

#FIRSTCON24



福

36TH ANNUAL
FIRST CONFERENCE
FUKUOKA
JUNE 9-14, 2024 JAPAN

Building up a PSIRT team for
an open source project:
Lessons learned from Zephyr

Kate Stewart,
VP Dependable Embedded Systems, The Linux Foundation
kstewart@linuxfoundation.org

Who am I?

VP Dependable Embedded Systems
The Linux Foundation

- Zephyr Project: 2016 →
- ELISA Project: 2018 →

Volunteer:

- SPDX: 2009 →
- NTIA SBOM Formats & Tooling co-lead
2018 → 2021
- DHS CISA SBOM Tooling &
Implementation WG co-lead 2022 →

Contact:

kstewart@linuxfoundation.org



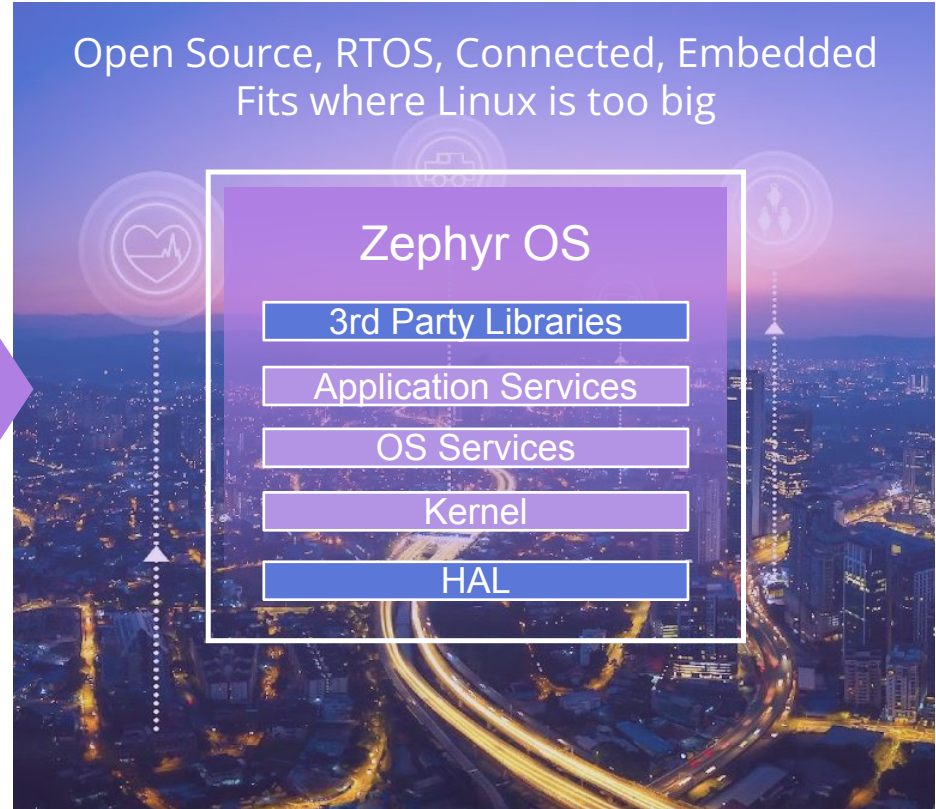
Agenda

- **Zephyr overview**
- Security best practices focus from the start
- Lessons learned after becoming a CNA
- Summary and Looking forward

Zephyr Project



- **Open source** real time operating system
- Vibrant Community participation
- Built with safety and **security in mind**
- Cross-architecture with broad SoC and development board support.
- **Vendor Neutral governance**
- Permissively licensed - Apache 2.0
- Complete, fully integrated, highly configurable, modular for flexibility
- Product development ready using **LTS includes security updates**
- Certification ready with Auditable



Products Running Zephyr Today



Proglove



Ruuvi Tag



PHYTEC Distancer



Keeb.io BDN9



Hati-ACE



Oton More



Adhoc Smart Waste



GNARBOX 2.0 SSD



Framework laptop



Safety Pod



**BLiXT solid state
circuit breaker**



Moto Watch 100



**Lildog & Lilcat pet
tracker**



Rigado IoT Gateway



Livestock Tracker



**Laird Connectivity
sensors & gateways**



**BeST pump
monitoring**



**Vestas Wind
Turbines**

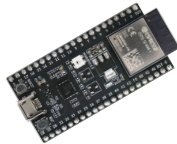


zephyrproject.org/products-running-zephyr

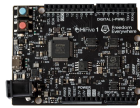
700+ supported boards... and growing



Arduino Portenta H7



ESP32



Sipeed HiFive1



nRF9160 DK



STM32F746G Disco



M5StickC PLUS



TDK RoboKit 1



BBC micro:bit v2



Blue Wireless Swan



Arduino Nano 33 BLE



Intel UP Squared



Dragino LSN50 LoRA Sensor Node



Microchip SAM E54 Xplained Pro Evaluation Kit



Raspberry Pi Pico



Altera MAX10



NXP i.MX8MP EVK



Adafruit Feather M0 LoRa



u-blox EVK-NINA-B3



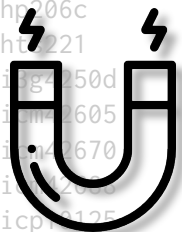
docs.zephyrproject.org/latest/boards

190+ Sensors Already Integrated



adt7420
adx1345
adx1362
adx1372
ak8975
amg88xx
ams_as5600
ams_iAQcore
apds9960
bma280
bmc150_magn
bme280
bme680
bmg160
bmi160
bmi270
bmm150
bmp388
bq274xx
ccs811

dht
dps310
ds18b20
ens
esp8266
fdd
fxos8700
fxos9500
grove
grow_r502a
hmc58831
hp206c
ht221
i2c50c
i2c605
i2c670
i2c720
icp1125
iis2dh
iis2dlpc



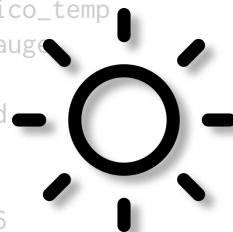
iis2iclx
iis2mdc
iis3dhhc
ina219
ina230
isl29035
ism30dtx
ite_tach_it8xxx2
ite_vcmp_it8xxx2
lis2dh
lis2ds12
lis2dw12
lis2tr
lm75
lm77
lps22
lps22hh
lps25hb
lsm303dlhc_magn



lsm6ds0
lsm6dsl
lsm6dsx
lsm9ds0
lsm9ds0_mfd
max17055
max17262
max30101
max31875
max44009
max6675
mchp_tach_xec
mcp9804
mcp9808
mcp9810
mhz19
mpr121
mpu6050
mpu9250
ms5607
ms5837



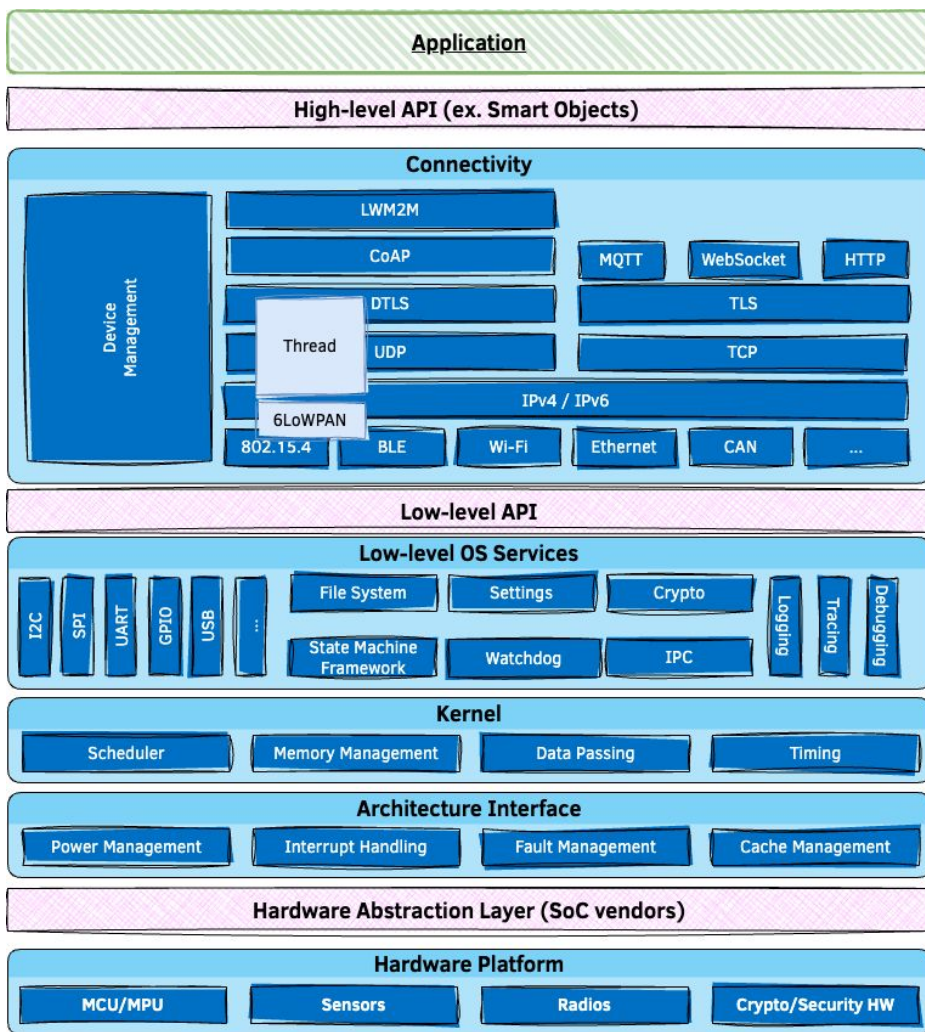
nrf5
nuvoton_adc_cmp_npcx
nuvoton_tach_npcx
nxp_kinco
opt3001
pcnt_e33
pms7003
qdec_mcp
qdec_nrfx
qdec_sam
qdec_stm32
rpi_pico_temp
sbs_gaug
sgp40
sht3xd
sht4x
shtcx
si7006
si7055
si7060



si7210
sm3511t
stm32_temp
stm32_vbat
stmesc
stts751
sx9500
th02
ti_hdc
ti_hdc20xx
tmp007
tmp108
tmp112
tmp116
vcnl4040
vl53l0x
wsen_hids
wsen_itds

 github.com/zephyrproject-rtos/zephyr/tree/main/drivers/sensor

Architecture



Supported Hardware Architectures



Cortex-M, Cortex-R
& Cortex-A



x86 & x86_64



32 & 64 bit

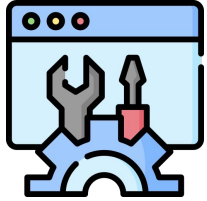


Xtensa



docs.zephyrproject.org/latest/hardware/index.html#hardware-support

Vibrant Ecosystem



Development Tools



Governing Board

Technical Steering Committee

Contributors



Applications & Middlewares



Training & Consulting

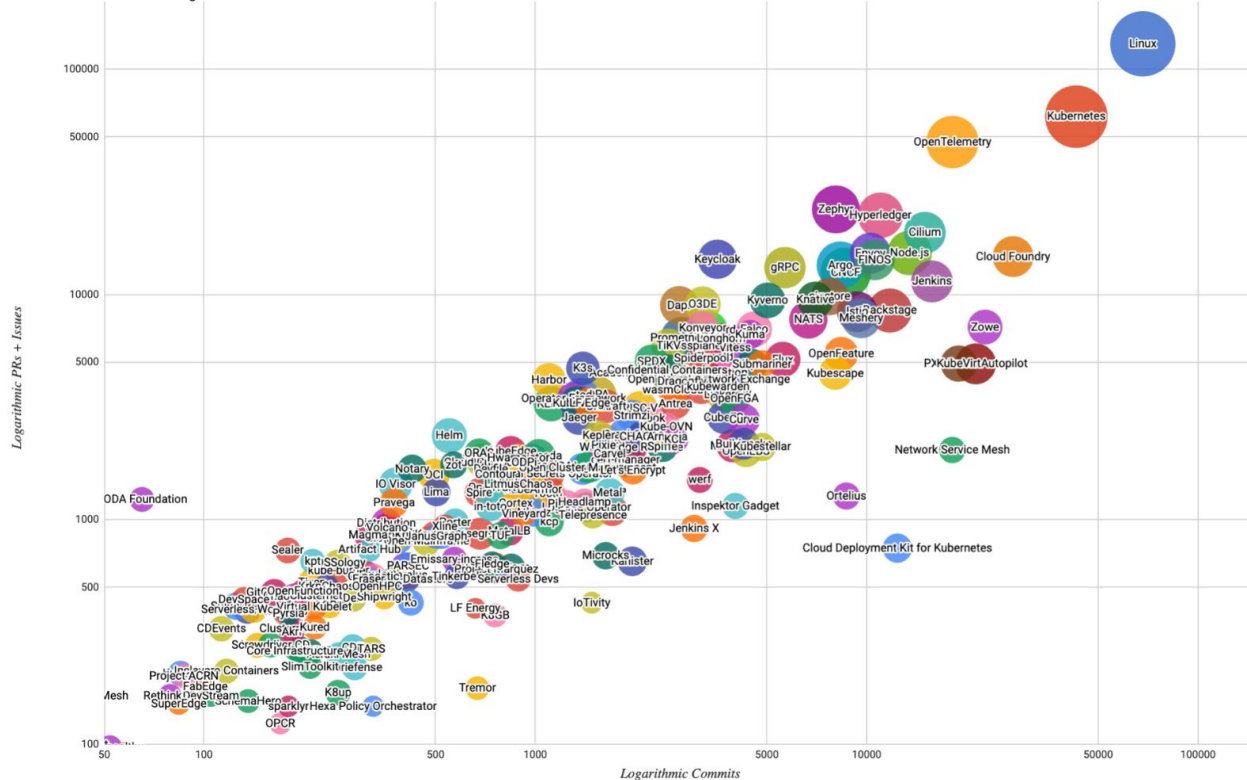


Firmwares & Libraries

Linux Foundation Project Velocity



Linux Foundation Projects 1/1/2023 - 1/1/2024



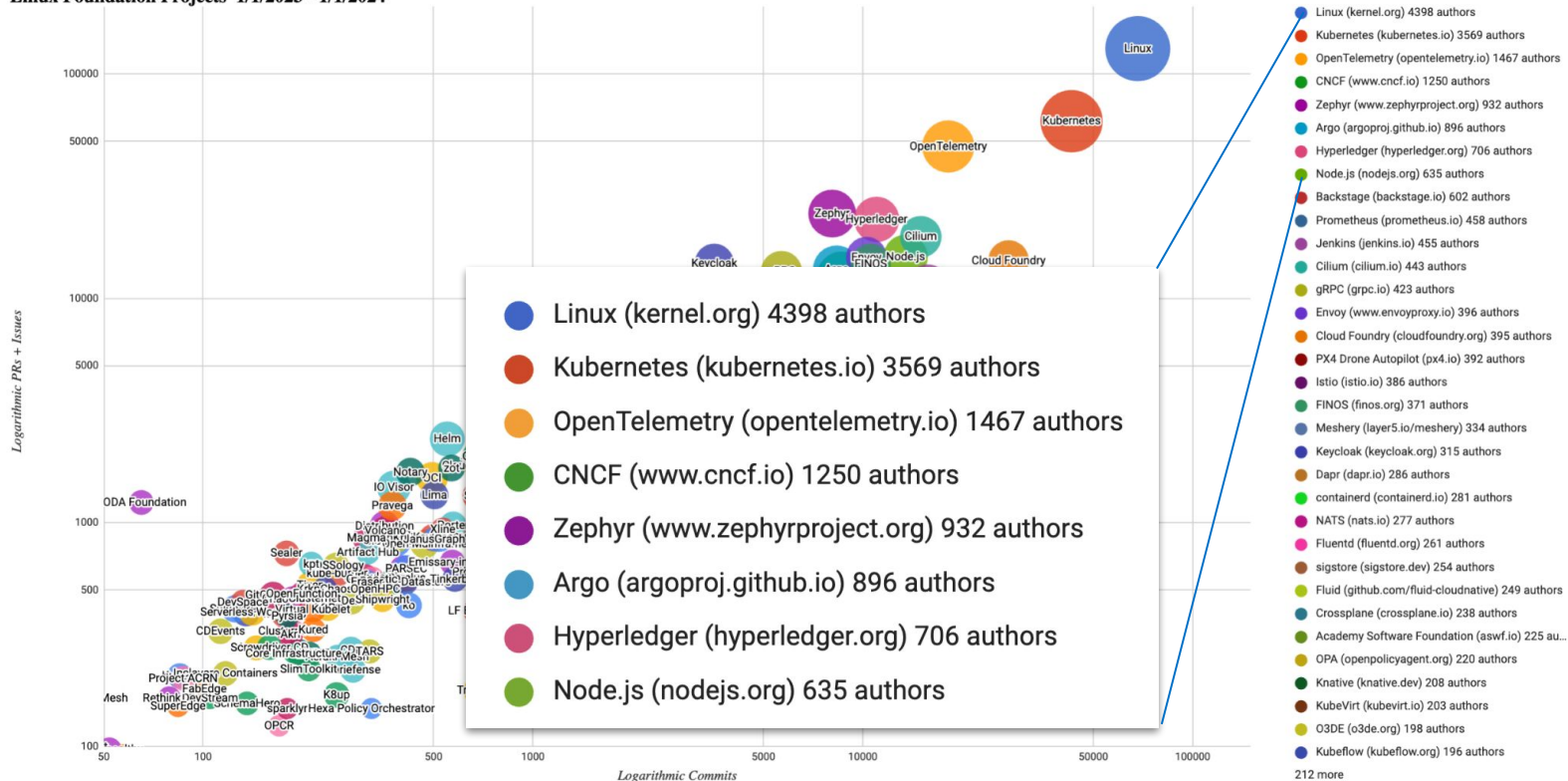
- Linux (kernel.org) 4398 authors
- Kubernetes (kubernetes.io) 3569 authors
- OpenTelemetry (opentelemetry.io) 1467 authors
- CNCF (www.cncf.io) 1250 authors
- Zephyr (www.zephyrproject.org) 932 authors
- Argo (argoproj.github.io) 896 authors
- Hyperledger (hyperledger.org) 706 authors
- Node.js (nodejs.org) 635 authors
- Backstage (backstage.io) 602 authors
- Prometheus (prometheus.io) 458 authors
- Jenkins (jenkins.io) 455 authors
- Cilium (cilium.io) 443 authors
- gRPC (grpc.io) 423 authors
- Envoy (www.envoyproxy.io) 396 authors
- Cloud Foundry (cloudfoundry.org) 395 authors
- PX4 Drone Autopilot (px4.io) 392 authors
- Istio (istio.io) 386 authors
- FINOS (finos.org) 371 authors
- Meshery (layer5.io/meshery) 334 authors
- Keycloak (keycloak.org) 315 authors
- Dapr (dapr.io) 286 authors
- containerd (containerd.io) 281 authors
- NATS (nats.io) 277 authors
- Fluentd (fluentd.org) 261 authors
- sigstore (sigstore.dev) 254 authors
- Fluid (github.com/fluid-cloudnative) 249 authors
- Crossplane (crossplane.io) 238 authors
- Academy Software Foundation (aswf.io) 225 authors
- OPA (openpolicyagent.org) 220 authors
- Knative (knative.dev) 208 authors
- KubeVirt (kubevirt.io) 203 authors
- O3DE (o3de.org) 198 authors
- Kubeflow (kubeflow.org) 196 authors
- 212 more

Source: <https://github.com/cncf/velocity>

Linux Foundation Project Velocity



Linux Foundation Projects 1/1/2023 - 1/1/2024



Source: <https://github.com/cncf/velocity>

And as of 2024-06-12?



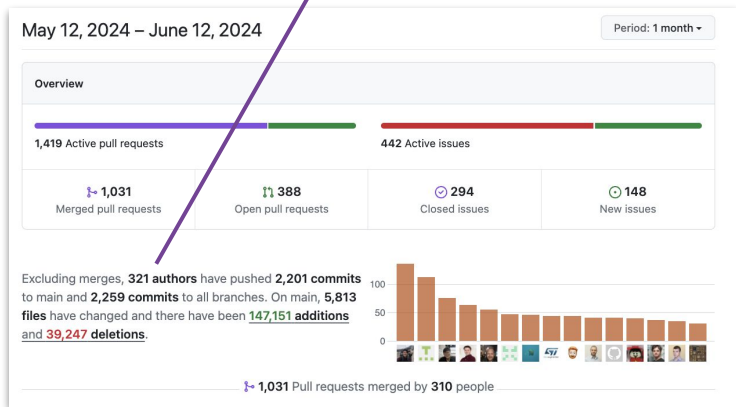
<https://github.com/zephyrproject-rtos/zephyr>

- Total commits: 97,280
- Total contributors: 2,126

<https://github.com/zephyrproject-rtos/zephyr/pulse/monthly>

- Monthly contributors: 321
- Monthly commits: 2,259

The screenshot shows the GitHub repository page for 'zephyr'. At the top, it displays '97,280 Commits' and '2,126 Contributors'. The commit history table lists recent commits with details like author, message, and time. On the right, the 'About' section describes the repository as the 'Primary Git Repository for the Zephyr Project' and lists various tags like 'iot', 'real-time', 'microcontroller', etc. The 'Releases' section shows 'Zephyr 3.6.0' as the latest release.



And as of 2024-06-12?



<https://github.com/zephyrproject-rtos/zephyr>

- Total commits: 97,280
- Total contributors: 2,126

<https://github.com/zephyrproject-rtos/zephyr/pulse/monthly>

- Monthly contributors: 321
- Monthly commits: 2,259

zephyr Public

Unwatch 398 Fork 6.1k Starred 9.9k

main 83 Branches 188 Tags

Go to file Add file Code

About

Primary Git Repository for the Zephyr Project. Zephyr is a new generation, scalable, optimized, secure RTOS for multiple hardware architectures.

docs.zephyrproject.org

iot real-time microcontroller embedded bluetooth bluetooth-le mcu rtos zephyr zephyros embedded-c zephyr-rtos

Readme Apache-2.0 license Code of conduct Security policy Activity Custom properties 9.9k stars 398 watching 6.1k forks Report repository

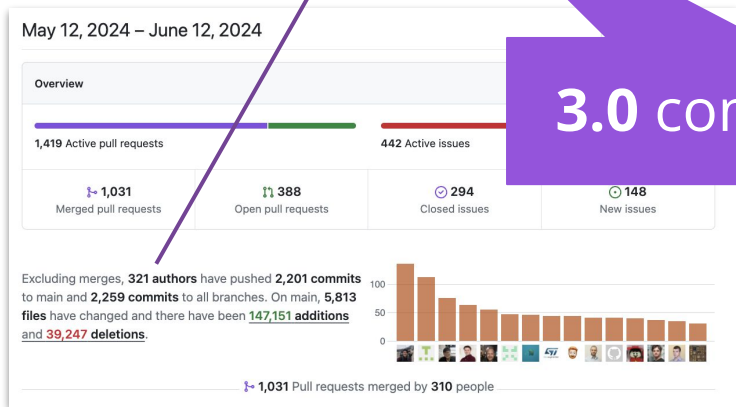
Releases 117

Zephyr 3.6.0 (Latest) on Feb 23

4145 releases

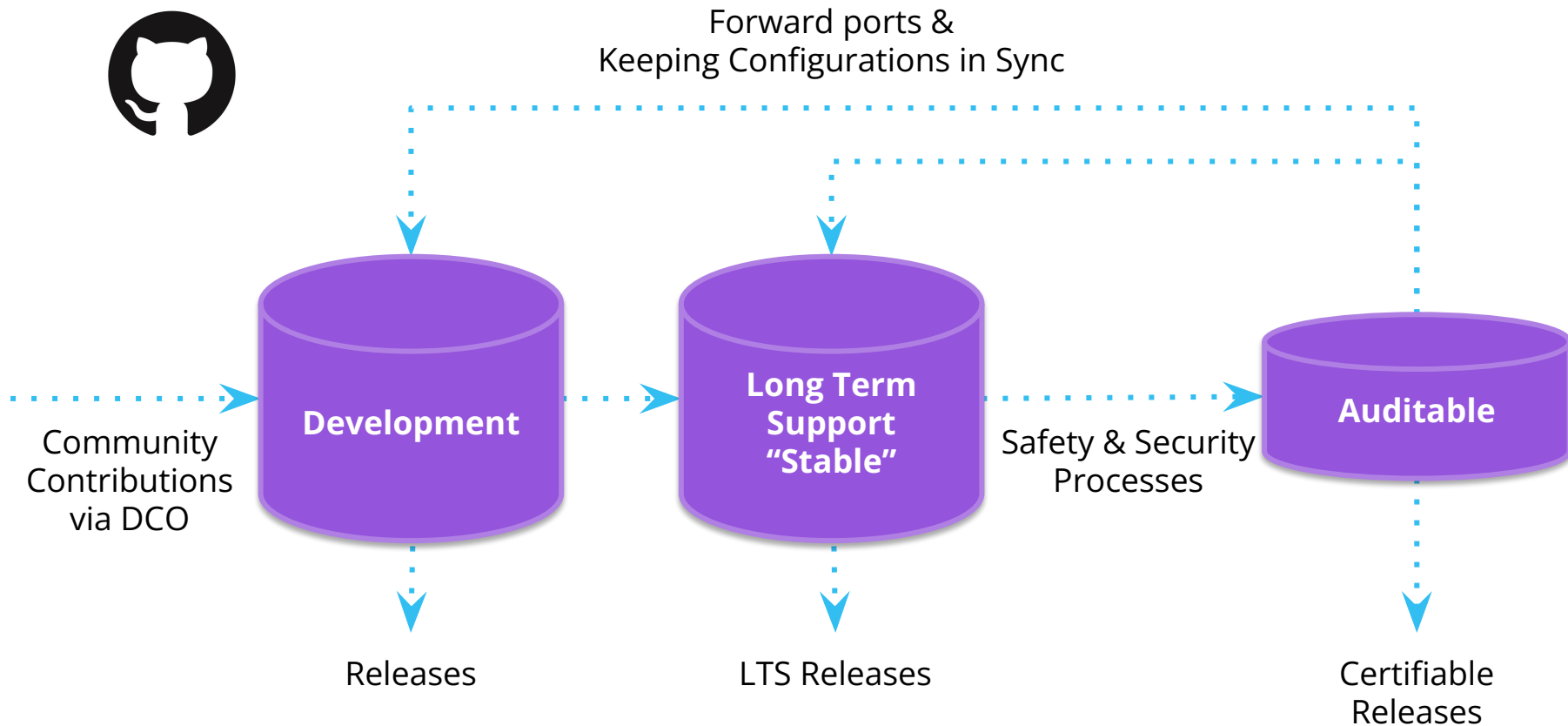
Contributors 2,126

+ 2,112 contributors



3.0 commits/hour

Code Repositories



Long Term Support (Zephyr 2.7.x)



- **Product Focused**
- Current with latest **Security Updates**
- Compatible with new hardware
 - Functional support for new hardware is regularly backported
- **Tested:** Shorten the development window and extend the Beta cycle to allow for more testing and bug fixing
- **Supported for 2+ years**

 **Doesn't include cutting-edge functionality**



github.com/zephyrproject-rtos/zephyr/releases/tag/zephyr-v2.7.0

Agenda

- Zephyr overview
- **Security best practices focus from the start**
- Lessons learned after becoming a CNA
- Summary and looking forward

Security Focus From the Start



Exhibit B

Zephyr Project Charter (the “Charter”)

The Linux Foundation
Updated August 21, 2023

1. Mission of the Zephyr Project (“Zephyr,” or, alternatively, the “Project”).

The mission of the Project is to:

- a. deliver the best-in-class RTOS for connected resource-constrained devices, built to be secure and safe.
- b. maintain an auditable code base, while taking advantage of community participation; this auditable code base is open source;
- c. include participation of leading members of this ecosystem, including micro-controller manufacturers, hardware developers, software developers and other members of the ecosystem; and
- d. host the infrastructure for the open source Project and sub-projects, establishing a neutral home for community meetings, events and collaborative discussions and providing structure around the business and technical governance of the Project.

Security Focus From the Start



Exhibit B

Zephyr Project Charter (the “Charter”)

The Linux Foundation

1. Mission of the Zephyr

The mission of the Project is

- a. deliver the best-in-class embedded Linux operating system to be secure and
- b. maintain an audit-ready code base with high developer participation; the
- c. include participation from all Zephyr controller manufacturer members of the
- d. host the infrastructure for the project as a neutral home for providing structure

6. Security Committee

- a. Composition – the Security Committee members shall consist of:
 - i. one appointed voting representative from each Platinum Member, plus
 - ii. non-voting Silver Member representatives who shall not count towards quorum.
- b. Responsibilities – the Security Committee shall be responsible for:
 - i. the definition of the processes to ensure an auditable code base, as well as any associated certification artifacts (“Security Artifacts”);
 - ii. annually elect a Representative on the Security Committee to serve as chair of the Security Committee; and
 - iii. annually elect a security architect (the “Security Architect”), who may be different from the chair of the Security Committee.

Starting Point: Adopt Known Best Practices



The screenshot shows a web browser window displaying the homepage of the CII Best Practices Badge Program. The browser's address bar shows the URL <https://bestpractices.coreinfrastructure.org/en>. The website has a dark header with the CII logo and the text "CII Best Practices". Navigation links for "Projects", "Sign Up", and "Login" are visible. The main content area features a large heading "CII Best Practices Badge Program" and a green button that says "Get Your Badge Now!". Below this, there is a paragraph of text explaining the program, followed by a section titled "Some badge earners:" which displays a grid of logos for various projects including Kubernetes, LibreOffice, curl, pkgsrc, Xen Project, and others. A circular logo for the "CORE INFRASTRUCTURE INITIATIVE BEST PRACTICES" is also present.

<https://bestpractices.coreinfrastructure.org>

Best Practices Badge



Identified best practices for OSS projects

- For *production* of OSS
- Based on practices of well-run OSS projects
- Increase likelihood of better quality & security
- Criteria designed for *any* OSS project

Web application: OSS projects self-certify

- If OSS project meets criteria, it gets a badge
- No cost
- Self-certification mitigated by automation, public display of answers (for criticism), spot-checks, and can be overridden if false

⇒ moved under Open SSF in 2021



OpenSSF Best Practices Badge Program

Get Your Badge Now!

The [Open Source Security Foundation \(OpenSSF\)](#) Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices. Projects can voluntarily self-certify, at no cost, by using this web application to explain how they follow each best practice. The OpenSSF Best Practices Badge is inspired by the many badges available to projects on GitHub. Consumers of the badge can quickly assess which FLOSS projects are following best practices and as a result are more likely to produce higher-quality secure software.

You can easily see the [criteria for the passing badge](#). More information on the OpenSSF Best Practices Badging program is [available on GitHub](#). [Project statistics](#) and [criteria statistics](#) are available. The [projects page](#) shows participating projects and supports queries (e.g., you can see [projects that have a passing badge](#)). You can also see [an example \(where we try to earn our own badge\)](#). This project was formerly known as the Core Infrastructure Initiative (CII) Best Practices badge, and was originally developed under the CII. It is now part of the [OpenSSF Best Practices Working Group \(WG\)](#). The OpenSSF is a foundation of the [Linux Foundation \(LF\)](#). The project was formally renamed from "CII Best Practices badge" on 2021-12-24.



Some badge earners:



Source: <https://www.bestpractices.dev>

Criteria



Three badge levels (passing, silver, gold)

- Any level is an achievement
- For higher levels, must meet previous level
- Based on real projects
 - Not “people should do X, but no one does that”
- Gold requires multiple developers
 - bus factor > 1*, 2-person review



More info at: <https://github.com/coreinfrastructure/best-practices-badge>

* A “bus factor” is how many people would have to be hit by a bus before a project stalls (e.g., due to lack) knowledge)

Statistics about Criteria & Levels



Criteria Statistics

Level	Total active	MUST	SHOULD	SUGGESTED	Allow N/A	Met justification required	Require URL	Met justification or URL required	Includes details	New at this level	Future
Passing	67	43	10	14	27	1	8	9	52	67	0
Silver	55	44	10	1	41	38	17	54	39	48	0
Gold	23	21	2	0	9	13	9	22	16	14	0

The "active" criteria are criteria that are included in the percentage calculations (as opposed to "future" criteria). The next columns identify the number of active criteria in each level that are MUST, SHOULD, SUGGESTED, allow a "N/A" as an answer, require justification when "met" is the answer, require a URL, require justification when "met" is the answer or a URL, include details, or are new at this level. "Future" criteria are shown on the form, and are expected to be added as active criteria to some level in the future, but are not included in completion calculations.

You can see statistics about projects over time at the [project stats page](#).

You may also see the [actual criteria](#).

- There are not a lot of gold criteria, but they are challenging.
- Source: <https://www.bestpractices.dev/en/criteria>

Zephyr's Path - Initial Passing Badge



Zephyr Launched 2016/2

- Initial security team was composed of device security experts or either open source embedded experts from our members, but limited knowledge domain overlap and understanding of issues in either space.

CII badge program launched 2016/5

- Looked through the criteria and decided to aim for passing badge.
- 75% was straight forward to fill out and was done within first week.
- Security and Analysis sections served as a focus to start organizing knowledge from diverse participants in the security team.



Zephyr achieved "Passing" badge 2016/11

- Some criteria we met fairly easily, other criteria caused significant discussion, and took a while to create the documentation (which we needed to do!)

cii best practices **passing**

Project Security Documentation



- [Project Security Overview](#)
- Started with documents from other projects
- Built around Secure Development, Secure Design, and Security Certification
- Ongoing process, rather than something to just be accomplished



[Docs / Latest](#) » [Security](#) » [Zephyr Security Overview](#) 🔍
[Open on GitHub](#) [Report an issue with this page](#)

This is the documentation for the latest (main) development branch of Zephyr. If you are looking for the documentation of previous releases, use the drop-down menu on the left and select the desired version.

Zephyr Security Overview

Introduction

This document outlines the steps of the Zephyr Security Subcommittee towards a defined security process that helps developers build more secure software while addressing security compliance requirements. It presents the key ideas of the security process and outlines which documents need to be created. After the process is implemented and all supporting documents are created, this document is a top-level overview and entry point.

Overview and Scope

We begin with an overview of the Zephyr development process, which mainly focuses on security functionality.

In subsequent sections, the individual parts of the process are treated in detail. As depicted in Figure 1, these main steps are:

1. **Secure Development:** Defines the system architecture and development process that ensures adherence to relevant coding principles and quality assurance procedures.
2. **Secure Design:** Defines security procedures and implement measures to enforce them. A security architecture of the system and relevant sub-modules is created, threats are identified, and countermeasures designed. Their

Zephyr's Path - Oops... Passing Regained



Zephyr stopped “Passing” 2017/2

- Zephyr project infrastructure underwent significant transition in 2017 (JIRA → Issues, Gerrit → github)
- Prior data was inaccurate, and we had forgotten to update it.
- Badge app notified us we were not longer “passing”

cii best practices in progress 85%

Zephyr regains passing 2017/8

- After all transitions done, updated documentation to reflect the infrastructure and we were passing again.
- **Decided to try for Silver** – but there were some big lifts for the project: key roles and responsibilities documented, longer roadmap than we’d been keeping, TLS certificate verification

cii best practices passing

Zephyr's Path - Become a CNA?



A CNA allows Zephyr Project to manage vulnerabilities, assign them CVE IDs, and handle the disclosure of information pertaining to those vulnerabilities.

- Zephyr Project CNA determines the validity of issues/vulnerabilities,
- whether or not they will be publicly disclosed,
- the amount of information that will be disclosed,
- the timing for that disclosure.

Changes made by the Zephyr Project to become a CNA:

- Zephyr Project security **documentation was be reviewed and modified** to handle the new requirements levied by the CNA process.
- **New email lists** were created to be used as points of contact for external entities (provided to MITRE to be used for contact and also will be added to Zephyr Project websites).
 - vulnerabilities@zephyrproject.org (used as primary contact for external entities)
 - zephyr-psirt-request@lists.zephyrproject.org (internal project list for CNA communications)

Zephyr's Path - Become a CNA? Yes!



Four things required* for getting a CNA in place:

1. Definition of scope:
All Zephyr project components and vulnerabilities discovered by Zephyr project participants that are not covered by another CNA.
2. Public point of contact:
vulnerabilities@zephyrproject.org was listed on websites (both Zephyr project and MITRE).
3. Direct point of contact for backdoor communications from MITRE:
zephyr-psirt-request@lists.zephyrproject.org
4. A list of email addresses that will be added to the MITRE announcement:
zephyr-psirt-request@lists.zephyrproject.org

Sent email with above in August 2017, and MITRE announced Zephyr as CNA

*per phone discussion with MITRE, July 2017

Zephyr Listed as CNA in NVD in 2017



Product, Vendor, or Product Category Name	Scope	CNA Contact Email and/or Webpage (if applicable)	CNA Type*
MITRE Corporation	All vulnerabilities not already covered by a CNA listed on this page	MITRE CVE Request web form	Primary CNA
Zephyr Project	Zephyr project components and vulnerabilities that are not covered by another CNA	vulnerabilities@zephyrproject.org	Vendors and Projects
Zero Day Initiative	Products and projects covered by its bug bounty programs not already covered by another CNA	zdi-disclosures@trendmicro.com ZDI contact page	Bug Bounty Programs
ZTE Corporation	ZTE products only	psirt@zte.com.cn	Vendors and Projects

* Key for CNA Types:

Bug Bounty Programs - assigns CVE IDs to products and projects that utilize the Bug Bounty service's product offerings.

National and Industry CERTs - performs incident response and vulnerability disclosure services for nations or industries. They may assign CVE IDs as part of their role and scope.

Primary CNA - oversees the CNA program.

Root CNA - manages a group of sub-CNAs within a given domain or community.

Vendors and Projects - assigns CVE IDs for vulnerabilities found in their own products and projects.

Vulnerability Researchers - assigns CVE IDs to products and projects upon which they perform vulnerability analysis.

* https://cve.mitre.org/cve/request_id.html#cna_participants

Zephyr CNA Entry Today



CVE [About](#) [Partner Information](#) [Program Organization](#) [Downloads](#) [Resources & Support](#) [Report](#)

Zephyr Project

Links that redirect to external websites [↗](#) will open a new window or tab depending on the web browser used.

Steps to Report a Vulnerability or Request a CVE ID

Step 1: Read disclosure policy View Policy	Step 2: Contact Email
---	--

Scope	Zephyr project components, and vulnerabilities that are not in another CNA's scope
Program Role	CNA
Top-Level Root	MITRE Corporation
Security Advisories	View Advisories
Organization Type	Vendor Open Source
Country*	USA

* Self-identified by CNA

Source: <https://www.cve.org/PartnerInformation/ListofPartners/partner/zephyr>

Zephyr PSIRT Today



Project Security Incident Response Team

- Led by Zephyr Security Architect (elected annually from peers)
- Volunteers from Security Committee (Zephyr Project Members) do initial triage
- Manage embargo windows and interaction with maintainers for fixes into upstream and then backports to LTS
- Responsible for satisfying evolving CVE Program & CNA Process Requirements.

Agenda

- Zephyr overview
- Security best practices focus from the start
- **Lessons learned after becoming a CNA**
- Summary and looking forward

Zephyr's Badge Path Continues...



Zephyr almost at "Silver" 2018/4

- Zephyr addressed all issues except "TLS certificate verification", we had a TLS library, but Zephyr is an OS, not an App.
- Threat model and justification documents that security requirements are met had to be created, again issue not an App.

cii best practices **passing**

Zephyr gets Silver 2018/9

- After implementing a separate application as a sample for TLS issue

cii best practices **silver**

Zephyr's Gold Badge - Feb 2019!



Zephyr Project

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

If this is your project, please show your badge status on your project page! The badge status looks like this: `openssf best practices gold` Here is how to embed it: [Show details](#)

These are the `passing` level criteria. You can also view the `silver` or `gold` level criteria.

[Expand panels](#)

[Show all details](#)

[Show only incomplete criteria](#)

Basics	13/13
Change Control	9/9
Reporting	8/8
Quality	13/13
Security	16/16
Analysis	8/8

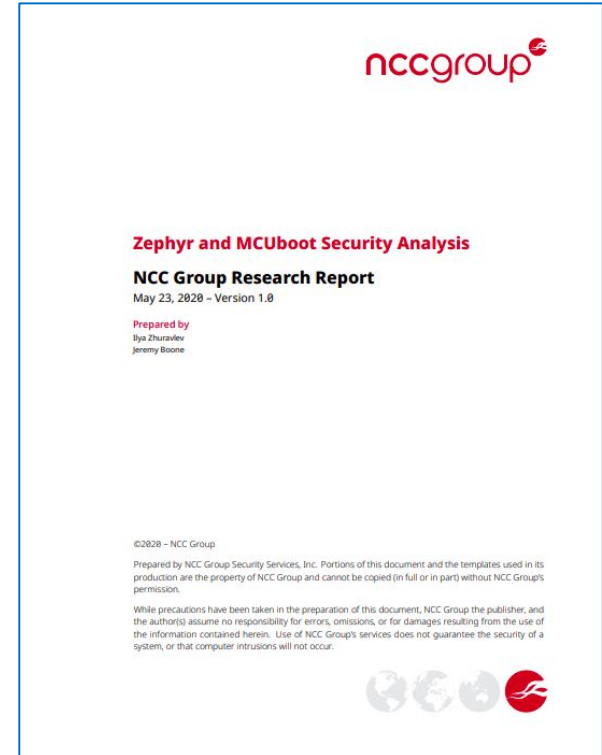
This data is available under the [Creative Commons Attribution version 3.0 or later license \(CC-BY-3.0+\)](#). All are free to share and adapt the data, but must give appropriate credit.

Source: <https://www.bestpractices.dev/en/projects/74>

First Bulk Security Report (2019)



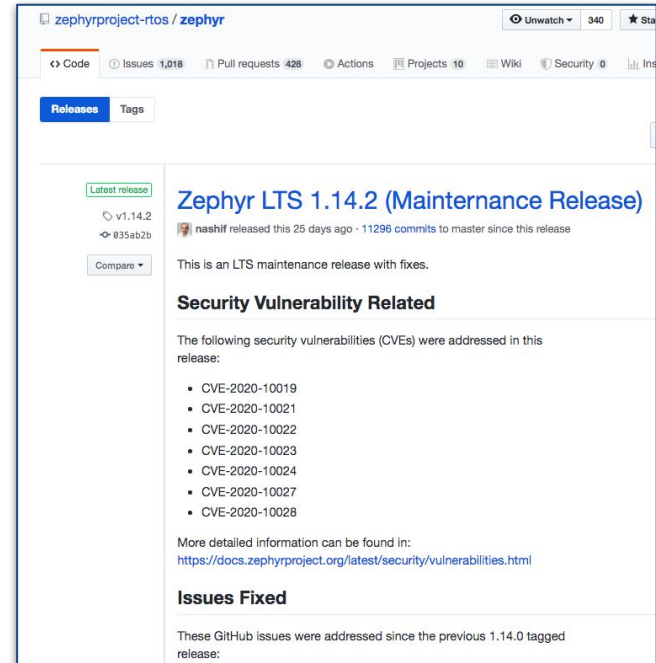
- [NCC Group reported](#) ~26 issues
- Critical, High and Medium made into JIRA tickets (we used JIRA before transitioning private github we use today)
- All were addressed
- After embargo, everything updated in the [vulnerability report](#) page
- Most resulted in 1 or more CVEs being reported



Results from the NCC Report



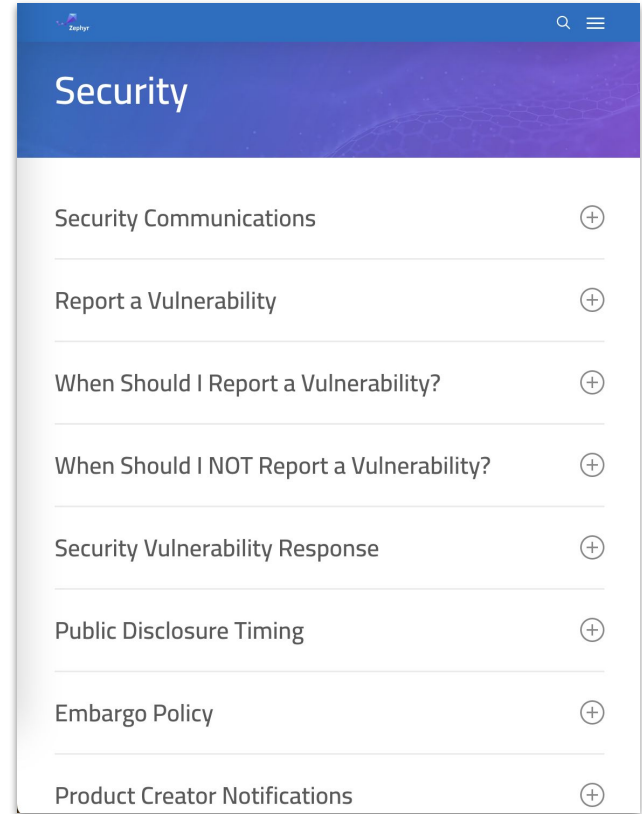
- Most issues were fixed in reasonable time and included in releases
- One issue, recommendation is to disable
- Increased embargo from 60 to 90 days
 - Zephyr isn't an end product, vendors need time to incorporate fixes into products
 - Zephyr needs alert system to notify vendors
- Continue to improve processes



Improving Processes...



- Highlighted need to better document process
- Added [vulnerability reporting](#) to project docs
- Added [security section](#) to main project page
- Process:
 - Embargo period extended
 - Stages issue goes through
 - Working with maintainers to see issues fixed
 - Public disclosure at end



Better Support for Product Makers



- For an embargo to work, product makers need to be notified early so they can remediate.
- Created [Vulnerability Registry](#) for vendors to register to receive these alerts for **free**
- **Goal:** Zephyr to fix issues within 30 days to give vendors 60 days before publication of vulnerability

Product Creators Vulnerability Alert Registry

If you believe your organization meets the criteria to be eligible to receive vulnerability alerts please fill out the form below.

Criteria for Participation

- Have a contact who will respond to emails within a week and understands how Zephyr is being used in the product.
- Have a publicly listed product based on some release of Zephyr.
- Have an actively monitored security email alias.
- Accept the Zephyr Embargo Policy that is outlined below.

Removal: If a member stops adhering to these criteria after joining the list then the member will be unsubscribed.

More information on Zephyr's Security and Disclosure practices can be found at [Security](#).

Source: <https://www.zephyrproject.org/vulnerability-registry/>

What we had to do before VEX...



Advisory Issued by project on 20201208:

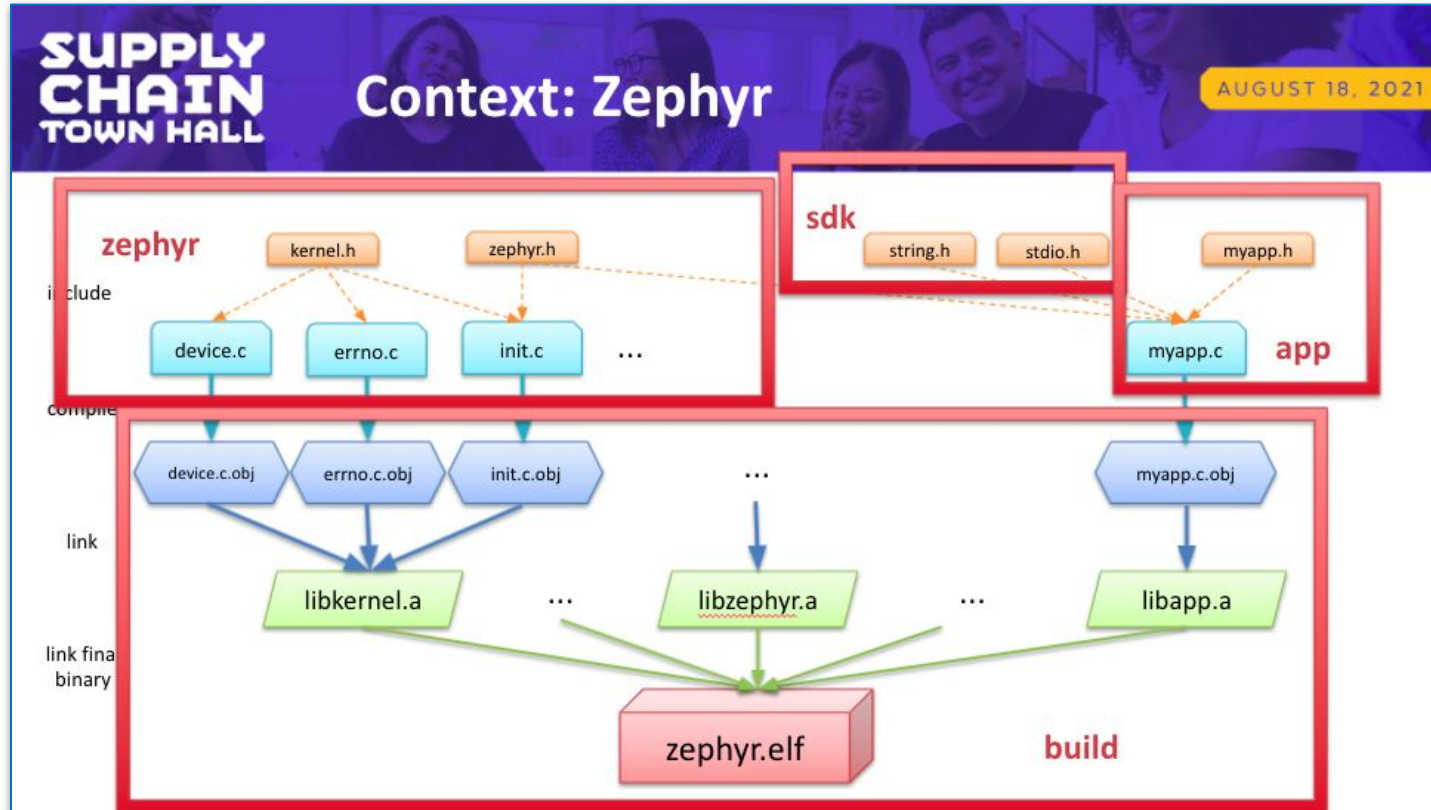
Zephyr current release (2.4) does **not use** Fnet or other stacks.

The Zephyr LTS release 1.14 contains an implementation of the TCP stack from Fnet.

- Of the vulnerabilities reported in Fnet, 2, [CVE-2020-17468](#), and [CVE-2020-17469](#), are in the IPv6 Fnet code, one, [CVE-2020-17467](#), affects Link-local Multicast Name Resolution (LLMNR), and 2, [CVE-2020-24383](#), and [CVE-2020-17470](#) affect DNS functionality.
- **None of the affected code has been used** in the Zephyr project, while 1.14 does use the Fnet TCP, it does not use the affected IPv6, DNS or LLMNR code.

<https://www.zephyrproject.org/zephyr-security-update-on-amnesia33/>

SBOM generation added in 2021



Learn more at: <https://www.youtube.com/watch?v=KYC3YpSu9zs>

Automating SBOM Generation During Build!



1. Create a build directory with CMake file API enabled
2. Build project with “build metadata” enabled
3. Compute SBOM(s)

```
west spdx --init -d BUILD_DIR
west build -d BUILD_DIR -- -DCONFIG_BUILD_OUTPUT_META=y
west spdx -d BUILD_DIR
```



- | | |
|--------------------|---|
| zephyr.spdx | SBOM for the Zephyr source files actually used by your application |
| app.spdx | SBOM for the source files of your application |
| build.spdx | SBOM for all the build objects , inc. of course your final image |

SBOM's at Scale...Automatically



708 boards

13 apps

**All BUILT,
PASSED,
GENERATED**
have **3 SBOMs**
available to
download &
inspect

A screenshot of the Zephyr Dashboard web interface. The browser address bar shows "zephyr-dashboard.renode.io". The page features a search bar, a navigation menu on the left, and a main content area displaying a list of boards. The boards are grouped by architecture: ARC (20), ARM32 (529), ARM64 (26), MIPS (2), NIOS2 (2), and RISCv32 (35). Each board entry shows its name and a row of five status buttons: PASSED (green), GENERATED (orange), PASSED (green), Download SBOM (blue button with a download icon), and PASSED (green). The boards listed include Andes ADP-XC7K AE350, ESP32-C3, ESP32C3 LuatOS Core, ESP32C3 LuatOS Core USB, GigaDevice GD32VF103C-STARTER, GigaDevice GD32VF103V-EVAL, and ICE-V Wireless. The left sidebar contains the "RENODE ZEPHYR DASHBOARD" logo, an "ARCHITECTURE" menu, "BUILD DETAILS" section with "SHOW SIMULATION" and "E017006BE4" and "9D46C2F8BE" options, and a "DO YOU WANT YOUR BOARD SUPPORTED IN RENODE?" section with a "CONTACT US" link.

Source: <https://zephyr-dashboard.renode.io/>

Dashboard SBOM

blinky-build.spdx

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: build
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/build
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z
```

```
ExternalDocumentRef: DocumentRef-app http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/app SHA1: 594de9d45188c55bdb059a2b0045987bb87e79be
ExternalDocumentRef: DocumentRef-zephyr http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/zephyr SHA1: 4ae97af97a0e9fbc0507f72ea71ad3bf2f9caffa7
```

```
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-zephyr-final
```

...

```
FileName: ./zephyr/arch/arm/core/cortex_m/libarch_arm_core_cortex_m.a
SPDXID: SPDXRef-File-libarch-arm-core-cortex-m.a
FileChecksum: SHA1: 310c7abd765821c8e8df8ceb1ac8bae330f371b1
FileChecksum: SHA256: 5efe0a524dd3a48e7cf6d637966a46fffa60119f4ab2b2b2f3ec4d924f5ea2a
LicenseConcluded: NOASSERTION
LicenseInfoInFile: NONE
FileCopyrightText: NOASSERTION
```

```
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-exc-exit.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fault.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fault-s.s
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-fbw.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-reset.s
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-scb.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-thread-abort.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-vector-table.s
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-swap.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-swap-helper.s
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-irq-manage.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-irq-init.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-prepare.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-thread.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-cpu-idle.c
Relationship: SPDXRef-File-libarch-arm-core-cortex-m.a GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-irq-init.c
```

...

```
FileName: ./zephyr/zephyr.elf
SPDXID: SPDXRef-File-zephyr.elf
FileChecksum: SHA1: 2e80741d3c373bd7626bc49625783ea8f1bcbcb
FileChecksum: SHA256: 7a838128652e85835f9167be429d41559701533fbd0d09b6bab9176a289f5dc5e
LicenseConcluded: NOASSERTION
LicenseInfoInFile: NONE
FileCopyrightText: NOASSERTION
```

```
Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM DocumentRef-zephyr:SPDXRef-File-empty-file.c
Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM SPDXRef-File-isc-tables.c
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libapp.a
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libzephyr.a
```

...

blinky-app.spdx

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: app-sources
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/app
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z
```

```
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-app-sources
```

```
#### Package: app-sources
```

```
PackageName: app-sources
SPDXID: SPDXRef-app-sources
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: Apache-2.0
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PrimaryPackagePurpose: SOURCE
PackageLicenseInfoFromFiles: Apache-2.0
FilesAnalyzed: true
PackageVerificationCode: a5993032fe245294fb73f4ed2f53be3356662f6
```

```
FileName: ./src/main.c
SPDXID: SPDXRef-File-main.c
FileChecksum: SHA1: d71a9d7b80f5eac4b749b84c57297614ef8e3899
FileChecksum: SHA256: cdc42b14891c38dfc131eb3dea80986698289496a18c7e76e9945f2e3dd17152
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: NOASSERTION
```

blinky-zephyr.spdx

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: zephyr-sources
DocumentNamespace: http://spdx.org/spdxdocs/zephyr-ab992a5d-47b4-44ee-8357-1b68719b389b/zephyr
Creator: Tool: Zephyr SPDX builder
Created: 2024-06-07T01:12:51Z
```

```
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-zephyr-sources
```

```
#### Package: zephyr-sources
```

```
PackageName: zephyr-sources
SPDXID: SPDXRef-zephyr-sources
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: Apache-2.0
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PackageLicenseInfoFromFiles: Apache-2.0
FilesAnalyzed: true
PackageVerificationCode: f10da9dec03dd29bb556c72963bf33ae9f840643
```

```
FileName: ./zephyr/arch/arm/core/cortex_m/_aeabi_read_tp.S
SPDXID: SPDXRef-File-aeabi-read-tp.S
FileChecksum: SHA1: 62d0921844d538be8c28ae5bc4c0b9f7692bd3
FileChecksum: SHA256: 1ba5712dbc2a5d48a57fde5070b2cdc0f6b2bb86a740ae2f55811aaf1bea0a1
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: NOASSERTION
```



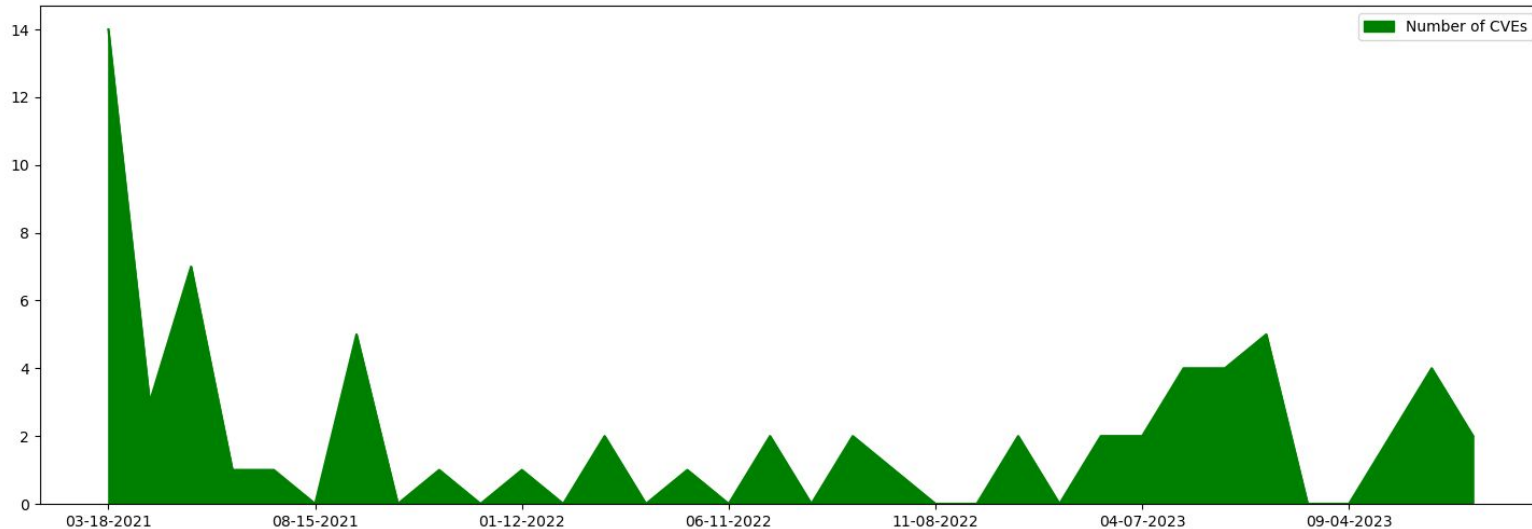
Vulnerability Infrastructure → Github 2021



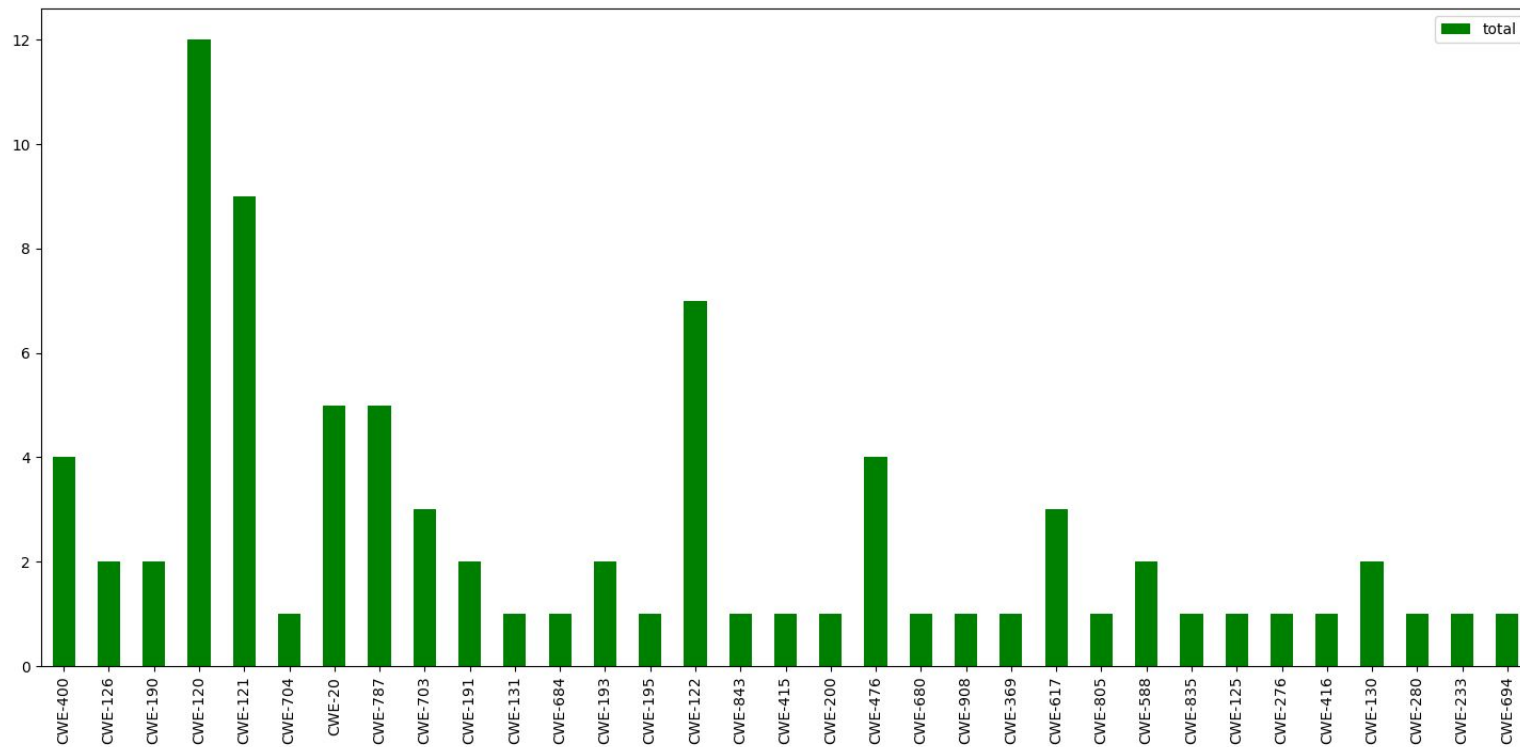
Why Transition?

Private repos became available. Better integration with rest of code.
No additional ids to manage. Improved analysis capabilities

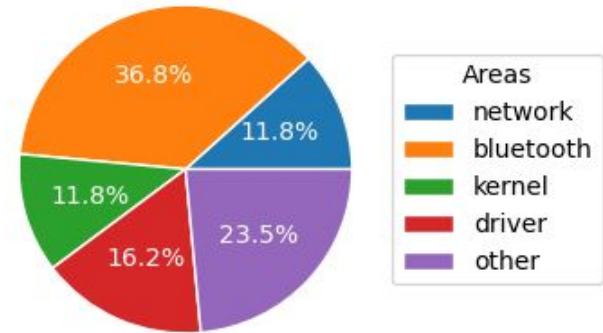
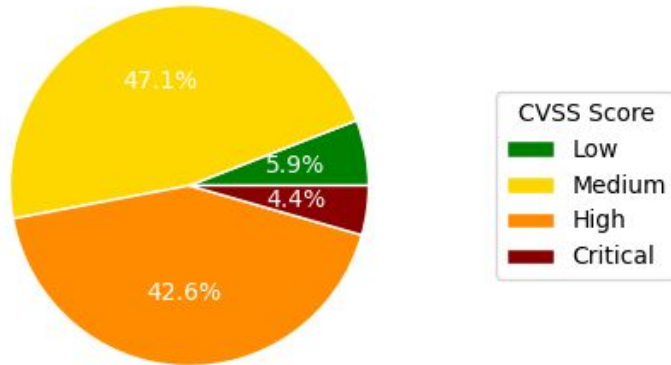
Total of CVEs published : 68 (since we started using github)



CWE Breakdown



Scoring & Code Area Breakdown



Security Working Group added March 2022



Security Committee

- **Restricted** to one representative from each platinum member, an architect (Flavio Ceolin), and a chair (David Brown)
- Meeting: Every 2 weeks
- Topics:
 - Vulnerabilities
 - PSIRT processes
 - Financial/contracts
 - Other sensitive information

Security Working Group

- **Open** to any participant
- Meeting: Every 2 weeks
- Topics:
 - Security Standards
 - ETSI EN 303-645
 - FIPS 140-3
 - SP 800-128
 - Annex K (C11 standard)
 - Evolving Security Processes
 - Code Analysis Tools
 - Documentation


Work on ETSI EN 303-645 in 2023




A screenshot of the Zephyr project's documentation website. The left sidebar is dark blue and contains the Zephyr logo, the version number "3.6.99", a search bar, and a navigation menu with categories like "Project and Governance", "Security", and "Security standards and Zephyr". The main content area is white and shows the breadcrumb "Docs / Latest » Security » Security standards and Zephyr » ETSI 303-645", a link to "Open on GitHub", and a "Report an issue" button. A light pink box contains a note about the latest development branch. The main heading is "ETSI 303-645", followed by a paragraph describing the standard as "Cyber Security for Consumer Internet of Things: Baseline Requirements," and another paragraph detailing its provisions for secure software updates and data protection. A link "here" points to the full version of the standard.

Source: <https://docs.zephyrproject.org/latest/security/standards/etsi-303645.html>

Work on ETSI EN 303-645 in 2023



Zephyr
3.6.99

Search docs (powered by Google) 

Project and Governance

Security

- Zephyr Security Overview
- Security Vulnerability Reporting
- Secure Coding
- Sensor Device Threat Model
- Hardening Tool
- Vulnerabilities

Security standards and Zephyr

ETSI 303-645

Docs / Latest » Security » Security standards and Zephyr » ETSI 303-645

[Open on GitHub](#) [Report an issue with this page](#)

This is the documentation for
documentation of previous releases.

ETSI 303-645

ETSI EN 303 645, also known as ETSI EN 303 645-1, is a standard developed by the European Telecommunications Standards Institute (ETSI).

The standard includes provisions for the minimization of exposed attack surfaces, the identification of challenges and risks associated with the use of the standard, and the identification of security requirements.

Full version of the standard can be found at [ETSI EN 303 645-1](#).

Provision 5.6-3	Device hardware should not unnecessarily expose physical interfaces to attack.	R	Y	Kconfig and Hardening Tool
Provision 5.6-4	Where a debug interface is physically accessible, it shall be disabled in software.	M C	Y	Hardening Tool
Provision 5.6-5	The manufacturer should only enable software services that are used or required for the intended use or operation of the device.	R	Y	Kconfig and Hardening Tool
Provision 5.6-6	Code should be minimized to the functionality necessary for the service/device to operate.	R	Y	Kconfig
Provision 5.6-7	Software should run with least necessary privileges, taking account of both security and functionality.	R	Y	Security Overview
Provision 5.6-8	The device should include a hardware-level access control mechanism for memory.	R	Y	Memory protection
Provision 5.6-9	The manufacturer should follow secure development processes for software deployed on the device.	R	Y	Security Overview and Coding guidelines
Provision 5.7-1	The consumer IoT device should verify its software using secure boot mechanisms.	R	Y	Functionality provided by <i>MCUboot</i> < https://github.com/zephyrproject-rtos/mcuboot >. Also see Security Overview

Source: <https://docs.zephyrproject.org/latest/security/standards/etsi-303645.html#provisions-assessment>

2024 Security Audit with NCC Group



Why External Audit?

- Identifying Vulnerabilities
- Independent Assessment
- Best Practices
- Community Trust
- Reputation

Scope Definition

- Security Objectives
- Components
 - Narrow to something doable and that benefits most users
- Depth of Analysis
- Threat Model

Results from NCCGroup

- Target Zephyr 3.6 / 3.7
 - 02/2024 ~ 03/2024
- Three issues found
 - Two low severity caused by integer overflow and TOCTOU
 - One informational caused by integer overflow

Lessons Learned from the Audit



Defining the scope is hard

- Resource Constraints
- Depth and Breadth
- Future-Proofing
- Stakeholder Agreement

Threat model is useful

- Guiding the Audit Process
- Validating Security Controls
- Facilitating Communication

Comprehensive testing importance

- The audit make it clear the importance of comprehensive testing

Outcomes:

- Enhanced Security
 - The identification and subsequent remediation of even low-severity issues contribute to a more secure system
- Increased Confidence
 - Third-party auditor validated the security and quality of the code base increasing confidence among developers, stakeholders, and users
- Recommendations aligned with Zephyr plans
 - Guided Fuzzing of Libraries and Subsystems

More Details Available...



Details at:

<https://www.youtube.com/watch?v=vEG-Oww9TEs&list=PLzRQULb6-ipHnRUuy2UlpqZjTM9FPWtWx&index=22>

Agenda

- Zephyr overview
- Security best practices focus from the start
- Lessons learned after becoming a CNA
- **Summary and looking forward**

Best Practice Adoption Over Time



- Established Security Committee at project launch in 2016 – meets bi-weekly.
- Secure Coding Practices were publicly [documented](#) for project.
- Zephyr Project [registered as a CVE Numbering Authority](#) with MITRE since 2017.
- [“Gold” Best Practices Badge](#) criteria Core Infrastructure Initiative met in 2018
- Vulnerability Management Process in 2020
 - Vulnerability response criteria publicly documented
 - Product makers can register for free for embargo notifications Zephyr
- SBOM generation in 2021
 - Source SBOM’s for releases and updates going forward from version 2.5
 - Ability to automatically generate SBOM for built images included in version 2.6
- Infrastructure transition to github in 2021, improved automation & interaction with CVE
- Security working group formation in 2022 to improve transparency with community
- Work on self attestation for ETSI 303-645 started in 2023
- Project funded Audit by NCC group in 2024

Security Best Practices Evolve...



- PSIRT processes need to be updated to align with evolving CVE & NVD infrastructure
 - New API is helping with pulling the data, moving over to new infrastructure.
 - Improving consistency via scripting.
 - Open question on determining to what extent to should we assign for low issues, even if not critical to enforce static analysis.
 - Effective monitoring of vulnerabilities in modules is under investigation, as well as module interdependencies.
- Leveraging Automation to prevent security regressions:
 - Weekly Coverity Scans to detect bad practices in imported code have been in common use
 - MISRA scans being incorporated, to evolve to conformance and address issues. Challenge is staging the adoption of rules being enforced, so this can become part of commit testing.
- On Ramping others to help with security issues
 - Maintainers in problematic areas are responsive and effective at this point
 - Volunteers have different level of involvement and background - Secure practices training?
 - Definition of bite size tasks is proving problematic

Interested in Learning More?



zephyrproject.org



github.com/zephyrproject-rtos



lists.zephyrproject.org

[\(https://lists.zephyrproject.org/g/security-wg\)](https://lists.zephyrproject.org/g/security-wg)



chat.zephyrproject.org

#FIRSTCON24

Thank you

どうもありがとうございます

Danke

매우 감사합니다

Gracias

Σας ευχαριστώ

Grazie

Merci Beaucoup

आपका बहुत-बहुत धन्यवाद

Contact:

kstewart@linuxfoundation.org



Abstract

When the Zephyr project(<https://zephyrproject.org/>) launched in 2016, one of the goals was to apply known security best practices to make the S in IoT actually mean something.

This talk will go through the journey of the last 8 years of applying known best security practices to an open source project, including becoming a CVE Numbering Authority, and forming a PSIRT team from volunteers from different companies. Along the way we had to adjust embargo policies due to a bulk vulnerability report, in addition to the occasional vulnerability reported from the community.