itself**

other challenges

other challenges

keen readers noticed inconsistent `/init` - now fixed



other challenges

keen readers noticed inconsistent `/init` - now fixed

teaching diffoscope even more formats,
for human-readable output

other challenges

keen readers noticed inconsistent `/init` - now fixed

teaching diffoscope even more formats,
for human-readable output

defaults is a hexdump diff; nice but...

other challenges

keen readers noticed inconsistent `/init` - now fixed

teaching diffoscope even more formats,
for human-readable output

defaults is a hexdump diff; nice but...

lots of vendor-specific partitioning :(

other challenges

keen readers noticed inconsistent `/init` - now fixed

teaching diffoscope even more formats,
for human-readable output

defaults is a hexdump diff; nice but...

lots of vendor-specific partitioning :(

thanks to OpenWrt, now supported by diffoscope!

current status (images graph)

current status (images graph)

<https://rebuilder-01.infra.openwrt.org/>

current status (images graph)

<https://rebuilder-01.infra.openwrt.org/>



current status (images examples)

<https://rebuilder-01.infra.openwrt.org/>

▶ malta/le	5 good	3 bad	0 fail	0 unknown	62.5%
▶ malta/le64	5 good	3 bad	0 fail	0 unknown	62.5%
▶ mediatek/filogic	703 good	0 bad	0 fail	0 unknown	100.0%
▶ mediatek/mt7622	84 good	0 bad	0 fail	0 unknown	100.0%
▶ mediatek/mt7623	12 good	0 bad	0 fail	0 unknown	100.0%
▶ mediatek/mt7629	13 good	0 bad	0 fail	0 unknown	100.0%
▶ microchipsw/lan969x	5 good	4 bad	0 fail	0 unknown	55.6%
▶ mpc85xx/p1010	14 good	0 bad	0 fail	0 unknown	100.0%

current status (packages graph)

<https://rebuilder-01.infra.openwrt.org/>



current status (packages examples)

<https://rebuilder-01.infra.openwrt.org/>

▶ malta/le	89	good	0	bad	0	fail	0	unknown		100.0%
▶ malta/le64	89	good	0	bad	0	fail	0	unknown		100.0%
▶ mediatek/filogic	95	good	0	bad	0	fail	0	unknown		100.0%
▶ <u>mediatek/mt7622</u>	93	good	0	bad	0	fail	0	unknown		100.0%
▶ mediatek/mt7623	93	good	0	bad	0	fail	0	unknown		100.0%
▶ mediatek/mt7629	92	good	0	bad	0	fail	0	unknown		100.0%
▶ microchipsw/lan969x	89	good	0	bad	0	fail	0	unknown		100.0%
▶ mpc85xx/p1010	92	good	0	bad	0	fail	0	unknown		100.0%

verifying the results



verifying the results

<https://github.com/aparcar/openwrt-rebuilder>

verifying the results

<https://github.com/aparcar/openwrt-rebuilder>

builds exactly what the buildbots build

more on why



more on why

<https://reproducible-builds.org/>



more on why

<https://reproducible-builds.org/>

which problems do reproducible builds solve?

how about your build system?

https://ossguy.com/talks/20261007_lpc/

become a Sustainer:

<https://sfconservancy.org/sustainer/>



Presentation and slides are:

Copyright © 2023-2026 Denver Gingerich, and are licensed under the [Creative Commons Attribution-Share Alike 4.0 International License](#). Slide Source available.